

Министерство образования и науки Кыргызской Республики
Кыргызский Государственный Технический Университет
Кафедра Программное обеспечение компьютерных систем

Отчет по предмету
«Software Engineering Processes»

На тему: «Создание CRM системы операторов связи»

Выполнили студенты группы
ПИ(англ)-2-23

Ордobaев Амирхан
Турдахунов Альмир
Тайыпов Элдар

г. Бишкек — 2025

Содержание

| | | |
|-----|---------------------------------|----|
| 1 | Общие сведения | 2 |
| 2 | Термины | 3 |
| 3 | Функциональные требования (MVP) | 4 |
| 3.1 | Абоненты | 4 |
| 3.2 | SIM / Номера | 4 |
| 3.3 | Тарифы и опции | 4 |
| 3.4 | Договоры / Активация | 5 |
| 3.5 | Платежи / Баланс | 5 |
| 3.6 | Тикеты / Поддержка | 5 |
| 3.7 | Роли и доступы (ACL) | 5 |
| 4 | Нефункциональные требования | 6 |
| 5 | UI и навигация | 7 |
| 6 | API (DRF) | 8 |
| 7 | Импорт / Экспорт | 9 |
| 8 | Тестирование и приёмка | 10 |
| 9 | Развёртывание и окружение | 11 |
| 10 | План разработки | 12 |
| 11 | Риски и ограничения | 13 |

1 Общие сведения

Система должна демонстрировать процессы оператора связи: оформление договоров, управление SIM/номерами, биллинг, поддержка и мониторинг услуг в реальном времени. Она должна моделировать типовые сценарии работы оператора связи, обеспечивая наглядное представление всех ключевых бизнес-процессов, включая обработку платежей, управление тарифами, обслуживание абонентов и контроль качества обслуживания.

Система предназначена для использования в учебных и исследовательских целях, предоставляя студентам возможность понять работу CRM-систем в телеком-индустрии на практике. Она позволяет моделировать реальное взаимодействие оператора с абонентами и поддерживает пошаговое выполнение бизнес-процессов. Студенты смогут выполнять следующие действия:

- Создание и редактирование абонентских записей: ввод и обновление информации о клиентах, включая ФИО, контактные данные, документы и статус обслуживания, с автоматической проверкой корректности данных и уникальности ключевых полей.
- Управление SIM-картами: привязка SIM к абоненту, отслеживание статусов (free, active, blocked), просмотр истории использования, автоматическое уведомление о смене статуса и интеграция с договорами.
- Настройка тарифов и дополнительных опций: выбор и назначение тарифных планов, активация или деактивация опций, проверка совместимости тарифов с текущими услугами абонента, расчёт стоимости и возможных скидок.
- Формирование и обработка договоров: пошаговое оформление договора, изменение статусов (active, suspended, closed), управление сроками и условиями договора, автоматическое уведомление пользователей о статусе.
- Внесение платежей и управление балансом: регистрация платежей, автоматическое обновление баланса абонента, формирование детализированных выписок и отчётов, проверка транзакций на корректность, ведение истории платежей.
- Работа с тикетами поддержки: создание тикетов по обращениям абонентов, отслеживание статусов, прикрепление файлов и заметок, контроль сроков обработки, формирование отчетов о производительности службы поддержки.
- Настройка ролей и прав доступа: определение ролей (Admin, Operator, Supervisor, Employee), разграничение прав доступа к различным функциональным модулям, возможность расширения прав через конфигурацию и контроль безопасности данных.
- Аналитика и отчётность: формирование графиков и сводных таблиц по абонентам, платежам, договорам и тикетам, отслеживание ключевых показателей эффективности, поддержка учебных сценариев анализа данных.

Система должна быть интуитивно понятной, обеспечивать логичную навигацию и удобный интерфейс для выполнения всех вышеуказанных действий. Особое внимание уделяется:

- Целостности данных: предотвращение дублирования записей, корректная обработка ошибок, проверка взаимосвязей между сущностями.
- Обработка ошибок: детальная регистрация ошибок, уведомления пользователей и возможность их исправления без потери данных.
- Предотвращению конфликтов при массовых операциях: поддержка транзакций и блокировок для одновременной работы нескольких пользователей, безопасное выполнение массовых обновлений и импорта данных.

2 Термины

Система должна использовать единые термины для всех сущностей и процессов, чтобы обеспечить согласованность и однозначность понимания. Основные термины включают:

- Абонент — пользователь системы, которому предоставляются услуги связи. Абонент может иметь несколько SIM-карт и тарифных планов, а также историю взаимодействия с системой.
- SIM — идентификатор абонента в сети, который используется для предоставления услуг мобильной связи. Каждая SIM имеет уникальные параметры ICCID, IMSI и MSISDN.
- Тариф — план предоставления услуг связи, включающий набор опций, стоимость и условия использования. Тарифы могут быть активными или архивными.
- Договор — официальное соглашение между оператором и абонентом, определяющее права, обязанности и финансовые условия. Договоры могут иметь разные статусы: активный, приостановленный, закрытый.
- Платёж — операция пополнения баланса абонента. Платежи могут быть проведены различными методами и должны фиксироваться в истории транзакций.
- Баланс — текущий остаток средств абонента на счету. Баланс обновляется при проведении платежей и списании абонплаты.
- Роли — система разграничения прав доступа. Каждая роль определяет, какие действия может выполнять пользователь: просмотр данных, редактирование, создание или удаление сущностей.

Каждый термин сопровождается внутренними правилами обработки и ограничениями, которые необходимо учитывать при разработке и тестировании системы. Стандартизация терминологии обеспечивает корректность документации, отчетов и учебных материалов для студентов.

3 Функциональные требования (MVP)

Система должна обеспечивать следующие возможности, реализуемые через конкретные действия и сценарии использования. Каждая строка описывает отдельную реализуемую функцию, доступную в MVP.

3.1 Абоненты

- Система должна позволять создавать новые абонентские записи с проверкой корректности данных.
- Система должна позволять редактировать существующие записи абонентов, включая ФИО, телефон, электронную почту и документы.
- Система должна обеспечивать удаление абонентов с сохранением истории операций для аудита.
- Система должна поддерживать поиск абонентов по ФИО, телефону и номеру документа.
- Система должна обеспечивать фильтрацию абонентов по статусу, дате создания и активности.
- Система должна проверять уникальность номера телефона и документа при добавлении или обновлении записи.
- Система должна предоставлять возможность экспорта списка абонентов в CSV и PDF для отчетности.

3.2 SIM / Номера

- Система должна обеспечивать создание, редактирование и удаление записей SIM-карт.
- Система должна поддерживать массовый импорт SIM-карт из CSV с проверкой корректности данных.
- Система должна проверять уникальность и правильность ICCID, IMSI и MSISDN при добавлении SIM.
- Система должна индексировать поля для быстрого поиска и фильтрации SIM-карт.
- Система должна сохранять историю привязки SIM к абонентам, включая даты активации и деактивации.
- Система должна позволять изменение статуса SIM (free, active, blocked) с соответствующими уведомлениями.

3.3 Тарифы и опции

- Система должна обеспечивать создание, редактирование и удаление тарифов.
- Система должна позволять деактивировать тариф без удаления данных о действующих пользователях.
- Система должна поддерживать настройку дополнительных опций тарифов, таких как пакеты минут, SMS и интернет-трафика.
- Система должна проверять совместимость выбранного тарифа с активными услугами абонента.
- Система должна предоставлять возможность просмотра истории изменений тарифов.

3.4 Договоры / Активация

- Система должна обеспечивать пошаговое оформление договора с обязательной проверкой всех полей.
- Система должна управлять статусами договора: active, suspended, closed.
- Система должна автоматически списывать абонентскую плату по расписанию.
- Система должна привязывать баланс абонента и SIM-карту к договору.
- Система должна хранить историю изменений договора и действий пользователей с ним.
- Система должна уведомлять абонента и операторов о важных событиях по договору (активация, приостановка, закрытие).

3.5 Платежи / Баланс

- Система должна обеспечивать внесение платежей по договору и проверку корректности суммы.
- Система должна автоматически обновлять баланс абонента после внесения платежа.
- Система должна формировать выписки и отчеты по платежам за выбранный период.
- Система должна поддерживать разные методы оплаты (наличные, онлайн-платежи, банковские карты).
- Система должна хранить историю транзакций с возможностью поиска и фильтрации.
- Система должна уведомлять абонента о поступлении платежа и изменении баланса.

3.6 Тикеты / Поддержка

- Система должна обеспечивать создание, редактирование и удаление тикетов.
- Система должна предоставлять фильтрацию и поиск тикетов по статусу, абоненту и дате создания.
- Система должна вести журнал изменений каждого тикета, включая действия операторов.
- Система должна позволять прикрепление файлов и заметок к тикету.
- Система должна формировать отчеты по времени обработки тикетов и эффективности поддержки.
- Система должна уведомлять сотрудников о новых тикетах и изменениях статусов.

3.7 Роли и доступы (ACL)

- Система должна позволять создавать и редактировать роли пользователей: Admin, Operator, Supervisor, Employee.
- Система должна разграничивать права доступа к функционалу в зависимости от роли.
- Система должна обеспечивать аутентификацию пользователей с проверкой пароля и сессий.
- Система должна позволять настраивать расширенные права через конфигурацию и привязку к конкретным действиям.
- Система должна вести лог действий пользователей для аудита безопасности.
- Система должна поддерживать временные ограничения и блокировку пользователей при нарушениях правил.

4 Нефункциональные требования

Система должна удовлетворять следующим требованиям, обеспечивая стабильную, безопасную и масштабируемую работу:

- Технологический стек: использовать современный стек Django 5.x, Django REST Framework (DRF), HTMX для интерактивного интерфейса, TailwindCSS для стилизации и PostgreSQL 16 для хранения данных. Такой стек обеспечивает высокую производительность, безопасность и возможность быстрого расширения функционала.
- Производительность: обрабатывать до 50 000 абонентов и SIM-карт без деградации скорости работы. Система должна поддерживать индексацию и оптимизацию запросов для быстрого поиска и фильтрации данных.
- Транзакции: использовать атомарные транзакции при проведении платежей и оформлении договоров, чтобы гарантировать целостность данных и предотвратить потерю информации при сбоях.
- Безопасность: обеспечивать защиту от CSRF-атак, хеширование паролей с использованием современных алгоритмов (например, bcrypt), аудит действий пользователей и логирование критических событий для последующего анализа.
- Совместимость: корректная работа на последних версиях браузеров Chrome, Edge и Firefox, с адаптивным отображением на разных устройствах (ПК, планшет, мобильный).
- Локализация: поддержка русского языка в интерфейсе и документации, возможность расширения на другие языки.
- Расширяемость и интеграция: система должна позволять добавлять новые модули и интегрироваться с внешними сервисами через API без изменения основной архитектуры.

5 UI и навигация

Интерфейс системы должен быть удобным, интуитивным и обеспечивать пользователю возможность эффективно выполнять все ключевые операции. Каждое требование оформлено как реализуемая функция.

- Система должна предоставлять дашборд с ключевыми показателями работы системы: количество абонентов, активные договоры, общий баланс платежей, количество открытых тикетов и текущую нагрузку на сервис.
- Система должна отображать дашборд с графиками и визуальными индикаторами, такими как диаграммы по платежам, активации SIM и скорости обработки тикетов.
- Система должна использовать модульные карточки для наглядного представления каждой сущности (абонент, SIM, тариф, договор, платеж, тикет) с основной информацией и быстрыми действиями (редактирование, удаление, просмотр деталей).
- Система должна обеспечивать отдельные управляющие страницы для работы с абонентами, SIM-картами, тарифами, договорами, платежами и тикетами, включая полноценный функционал CRUD.
- Система должна предоставлять фильтры по статусам, датам, типам тарифов, суммам платежей и другим ключевым полям для упрощения поиска нужной информации.
- Система должна обеспечивать возможность быстрого поиска данных по ФИО абонента, номеру SIM, номеру договора или другим уникальным идентификаторам.
- Система должна предоставлять сортировку данных по различным полям, включая даты, суммы, статусы и алфавитный порядок.
- Система должна отображать подсказки и пояснения в интерфейсе для новых пользователей, включая обучение работе с формами и фильтрами.
- Система должна поддерживать адаптивный интерфейс, корректно отображающийся на настольных компьютерах, ноутбуках, планшетах и мобильных устройствах, с автоматическим изменением размера элементов.
- Система должна обеспечивать быстрое обновление данных на странице без полной перезагрузки, используя динамические компоненты и AJAX/HTMX технологии.
- Система должна предоставлять удобные формы ввода и редактирования данных с проверкой корректности и отображением ошибок в режиме реального времени.
- Система должна обеспечивать логичную навигацию между разделами, с единым меню, хлебными крошками и кнопками быстрого доступа к часто используемым функциям.
- Система должна предоставлять возможность персонализации интерфейса для различных ролей пользователей, включая отображение только разрешенных модулей и действий.

6 API (DRF)

Система должна предоставлять REST API, обеспечивающий полный доступ к данным и автоматизацию процессов. Каждая функция API должна быть реализуема и тестируема.

- Система должна предоставлять CRUD эндпоинты для создания, чтения, обновления и удаления сущностей: customers, sims, tariffs, contracts, payments, tickets.
- Система должна обеспечивать фильтрацию данных по ключевым полям, включая ФИО, телефон, статус договора, номер SIM и диапазоны дат.
- Система должна поддерживать поиск данных по ключевым атрибутам для быстрого нахождения нужной информации.
- Система должна обеспечивать пагинацию больших наборов данных для оптимизации работы фронтенда и предотвращения перегрузки сервера.
- Система должна предоставлять специальные эндпоинты для массового импорта и экспорта данных в формате CSV, с проверкой корректности и уникальности полей.
- Система должна генерировать подробные отчеты об импорте/экспорте, включая ошибки, дубликаты и некорректные записи.
- Система должна обеспечивать аутентификацию пользователей через сессии и токены, с контролем прав доступа в зависимости от роли (Admin, Operator, Supervisor, Employee).
- Система должна интегрироваться с инструментами автоматической генерации документации API (например, Swagger/OpenAPI), чтобы разработчики могли получать актуальные спецификации.
- Система должна поддерживать версионирование API для обратной совместимости при добавлении новых функций.
- Система должна логировать все обращения к API для последующего аудита и анализа производительности.

7 Импорт / Экспорт

Система должна обеспечивать удобные и безопасные механизмы работы с внешними данными, позволяя эффективно управлять информацией о клиентах, SIM-картах, тарифах и платежах.

- Система должна обеспечивать импорт данных из CSV с проверкой уникальности ключевых полей и корректности формата.
- Система должна обрабатывать ошибки импорта, предоставлять пользователю подробный отчет о некорректных строках и возможность повторного импорта.
- Система должна поддерживать экспорт данных в CSV, PDF и другие форматы для формирования отчетов по платежам, договорам, тикетам и другим сущностям.
- Система должна позволять фильтровать данные перед экспортом, включая выбор по дате, статусу и другим параметрам.
- Система должна поддерживать различные разделители (запятая, точка с запятой, табуляция) и кодировки (UTF-8, Windows-1251) при импорте и экспорте.
- Система должна сохранять историю операций импорта и экспорта для последующего аудита и контроля качества данных.
- Система должна уведомлять администратора о завершении массового импорта или экспорта, включая количество успешно обработанных и некорректных записей.
- Система должна обеспечивать безопасное выполнение массовых операций, предотвращая конфликты и потерю данных при параллельном использовании.

8 Тестирование и приёмка

Система должна проходить комплексное тестирование для обеспечения корректной работы и соответствия функциональным требованиям. Каждое требование оформлено как отдельная проверяемая задача.

- Система должна проходить юнит-тестирование моделей данных, бизнес-логики, функций и методов с различными входными данными.
- Система должна проверять корректность валидации данных при создании и редактировании абонентов, SIM-карт, тарифов и договоров.
- Система должна проходить интеграционное тестирование всех модулей, включая взаимодействие абонентов, платежей, договоров и тикетов.
- Система должна проверять сценарии реального использования: создание абонента, оформление договора, привязка SIM, внесение платежа, обработка тикета поддержки.
- Система должна формировать отчёты о покрытии тестами, найденных ошибках и их исправлении.
- Система должна проходить нагружочное тестирование при большом объёме данных (50k+ абонентов и SIM) и множестве одновременных пользователей.
- Система должна логировать все результаты тестов для последующего аудита и анализа эффективности.
- Система должна обеспечивать приёмку MVP с фиксированием результатов тестирования и сравниванием с ожидаемыми показателями.

9 Развёртывание и окружение

Система должна обеспечивать лёгкое развёртывание и переносимость между различными средами, включая учебные, девелоперские и продакшн.

- Система должна поддерживать Dev окружение: Python 3.12, виртуальное окружение (venv), SQLite, команды migrate, seed, run для локальной разработки.
- Система должна поддерживать Prod (demo/production) окружение с использованием Docker Compose: контейнеры web, db, nginx.
- Система должна использовать конфигурацию через .env переменные (SECRETKEY, DBURL, DEBUG, ALLOWEDHOSTS) для безопасности и гибкости.
- Система должна обеспечивать масштабируемость и возможность переноса на другие среды без изменения кода.
- Система должна логировать все операции развёртывания и проверять корректность подключения всех сервисов.
- Система должна иметь инструкции по установке, запуску и обновлению для разработчиков и тестировщиков.

10 План разработки

Разработка системы будет осуществляться по пошаговому плану, обеспечивающему поэтапную реализацию всех функциональных и нефункциональных требований. Каждый этап включает конкретные задачи и контрольные точки.

1. Подготовка окружения и проектирование: настройка локальных и серверных сред разработки, создание виртуальных окружений, установка всех зависимостей, построение ER-диаграммы базы данных с определением связей между сущностями.
2. Реализация моделей и бизнес-логики: создание моделей данных в Django, реализация правил валидации, бизнес-процессов (расчёт абонплаты, обработка платежей, управление статусами договоров), ведение истории изменений.
3. Разработка API с DRF: создание RESTful эндпоинтов для всех сущностей, настройка фильтров, поиска, пагинации, реализация аутентификации и авторизации на основе ролей.
4. Создание интерфейса на HTMX + Tailwind: разработка интерактивного и адаптивного интерфейса, реализация модульных карточек, фильтров, форм ввода, визуализации данных и динамических обновлений страниц.
5. Реализация биллинга и транзакций: интеграция платежной логики, обеспечение атомарности операций, ведение истории транзакций, генерация отчётов о платежах и балансах.
6. Импорт/Экспорт и тестирование: создание механизмов загрузки и выгрузки данных, написание юнит- и интеграционных тестов, проверка всех ключевых процессов, формирование отчетов о тестировании, исправление найденных ошибок.
7. Развёртывание через Docker Compose: подготовка production и демонстрационных сред, настройка контейнеров для веб-приложения, базы данных и веб-сервера, обеспечение масштабируемости, переносимости и логирования всех операций.

Каждый этап сопровождается подробной документацией, контрольными точками и отчётом о проделанной работе, чтобы упростить контроль проекта и приёмку выполненных задач.

11 Риски и ограничения

При разработке и эксплуатации системы необходимо учитывать следующие ограничения и возможные риски:

- Ограниченнная глубина биллинга и рейтинга: учебный проект не предусматривает сложных алгоритмов тарификации и динамического рейтинга, что может ограничить реализм симуляции.
- Отсутствие реальных телеком-интеграций: все процессы моделируются и эмулируются, нет подключения к реальным операторам связи, что снижает практическую применимость в продакшене.
- Эмуляция трафика и нагрузки: нагрузка на систему для графиков и аналитики создаётся искусственно, что может отличаться от реальных условий эксплуатации.
- Задержки при больших объёмах данных: при обработке десятков тысяч абонентов и SIM возможны замедления в интерфейсе и отчетах, требующие оптимизации запросов и индексации.
- Ограничения по безопасности: в учебной среде могут отсутствовать сложные механизмы защиты от DDoS и интеграции с внешними системами безопасности.
- Зависимость от стека технологий: использование конкретных версий Django, PostgreSQL и TailwindCSS может потребовать обновлений при переходе на новые версии.

Риски должны контролироваться через регулярное тестирование, аудит кода и документирование известных ограничений.

Приложение А — Примеры таблиц данных

Для обеспечения единого подхода к работе с данными система использует следующие структуры. Каждая таблица содержит ключевые поля и пояснения для разработчиков и тестировщиков.

Customer

| Поле | Описание |
|---------------------------------------|---|
| <i>id</i> | Первичный ключ (PK), уникальный идентификатор абонента (uuid или int) |
| <i>first_name, last_name</i> | Фамилия и имя абонента |
| <i>phone</i> | Телефонный номер абонента, должен быть уникальным |
| <i>document_type, document_number</i> | Тип и номер документа (паспорт, ID и т.д.) |
| <i>email</i> | Электронная почта для уведомлений и авторизации |
| <i>created_at, updated_at</i> | Метки времени создания и последнего обновления записи |
| <i>status</i> | Статус абонента (активный, приостановлен, закрыт) |

SIM

| Поле | Описание |
|-------------------------------------|--|
| <i>id</i> | Первичный ключ (PK) |
| <i>iccid</i> | Уникальный идентификатор SIM-карты |
| <i>imsi</i> | Уникальный идентификатор в сети оператора |
| <i>msisdn</i> | Номер SIM, используется для звонков и SMS (индексируемое поле) |
| <i>status</i> | Статус SIM: free, active, blocked |
| <i>contract_id</i> | Внешний ключ (FK) на договор (nullable), отображает текущую привязку |
| <i>activated_at, deactivated_at</i> | Даты активации и деактивации SIM-карты |

Дополнительно

Для других сущностей, таких как Tariffs, Contracts, Payments, Tickets, структура таблиц аналогично содержит ключевые поля, метки времени, статусы и ссылки на связанные объекты, обеспечивая целостность данных и удобство разработки.