

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Институт информационных технологий и анализа данных

наименование института

Допускаю к защите

Руководитель

подпись

Т.А. Баталова

И.О. Фамилия

Интеграция облачного хранилища Nextcloud в

платформу BitManager

наименование темы

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту по дисциплине

ТЕХНОЛОГИИ РАЗРАБОТКИ ПРОГРАММНЫХ КОМПЛЕКСОВ

1.001.00.00 - ПЗ

обозначение документа

Выполнил студент

АСУ6-22-2

шифр группы

подпись

М.П. Беляевский

И.О. Фамилия

Нормоконтроль

подпись

Т.А. Баталова

И.О. Фамилия

Курсовой проект защищен с оценкой

Иркутск 2025 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ЗАДАНИЕ
НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ**

По курсу Технологии разработки программных комплексов
Студенту _____

(фамилия, инициалы)

Тема проекта Интеграция облачного хранилища Nextcloud в
платформу BILmanager

Исходные данные: _____

Разработка модуля интеграции облачного хранилища Nextcloud в
платформу BILmanager (вариант № 1)

Рекомендуемая литература:

1. Гутгарц Р.Д Проектирование автоматизированных систем обработки информации и управления: учебное пособие для академического бакалавриата. – М.: Издательство Юрайт, 2019.
2. Проектирование АСОИУ [Электронный ресурс] : методические указания по выполнению курсового проекта: 09.03.01 «Информатика и вычислительная техника» для бакалавров по специальности «Автоматизированные системы обработки информации и управления» / Иркут. нац. исслед. техн. ун-т ; сост. Р. Д. Гутгарц. - Электрон. дан. - Иркутск : ИРНИТУ, 2018.
3. Проектирование информационных систем : учебник и практикум для академического бакалавриата / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук ; под общ. ред. Д. В. Чистова. – М. : Издательство Юрайт, 2016.
4. Рудинский И.Д. Технология проектирования автоматизированных систем обработки информации и управления : учебное пособие для вузов / И. Д. Рудинский. - Москва : Горячая линия - Телеком, 2011.

Графическая часть на _____ листах.

Дата выдачи задания «20» сентября 2025 г.

Задание получил _____

подпись

М.П. Беляевский

И.О. Фамилия

Дата представления проекта руководителю _____ 2025 г.

Руководитель курсового проектирования _____

Т.А. Баталова

Содержание

Введение.....	5
1 Анализ.....	7
1.1 Описание предметной области.....	7
1.1.1 Современный ландшафт IT-услуг для бизнеса: Сдвиг в сторону подписки и самообслуживания.....	7
1.1.2 Биллинговые системы как ядро автоматизации.....	7
1.1.3 Платформы для совместной работы и файлового хостинга: Спрос на корпоративные облака.....	8
1.1.4 Проблемы разрыва между бизнес-логикой и технической реализации	8
1.1.5 Актуальность разработки модуля интеграции для BILLmanager и Nextcloud.....	9
1.2 Обзор существующих программных средств.....	10
1.2.1 Общие тенденции на рынке биллинговых систем.....	10
1.2.2 HostBill.....	10
1.2.3 WHMCS.....	11
1.2.4 Blesta.....	11
1.2.5 WiseCP.....	11
1.2.6 Сравнительный анализ и вывод.....	12
1.3 Процесс AS-IS vs TO-BE.....	13
1.4 Описание вариантов использования.....	14
1.5 Постановка задачи.....	16
2 Проектирование.....	18
2.1 Выбор и обоснование средств проектирования и реализации.....	18
2.1.1 Средства визуального проектирования и моделирования.....	18
2.2 Проектирование архитектуры приложения.....	21
2.2.1 Проектирование архитектуры приложения.....	21
2.2.2 Описание компонентов.....	21
2.2.3 Обоснование выбранной архитектуры.....	23
2.3 Проектирование хранилища данных.....	24
2.3.1 Процесс создания аккаунта.....	24
2.3.2 Основные сущности.....	25
2.3.3 Логическая модель.....	31
2.4 Проектирование пользовательского интерфейса.....	32
2.4.1 Верхнеуровневое определение экранов.....	32

2.4.2 Иерархия и переходы между экранами.....	32
2.4.3 Спецификация экранов.....	34
2.4.4 Состояния экранов.....	36
2.4.5 Описание макетов экранов.....	39
.....	40
2.4.6 Карта переходов между экранами.....	43
Заключение.....	44
Список использованных источников.....	48

Введение

BILLmanager — это комплексная биллинговая панель управления для хостинг-провайдеров, дата-центров и IT-компаний. Это не просто система для выдачи счетов, а централизованная платформа для автоматизации всего бизнес-цикла:

- Управление клиентами и их услугами: Создание аккаунтов, назначение тарифов, приостановка и активация услуг.
- Биллинг и финансы: Автоматическое выставление счетов, прием платежей (онлайн-эквайринг, банковские переводы), учет дебиторской задолженности, формирование финансовых отчетов.
- Продажа и продление услуг: Настройка тарифных планов (например, "Старт", "Профи", "Бизнес") с гибкими периодами оплаты (месяц, год и т.д.), автоматическое напоминание о продлении.
- Панель клиента: Личный кабинет, где клиент сам может заказать новые услуги, оплатить счета, отслеживать состояние своих заказов и обращаться в поддержку.
- Интеграция с техническими панелями: Ключевая особенность BILLmanager — его способность взаимодействовать с другими системами (например, cPanel, ISPmanager, VMmanager) через API. Это позволяет автоматически создавать учетные записи хостинга или VPS при успешной оплате.

Nextcloud и ownCloud — это мощные, самохостируемые платформы для синхронизации и обмена файлами, альтернативы таким коммерческим сервисам, как Dropbox или Google Drive. Они предоставляют полный контроль над данными, так как развертываются на вашей собственной инфраструктуре.

Основные возможности:

- Синхронизация файлов: Клиенты для ПК (Windows, macOS, Linux) и мобильных устройств (iOS, Android) для синхронизации папок.
- Веб-интерфейс: Доступ к файлам через любой браузер из любой точки мира.
- Общий доступ к файлам и папкам: Настройка ссылок для скачивания с паролями и сроками действия.
- Коллаборация: Встроенные офисные приложения (текстовые редакторы, таблицы), календари, контакты, чаты и видеоконференции (особенно в Nextcloud).

- Экосистема приложений: Большое количество плагинов для расширения функционала (например, интеграция с почтой, CRM и т.д.).

Ключевое отличие от простого FTP-хостинга: это не просто дисковое пространство, а целая экосистема для совместной работы с файлами.

ownCloud является прародителем Nextcloud (Nextcloud был форком ownCloud) и предлагает схожий функционал, часто с более корпоративно-ориентированным подходом.

Проект bill-nextcloud представляет собой модуль для BILLmanager, который позволяет связывать продажи хранилищ данных Nextcloud/ownCloud с биллинг-системой. Идея в том, чтобы абоненты могли автоматически приобретать и продлевать объём дискового пространства или услуги хранения файлов через BILLmanager, а при изменении тарифов система сама синхронизировала квоты в Nextcloud/ownCloud. Проще говоря: сделать Nextcloud «товаром» внутри экосистемы BILLmanager — клиент оформляет услугу хранения в панели, а технически это отражается в настройках самого файлового облака.

Таким образом, проект объединяет биллинг и файловый сервис, создавая сквозной поток продаж, выставления счетов и управления ресурсами.

1 Анализ

1.1 Описание предметной области

1.1.1 Современный ландшафт IT-услуг для бизнеса: Сдвиг в сторону подписки и самообслуживания

Современный рынок IT-услуг для бизнеса и частных лиц характеризуется повсеместным переходом от единоразовых покупок программного обеспечения к модели SaaS (Software as a Service) — программному обеспечению как услуге по подписке

Ключевые ожидания клиентов сегодня — это:

- Гибкость: Возможность быстро выбрать, изменить или отменить необходимый объем услуг (тарифный план).
- Самообслуживание: Минимизация взаимодействия с менеджерами через автоматизированные личные кабинеты.
- Мгновенная активация: Получение услуги сразу после оплаты, без задержек.
- Прозрачность: Четкое понимание стоимости, сроков действия и потребляемых ресурсов.

Этот тренд является фундаментальным для хостинг-индустрии и провайдеров IT-услуг, которые стремятся соответствовать ожиданиям рынка.

1.1.2 Биллинговые системы как ядро автоматизации

Для автоматизации продаж и управления клиентами провайдеры используют специализированное программное обеспечение — биллинговые системы (или панели управления биллингом). Эти системы выступают центральным узлом, который:

- Автоматизирует финансовые операции: выставляет счета, учитывает платежи, контролирует дебиторскую задолженность.
- Управляет жизненным циклом услуги: активирует, приостанавливает, возобновляет или удаляет услуги клиента на основе их платежного статуса.
- Предоставляет клиенту инструменты самообслуживания через личный кабинет.

- Интегрируется с техническими платформами для автоматической поставки услуг (например, создание виртуальной машины или учетной записи хостинга).

BILLmanager является ярким представителем такого класса систем, предлагающим мощный функционал для хостинг-провайдеров.

1.1.3 Платформы для совместной работы и файлового хостинга: Спрос на корпоративные облака

Параллельно с трендом на SaaS растет спрос на решения для безопасного хранения, синхронизации и совместной работы с файлами. Такие платформы, как Nextcloud и ownCloud, занимают здесь уникальную нишу:

- Конфиденциальность и контроль: В отличие от публичных облаков (Google Drive, Dropbox), они разворачиваются на инфраструктуре провайдера или самого клиента, что обеспечивает полный контроль над данными.
- Богатая функциональность: Это не просто «диск», а полноценные рабочие среды, включающие общий доступ к файлам, календари, задачи, офисные пакеты и инструменты для коммуникации.
- Корпоративная ориентация: Идеально подходят для малого и среднего бизнеса, нуждающегося в защищенном и функциональном инструменте для командной работы.

Это делает их привлекательным товаром в портфеле услуг IT-провайдеров.

1.1.4 Проблемы разрыва между бизнес-логикой и технической реализацией

Несмотря на наличие готовых и зрелых решений в каждой из областей (биллинг и файловые облака), между ними существует значительный технологический разрыв.

Суть проблемы: Биллинговая система (BILLmanager) и платформа для предоставления услуги (Nextcloud/ownCloud) существуют изолированно. Процесс продажи и обслуживания услуги файлового хранилища требует ручного вмешательства:

1. При продаже: Клиент оплачивает тариф в биллинге. Далее менеджеру вручную необходимо:
 - Зайти в админ-панель Nextcloud/ownCloud.
 - Создать новую учетную запись.

- Вручную выставить квоту дискового пространства, соответствующую оплаченному тарифу.
 - Передать данные для доступа клиенту.
2. При смене тарифа (апгрейд/даунгрейд): Менеджер должен вручную найти пользователя и изменить его квоту.
 3. При просрочке платежа: Менеджер должен вручную заблокировать аккаунт или уменьшить квоту до нуля.
 4. При продлении: Процесс часто требует ручного подтверждения.

Последствия этой проблемы:

- Низкая операционная эффективность: Ручные операции отнимают время системных администраторов и менеджеров.
- Риск человеческих ошибок: Возможна опечатка в логине, неправильно выставленная квота, забытая блокировка.
- Невозможность масштабирования: Ручное управление становится узким местом при росте числа клиентов.
- Ухудшение клиентского опыта: Активация услуги и изменения в ней происходят не мгновенно, а с задержкой, зависящей от загруженности сотрудника.

1.1.5 Актуальность разработки модуля интеграции для BILLmanager и Nextcloud

Разрабатываемый модуль bill-nextcloud является прямым и актуальным ответом на выявленную проблему.

Актуальность заключается в следующем:

1. Полная автоматизация: Модуль устраняет разрыв между системами, создавая мост между бизнес-логикой BILLmanager и технической платформой Nextcloud/ownCloud. Это превращает сложную файловую платформу в простой для продажи и управления «товар».
2. Соответствие рыночным трендам: Позволяет провайдеру предлагать услугу в формате SaaS с мгновенной активацией и самообслуживанием, что соответствует ожиданиям современного клиента.
3. Конкурентное преимущество: Провайдер, который может автоматизировано предоставлять такие сложные услуги, как

корпоративное файловое облако, получает значительное преимущество на рынке.

4. Экономическая эффективность: Разработка модуля окупается за счет экономии времени сотрудников, исключения ошибок и возможности привлекать больше клиентов за счет автоматизации.

Таким образом, проблема ручного управления услугами файлового хостинга в связке «биллинг-платформа» является значительным барьером для развития провайдеров, а разработка модуля для их интеграции — это актуальная и практическая задача, направленная на устранение этого барьера.

1.2 Обзор существующих программных средств

1.2.1 Общие тенденции на рынке биллинговых систем

Рынок биллинговых и автоматизационных систем для хостинг-услуг предлагает множество решений, охватывающих управление клиентами, выставление счетов, автоматические продления, интеграцию с хостинг-панелями и API-интерфейсы. Эти системы ориентированы на упрощение процессов заказов и обслуживания услуг, снижения ручного труда и повышения скорости. Однако интеграция с облачными хранилищами, особенно с Nextcloud/ownCloud, реализована далеко не везде или лишь частично. Многие системы либо оставляют такую интеграцию как пользовательский плагин, либо не предусматривают её вовсе. Это создаёт возможность появления специализированных официальных решений, особенно в экосистемах, где BILLmanager имеет значительную долю рынка и где провайдеры заинтересованы в расширении портфеля облачных услуг.

1.2.2 HostBill

HostBill обеспечивает встроенную поддержку модулей Nextcloud и ownCloud как часть своих “Backup / File Sharing” интеграций. Модуль Nextcloud для HostBill позволяет продавать и автоматически provision-ить учётные записи, управлять жизненным циклом (создание, приостановка, удаление, апгрейд/даунгрейд ресурсов).

Модуль OwnCloud в HostBill также является фактически тем же модулем — один модуль может обслуживать и OwnCloud, и Nextcloud. Лицензия на этот модуль (OwnCloud & NextCloud) доступна через маркетплейс HostBill по цене \$99 (одноразово)

Таким образом, HostBill уже предлагает полнофункциональное решение для интеграции облачных хранилищ, что делает его конкурентом в части функциональности, хотя не в экосистеме BILLmanager.

1.2.3 WHMCS

WHMCS также поддерживает модули для интеграции с Nextcloud: существуют коммерческие расширения, такие как *Nextcloud Provisioning Module*, дополнения, реализующие автоматизацию учётных записей, квот и жизненного цикла пользователей.

Доступность таких модулей делает WHMCS одним из немногих конкурентных биллинговых решений с готовыми облачными интеграциями. Однако эти модули не ориентированы на BILLmanager и требуют использования другой экосистемы.

1.2.4 Blesta

Для Blesta существует коммерческий (платный) модуль Nextcloud Provisioning, который можно установить как расширение и настроить взаимодействие через API Nextcloud. Кроме того, на сайте расширений Blesta упоминается модуль *Nextcloud* как один из доступных модулей автоматизации.

Тем не менее, доступность и зрелость этого модуля могут быть ограничены (например, поддержка только одного сервера).

1.2.5 WiseCP

Среди других платформ есть WiseCP — для неё есть модуль *Nextcloud* разработанный NetHeberg, который позволяет предлагать клиентам тарифы на облачное хранилище, проверять и отображать использованный объём, активировать / приостанавливать / удалять услуги. Также имеется модуль *Digital Product* (через WiseCP/PuqCloud) для продажи цифровых товаров, где контент хранится в папках Nextcloud-сервера, и после оплаты клиент получает персональную ссылку на скачивание. Эти модули демонстрируют, что интеграция с Nextcloud не ограничивается крупнейшими биллинговыми системами, но также охватывает решения среднего класса.

1.2.6 Сравнительный анализ и вывод

Все конкуренты и их сравнение представлены в таблице 1.

Таблица 1 - Таблица конкурентов

Платформа	Наличие модуля	Функции	Сложность внедрения	Плюсы	Минусы
HostBill	Да	Автоматическое provision, suspend/un suspend, terminate, upgrade/downgrade, управление квотами, группы пользователей	Низкая — модуль “из коробки”	Сильная функциональность	Высокая цена
WHMCS	Да	Provision, квоты, статистика диска, поддержка языков, suspend/terminate	Средняя — установка коммерческого модуля, поддержка настроек	Широко используется	Требуется лицензирование
Blesta	Да	Provision аккаунтов, настройка квот, управление серверами	Средняя — модуль сторонний, возможно, с ограничениями	Подходит для провайдеров, готовых к настройке	Функциональность может быть меньше, чем у HostBill
WiseCP	Да	Хранилище как	Низкая-средняя —	Более доступная	меньше известности; может быть меньше

Платформа	Наличие модуля	Функции	Сложность внедрения	Плюсы	Минусы
		услуга, учет использованного объема, управление услугой	модуль куплен и установлен	цена	поддержки/дополнительных функций

Вывод: большинство готовых функциональных модулей интеграции с Nextcloud/ownCloud присутствует в платформах HostBill, WHMCS, Blesta, а также в WiseCP. Однако все они вне экосистемы BILLmanager. Для пользователей BILLmanager (что особенно актуально для рынка СНГ и Восточной Европы) отсутствует готовое решение. Это создаёт пустующую нишу, которую может заполнить проект bill-nextcloud. Его разработка принесёт реальную ценность: официальное, готовое к внедрению решение, устраняющее разрыв между BILLmanager и облачными хранилищами, повышающее автоматизацию и эффективность провайдеров.

1.3 Процесс AS-IS vs TO-BE

Два процесса AS-IS и TO-BE представлены на рисунке 1, в нотации BPMN.

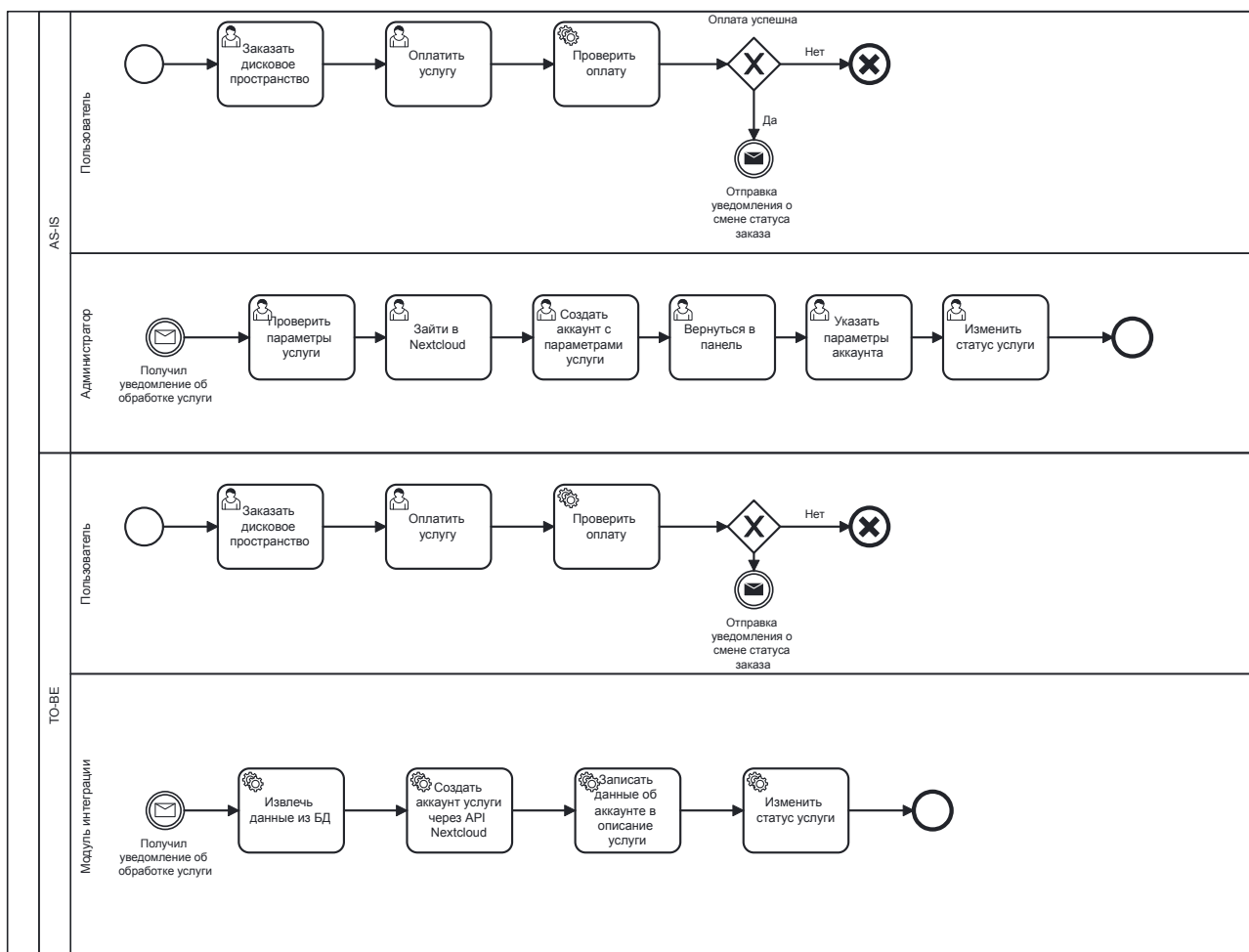


Рисунок 1 - BPMN визуализация бизнес-процессов

1.4 Описание вариантов использования

Варианты использования перечислены на рисунок 2.

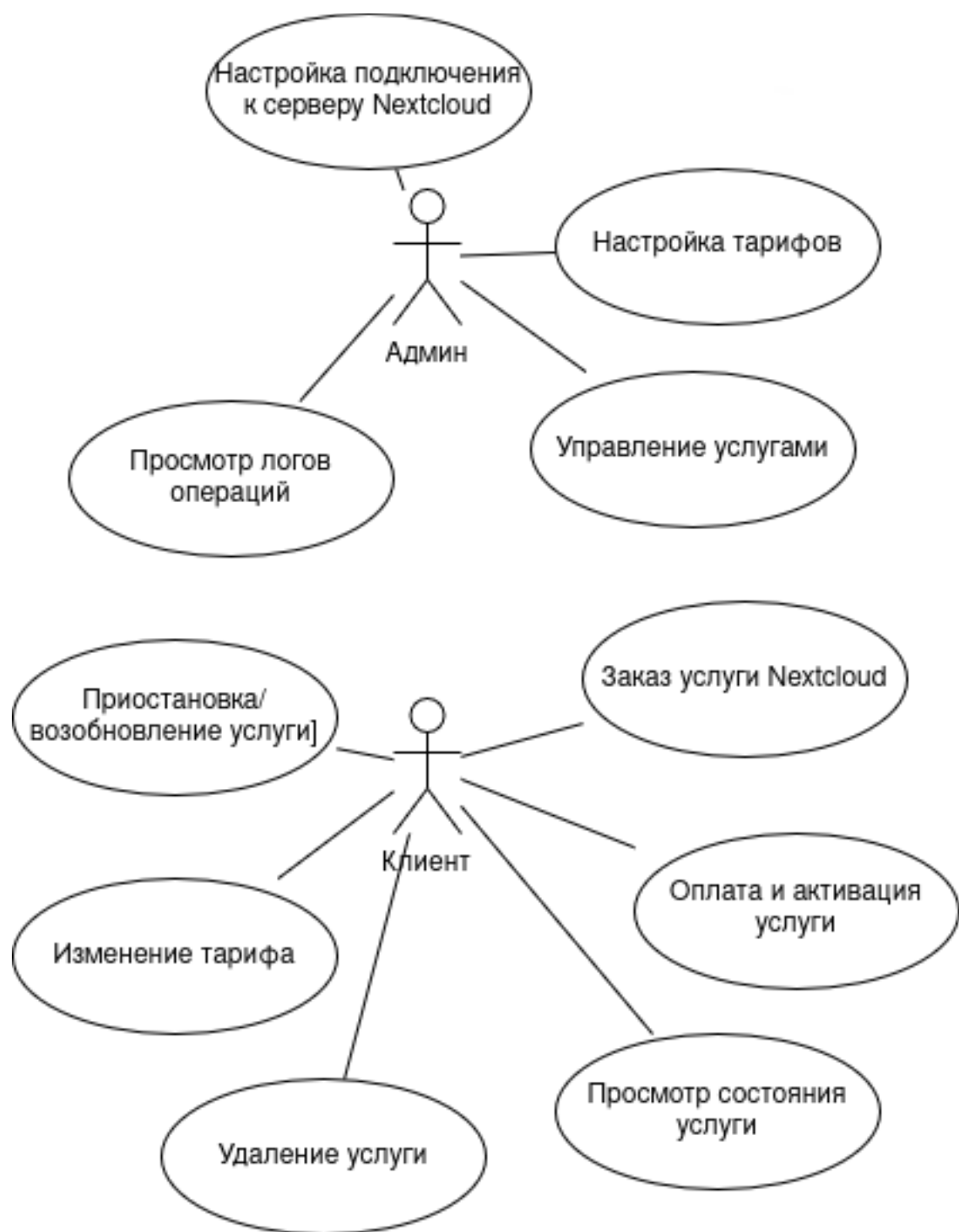


Рисунок 2 - Варианты использования

1.5 Постановка задачи

Цель проекта

На основании анализа предметной области и выявленной проблемы разрыва между бизнес-логикой биллинговой системы и технической платформой предоставления услуги, целью работы является разработка программного модуля bill-nextcloud для интеграции BILLmanager с платформами Nextcloud и ownCloud.

Проект направлен на автоматизацию процессов управления услугами файлового хостинга, включая создание, изменение, приостановку, возобновление и удаление аккаунтов на стороне облачной платформы, с полной синхронизацией с биллинг-системой.

Основные задачи проекта

Для достижения поставленной цели необходимо реализовать следующие задачи:

1. Разработка архитектуры модуля. Необходимо спроектировать структуру модуля, определяющую:

- Способ его установки и взаимодействия с ядром BILLmanager.
- Модель данных для хранения параметров, связывающих услугу в биллинге с аккаунтом в облаке (идентификатор пользователя, URL сервера, токены доступа).
- Алгоритмы обработки событий жизненного цикла услуги.

2. Реализация функционала конфигурации модуля в админ-панели BILLmanager. Модуль должен предоставлять администратору интерфейс для:

- Настройки подключения к одному или нескольким серверам Nextcloud/ownCloud (URL, учетные данные администратора или OAuth-токен).
- Определения шаблонов тарифных планов, где будет задаваться объем дискового пространства (квота).

3. Реализация автоматизации жизненного цикла услуги. Модуль должен отслеживать события в BILLmanager и выполнять соответствующие действия на стороне облачной платформы через ее API:

- Создание услуги (активация после оплаты): Автоматическое создание пользователя на сервере Nextcloud/ownCloud и назначение ему квоты, соответствующей тарифному плану.
- Изменение услуги (смена тарифа): Автоматическое изменение квоты дискового пространства для существующего пользователя при апгрейде или даунгрейде тарифа.
- Приостановка/возобновление услуги: Автоматическая блокировка или разблокировка аккаунта пользователя при просрочке платежа или его погашении.
- Удаление услуги: Автоматическое удаление пользователя из облачной платформы при окончательном удалении услуги из биллинга.

4. Обеспечение обработки ошибок и отказоустойчивости. Модуль должен быть устойчив к сетевым сбоям и ошибкам API:

- Реализация механизма повторных попыток отправки запросов.
- Логирование всех операций и ошибок для последующего анализа администратором.
- Предоставление понятного статуса выполнения операции в интерфейсе BILLmanager.

5. Тестирование работоспособности решения. Проверка корректности работы модуля на тестовых стендах, включая:

- Тестирование сценариев активации, изменения, приостановки и удаления услуг.
- Проверка совместимости с различными версиями ОС на которых распространяется BILLmanager

Ожидаемый результат

В результате успешной реализации проекта будет создан программный продукт, который позволит:

- Синхронизировать биллинг и файловый сервис в единой экосистеме.
- Минимизировать ручной труд сотрудников провайдера.
- Снизить риск ошибок при управлении аккаунтами клиентов.
- Обеспечить масштабируемость услуг при росте числа клиентов.
- Повысить качество клиентского опыта за счёт мгновенной активации и обновления услуг.

2 Проектирование

2.1 Выбор и обоснование средств проектирования и реализации

2.1.1 Средства визуального проектирования и моделирования

BPMN 2.0 (Business Process Model and Notation)

Описание: Стандарт нотации для моделирования бизнес-процессов, позволяющий визуально представить последовательность операций и взаимодействие между системами.

Обоснование выбора:

- Универсальный стандарт, понятный как техническому, так и деловому окружению
- Позволяет чётко описать процессы AS-IS (текущее состояние с ручными операциями) и TO-BE (автоматизированное состояние)
- Применимо для описания жизненного цикла услуги (активация → изменение → приостановка → удаление)

Инструменты: Draw.io, Mermaid

Применение в проекте: Моделирование бизнес-процессов управления аккаунтами Nextcloud

UML диаграммы

Описание: Унифицированный язык моделирования для представления архитектуры и взаимодействия компонентов системы.

Обоснование выбора:

- Позволяет чётко описать структуру модуля и взаимодействие между компонентами
- Поддерживает различные типы диаграмм: классов, последовательности, компонентов, развёртывания
- Стандарт для проектирования ПО

Инструменты: Draw.io, Mermaid

Применение в проекте:

- Диаграммы компонентов для показа архитектуры интеграции

ER-диаграммы (Entity-Relationship)

Описание: Модель для представления структуры данных и связей между сущностями базы данных.

Обоснование выбора:

- Стандарт для проектирования схемы БД
- Позволяет визуализировать отношения между таблицами (1:1, 1:N, M:N)
- Упрощает нормализацию схемы БД

Инструменты: Draw.io, Mermaid

Применение в проекте: Логическая модель данных для сущностей Nextcloud (пользователи, серверы, операции, логи)

2.1.2 Средства разработки программного обеспечения

Язык программирования: Python 3.8+

Обоснование выбора:

- Совместимость с экосистемой BILManager (BILManager поддерживает обработчики на Python)
- Богатый набор библиотек для работы с HTTP (requests), криптографией (cryptography), логированием
- Простота и быстрое прототипирование
- Кроссплатформенность (Linux, Windows, macOS)
- Большое сообщество разработчиков

Основные библиотеки:

- requests — HTTP клиент для REST API
- cryptography — шифрование учётных данных
- Стандартные: json, logging, xml.etree.ElementTree, hashlib, re, time, datetime

Средства управления версиями

Git + GitHub

Обоснование выбора:

- Стандарт для контроля версий в современной разработке
- Позволяет отслеживать изменения, ветвление, коллаборацию
- Проект уже размещён на GitHub: [LC208/bill-nextcloud](https://github.com/LC208/bill-nextcloud)

Инструменты тестирования

Фреймворки:

- pytest — модульное и интеграционное тестирование
- unittest.mock — мокирование зависимостей (API Nextcloud, БД)

Анализ качества кода:

- pylint, flake8 — проверка стиля PEP 8
- coverage.py — измерение покрытия тестами (целевое значение $\geq 80\%$)

Интеграционное тестирование:

- Docker контейнеры (docker-compose.yml) для развёртывания тестовых инстансов Nextcloud

Средства логирования и мониторинга

Логирование:

- Модуль logging Python для вывода в файл /var/log/mgr5/pmnextcloud.log
- Интеграция с системой логирования BILLmanager

Мониторинг (опционально):

- Zabbix для экспорта метрик (время обработки, количество ошибок)
- Prometheus для сбора метрик

2.2 Проектирование архитектуры приложения

2.2.1 Проектирование архитектуры приложения

Архитектура приложения показана на рисунке 3.

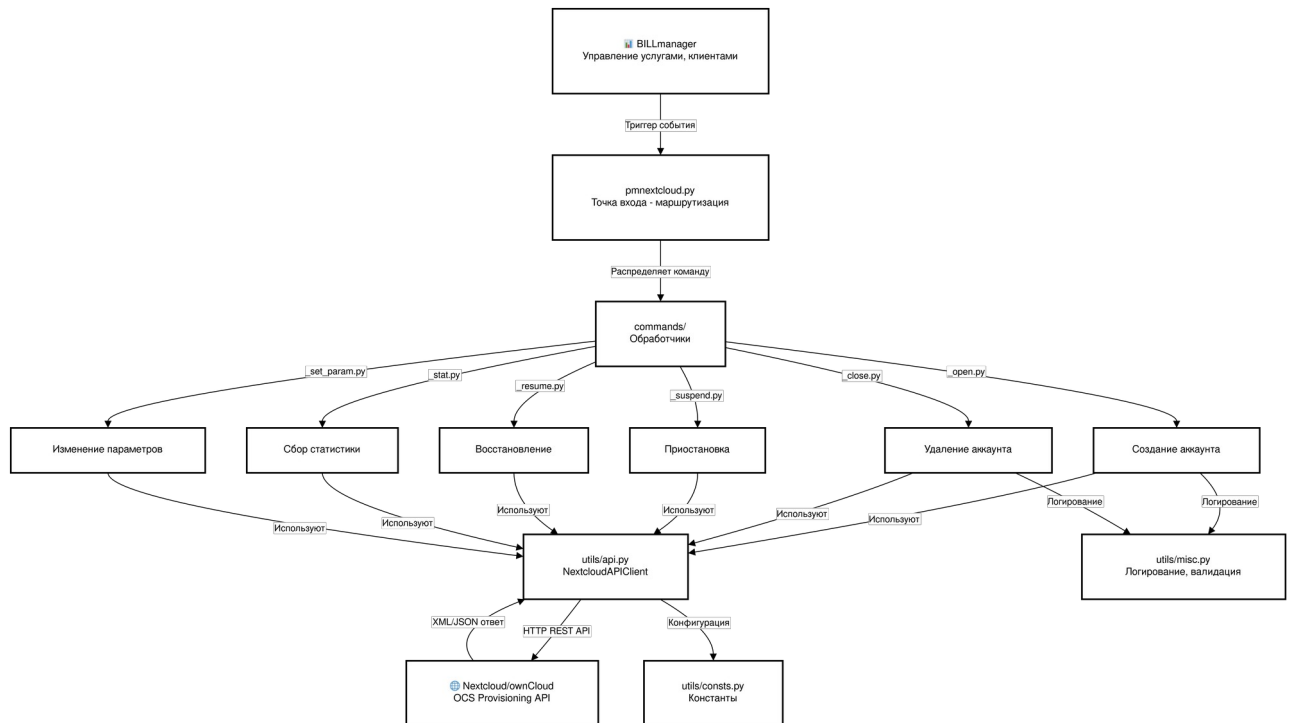


Рисунок 3 - Архитектура модуля

2.2.2 Описание компонентов

Точка входа: pmnextcloud.py

Главный модуль, регистрируется в BILLmanager как обработчик услуг (pmnextcloud).

Функции:

- Получение команды от BILLmanager
- Маршрутизация на соответствующий обработчик
- Обработка результатов и возврат статуса

Входные параметры: --command=<command> --subcommand --<param>=<val>.

Команды: open, close, suspend, resume, stat, set_param, check_connection

Обработчики команд (commands/)

Команда	Файл	Операция	Описание
open	_open.py	Создание	Активация услуги → создание пользователя в Nextcloud

Команда	Файл	Операция	Описание
close	_close.py	Удаление	Отмена услуги → удаление пользователя
suspend	_suspend.py	Приостановка	Неоплата → деактивация аккаунта
resume	_resume.py	Восстановление	Оплата → активация аккаунта
stat	_stat.py	Статистика	Сбор данных об использовании квоты
set_param	_set_param.py	Изменение	Апгрейд тарифа → изменение квоты
check_connection	_check_connection.py	Проверка	Тест соединения с сервером Nextcloud

REST API клиент (utils/api.py)

Класс NextcloudAPIClient для взаимодействия с OCS Provisioning API Nextcloud.

Основные методы:

- create_user() → POST /users
- delete_user() → DELETE /users/{userid}
- get_user() → GET /users/{userid}
- set_quota() → PUT /users/{userid}/quota
- disable_user() / enable_user() → PUT /users/{userid}/disable|enable
- add_to_group() → POST /users/{userid}/groups
- test_connection() → проверка доступности API

Парсинг ответов: XML (OCS формат) или JSON

Утилиты (utils/)

utils/consts.py:

- Константы: статусы, типы операций, пределы квот, таймауты

utils/misc.py:

- Логирование (setup_logging())
- Генерация паролей (generate_secure_password())
- Валидация (validate_username(), validate_email(), validate_quota())
- Повторные попытки (retry_with_exponential_backoff())

2.2.3 Обоснование выбранной архитектуры

Обоснование представлено в виде интеллект-карты, каждая ветвь которой представляет плюсы выбранного подхода к архитектуре приложения. Интеллект-карта показана на рисунке 4.

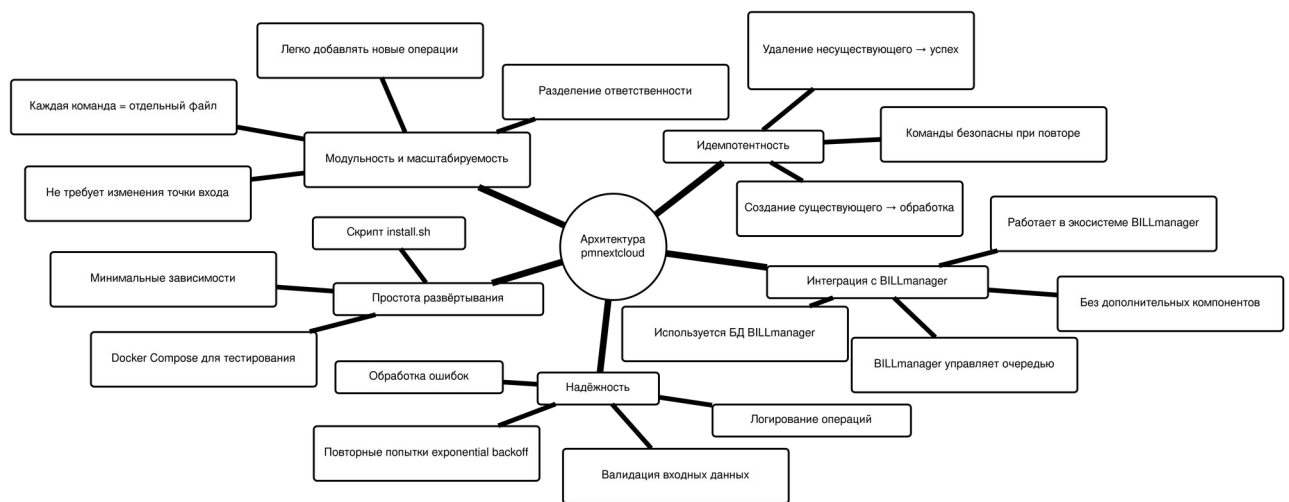


Рисунок 4 - Интеллект-карта

2.3 Проектирование хранилища данных

2.3.1 Процесс создания аккаунта

Процесс создания аккаунта показан в виде диаграммы последовательности на рисунке 5.

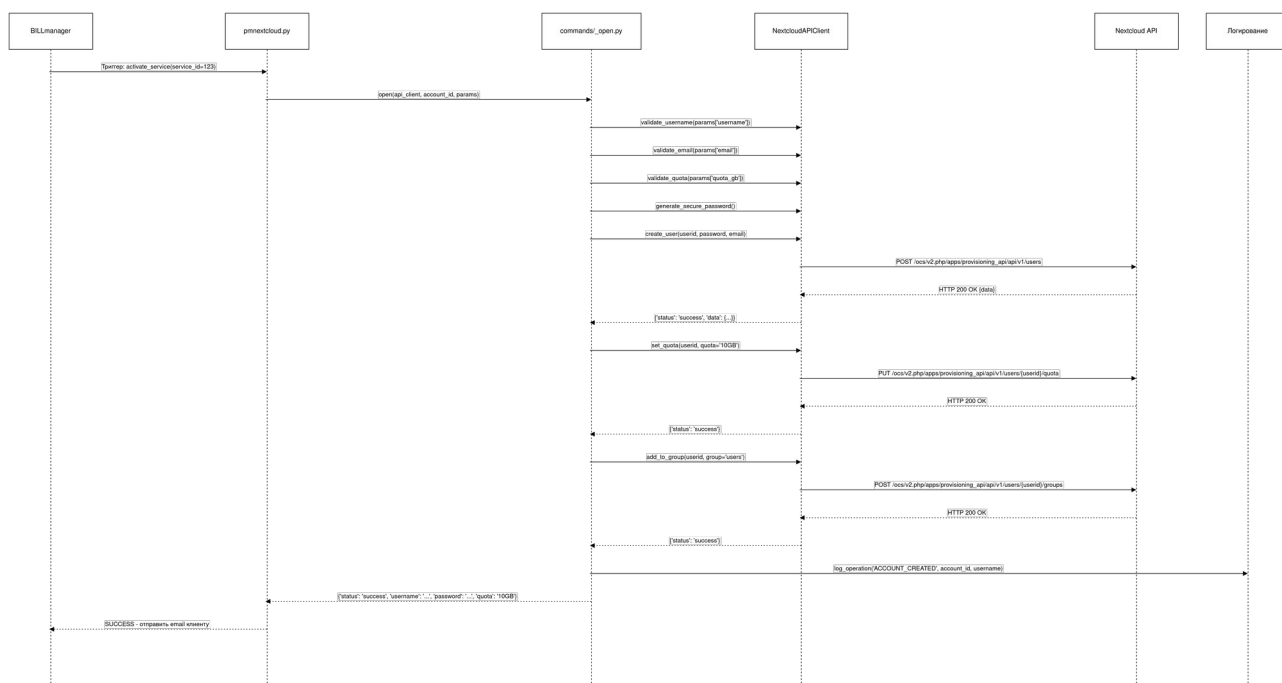


Рисунок 5 - Диаграмма последовательности

2.3.2 Основные сущности

Таблицы BILLmanager, используемые модулем

1. **account** — Клиенты (см. таблицу 2)

Клиенты, для которых активирована услуга Nextcloud.

Таблица 2 - Клиенты

Поле	Тип	Назначение
id	INT	Первичный ключ — ID клиента
name	VARCHAR	Наименование клиента
parent	INT (FK)	Родительская группа клиентов
level	INT	Уровень доступа
registration_date	TIMESTAMP	Дата регистрации
country	INT (FK)	Страна клиента

СВЯЗЬ: account → item (один клиент может иметь много услуг)

2. **user** — Пользователи BILLmanager (см. таблицу 3)

Пользователи, которые получают доступ к Nextcloud.

Таблица 3 - Пользователи

Поле	Тип	Назначение
id	INT	Первичный ключ — ID пользователя
account	INT (FK)	Ссылка на клиента
name	VARCHAR	Имя пользователя (username)
password	VARCHAR	Хеш пароля в BILLmanager
email	VARCHAR	Email (опционально)
level	INT	Уровень доступа (16 = пользователь клиента)
enabled	BOOLEAN	Активность пользователя
phone	VARCHAR	Телефон

Связь: user.account → account.id

ВАЖНО: user в BILLmanager НЕ синхронизируется автоматически с пользователями Nextcloud. Для каждой услуги **item** создаётся отдельный пользователь Nextcloud.

3. **processingmodule** — Обработчик услуг (см. таблицу 4)

Регистрация модуля pmnextcloud в BILLmanager.

Таблица 4 - Обработчик услуг

Поле	Тип	Назначение
id	INT	Первичный ключ
name	VARCHAR	Наименование (например, "Nextcloud Integration")
module	VARCHAR	Тип модуля (например, "pmnextcloud")
active	BOOLEAN	Активность обработчика
config	JSON	Внутренняя конфигурация
laststatdate	TIMESTAMP	Дата последней проверки статуса

Пример: processingmodule.id = 1, name = "Nextcloud Integration", module = "pmnextcloud"

4. **processingparam** — Параметры обработчика (открытые) (см. таблицу 5)

Конфигурация подключения к Nextcloud (незашифрованные параметры).

Таблица 5 - Параметры обработчика

Поле	Тип	Назначение
id	INT	Первичный ключ
processingmodule	INT (FK)	Ссылка на обработчик

Поле	Тип	Назначение
intname	VARCHAR	Внутреннее имя параметра
value	TEXT	Значение параметра

Примеры параметров:

- intname="url", value="https://nextcloud.example.com"
- intname="verify_ssl", value="true"
- intname="timeout", value="30"

5. **processingcryptedparam** — Параметры обработчика (зашифрованные) (см. таблицу 6)

Зашифрованные учётные данные для доступа к Nextcloud.

Таблица 6 - Зашифрованные параметры обработчика

Поле	Тип	Назначение
id	INT	Первичный ключ
processingmodule	INT (FK)	Ссылка на обработчик
intname	VARCHAR	Внутреннее имя параметра
value	TEXT	Зашифрованное значение

Примеры параметров:

- intname="admin_username", value="<encrypted:admin>"
- intname="admin_password", value="<encrypted:secretpass>"

6. **item** — Услуги (основная таблица) (см. таблицу 7)

ГЛАВНАЯ ТАБЛИЦА: Каждая запись = одна активная услуга Nextcloud для клиента.

Таблица 7 - Услуги

Поле	Тип	Назначение
id	INT	Первичный ключ — ID услуги
pricelist	INT (FK)	Ссылка на тариф (определяет квоту)
account	INT (FK)	Ссылка на клиента
processingmodule	INT (FK)	Ссылка на обработчик (pmnextcloud)
status	INT	Статус услуги (1=заказан, 2=активен, 3=остановлен, 4=удален)
createdate	TIMESTAMP	Дата создания услуги
expiredate	DATE	Дата окончания подписки
suspenddate	TIMESTAMP	Дата приостановки (если приостановлена)
name	VARCHAR	Наименование услуги (например, "Nextcloud")

Поле	Тип	Назначение
		Storage 100GB")
note	TEXT	Комментарий
autosuspend	BOOLEAN	Автоматическая приостановка за неуплату

Связи:

- item.pricelist → pricelist.id (определяет квоту)
- item.account → account.id (клиент)
- item.processingmodule → processingmodule.id (обработчик)

Статусы:

- 1 = заказана (открытие в процессе)
- 2 = активна (готова к использованию)
- 3 = остановлена (приостановка)
- 4 = удалена (удаление)

7. pricelist — Тарифные планы (см. таблицу 8)

Определяют квоту и параметры услуги Nextcloud.

Таблица 8 - Тарифные планы

Поле	Тип	Назначение
id	INT	Первичный ключ
itemtype	INT (FK)	Тип услуги (должен быть "Nextcloud")
name	VARCHAR	Наименование тарифа (например, "Nextcloud 100GB")
intname	VARCHAR	Внутреннее имя
active	BOOLEAN	Доступность для заказа

Примеры:

- Тариф "Nextcloud 100GB" → Квота 100 GB
- Тариф "Nextcloud 1TB" → Квота 1000 GB

Параметры хранятся в: itemtypeparam (связанная таблица)

8. itemtype — Типы услуг (см. таблицу 9)

Категоризация услуг.

Таблица 9 - Типы услуг

Поле	Тип	Назначение
id	INT	Первичный ключ
name	VARCHAR	Наименование типа (например, "Nextcloud Storage")
intname	VARCHAR	Внутреннее имя
embedded	BOOLEAN	Системная ли услуга

9. **itemtypeparam** — Параметры типов услуг (см. таблицу 10)

Определяют какие параметры может иметь услуга (например, квота).

Таблица 10 - Параметры типов услуг

Поле	Тип	Назначение
id	INT	Первичный ключ
itemtype	INT (FK)	Ссылка на тип услуги
intname	VARCHAR	Внутреннее имя параметра (например, "quota_gb")
name	VARCHAR	Отображаемое имя (например, "Storage Quota (GB)")
inputtype	VARCHAR	Тип ввода (например, "integer", "string")

10. **itemstat** — Статистика использования (см. таблицу 11)

Сбор данных об использовании ресурсов (квоты).

Таблица 11 - Статистика использования

Поле	Тип	Назначение
item	INT (FK)	Ссылка на услугу
statdate	DATE	Дата сбора статистики
param	VARCHAR	Имя параметра (например, "used_bytes", "quota_bytes")
measure	INT (FK)	Единица измерения
value	BIGINT	Значение параметра

Примеры:

- param="used_bytes", value=536870912 (512 MB)
- param="quota_bytes", value=107374182400 (100 GB)

11. **subaccount** — Лицевые счета (для оплаты) (см. таблицу 12)

Информация о балансе и статусе подписки клиента.

Таблица 12 - Лицевые счета

Поле	Тип	Назначение
id	INT	Первичный ключ
account	INT (FK)	Ссылка на клиента
project	INT (FK)	Ссылка на провайдера (BILLmanager)
balance	DECIMAL	Баланс счёта
active	BOOLEAN	Активность счёта
currency	INT (FK)	Валюта счёта

Связь: Для определения статуса оплаты и возможности приостановки услуги.

2.3.3 Логическая модель

Логическая модель описанная выше в виде таблиц, представлена на рисунке 6.

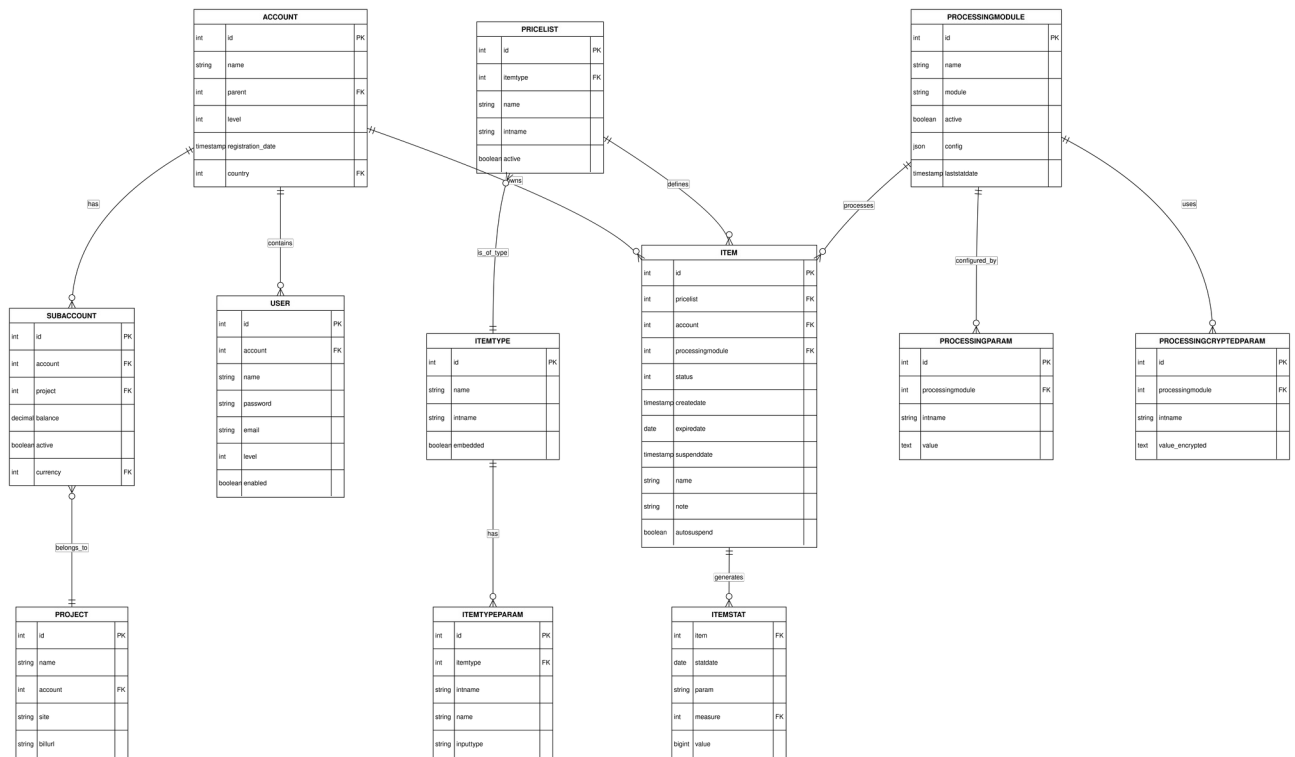


Рисунок 6 - Логическая модель БД

2.4 Проектирование пользовательского интерфейса

Модуль pmnextcloud использует стандартные формы BILLmanager. Все экраны относятся к двум разделам:

1. Административные
2. Пользовательские

2.4.1 Верхнеуровневое определение экранов

Раздел 1: Административные функции

Экраны для управления модулем интеграции и тарифными планами

- **1.1 Создание модуля** — конфигурация параметров интеграции (WebPay, Nextcloud)
- **1.2 Создание/изменение тарифа** — управление тарифными планами с указанием стоимости и размера хранилища

Раздел 2: Пользовательские функции

Экраны для клиентов, позволяющие приобрести услуги и просмотреть информацию

- **2.1 Покупка услуги** — выбор тарифа, оплата через WebPay, подтверждение
- **2.2 Просмотр информации об услуге** — отображение данных для входа и статуса услуги

2.4.2 Иерархия и переходы между экранами

Переходы в виде иерархии между экранами представлены на рисунке 7.

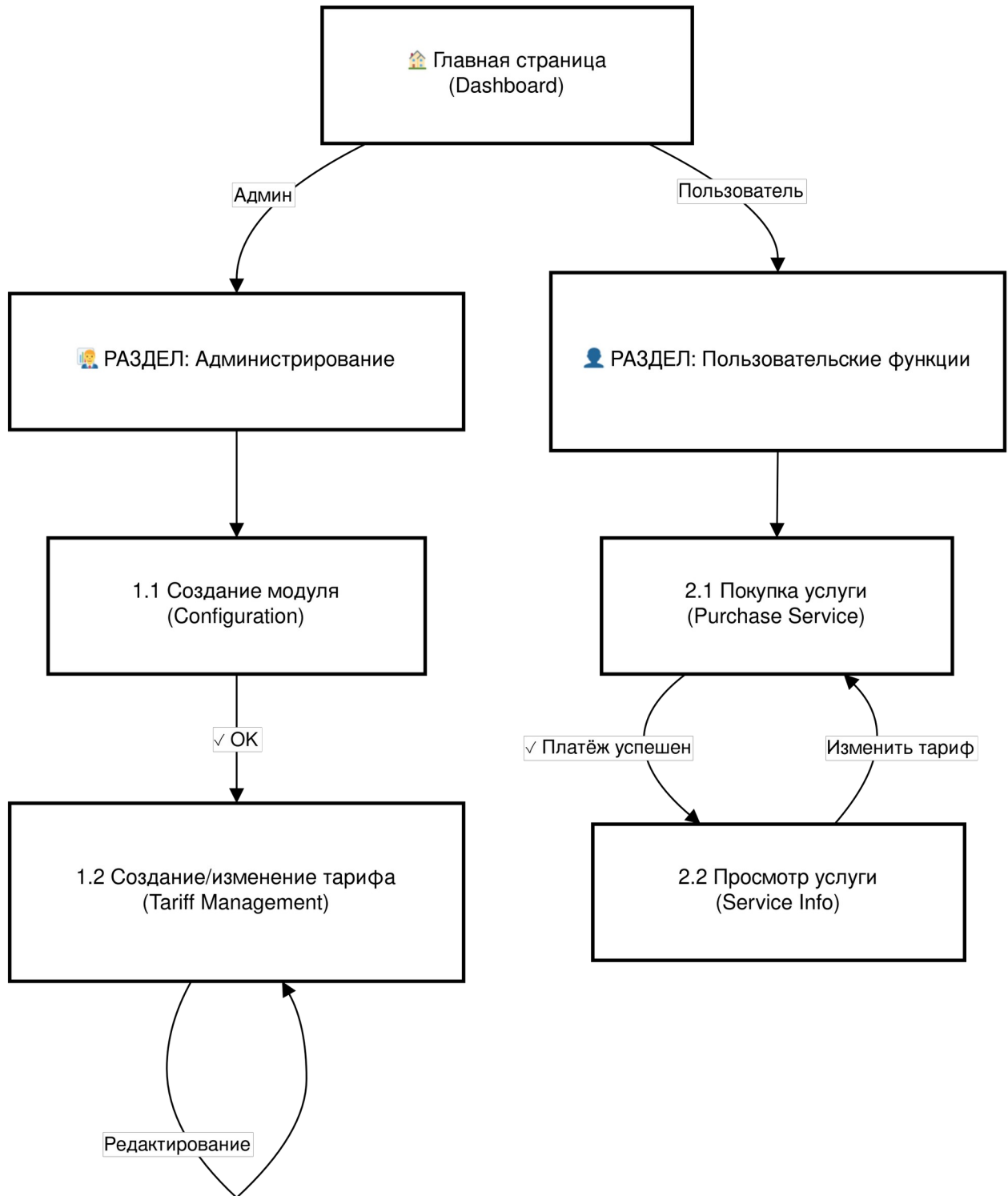


Рисунок 7 - Иерархия переход между экранами

2.4.3 Спецификация экранов

Спецификация экранов приложения можно увидеть в таблице 13.

Таблица 13 - Спецификация экранов

№	Название экрана	Назначение	Поля ввода	Валидация	Поведение
1.1	Создание модуля	Конфигурация параметров интеграции с WebPay и Nextcloud	<ul style="list-style-type: none"> • nextcloudURL (обязательно) • nc_username (обязательно) • nc_password(обязательно) 	<ul style="list-style-type: none"> • Формат URL (https://) • Непустые поля • Валидация подключения к сервисам • Длина: 1-255 символов 	При нажатии «Создать»: <ol style="list-style-type: none"> 1) Проверка полей 2) Валидация URL (HTTPS) 3) Проверка доступности сервисов 4) Сохранение в БД 5) Перенаправление на 1.2 или вывод сообщения об ошибке
1.2	Создание/изменение тарифа	Управление тарифными планами: name, price, storage, description	<ul style="list-style-type: none"> • Все основные поля предоставляются системой BILLmanager по административной настройке • Поле user_group в которой надо выбрать группу аккаунта по тарифу, группы подгружаются из панели 	На BILLmanager	При нажатии «Сохранить»: <ol style="list-style-type: none"> 1) Валидация полей 2) Проверка уникальности имени 3) Сохранение/обновление тарифа в БД 4) Отображение списка всех тарифов 5) Возможность редактирования/удаления существующих

№	Название экрана	Назначение	Поля ввода	Валидация	Поведение
2.1	Покупка услуги	Выбор тарифного плана, ввод данных покупателя и отправка на оплату WebPay	<ul style="list-style-type: none"> Все основные поля предоставляют ся системой BILLmanager по административ ной настройке 	На BILLmanager	При нажатии «Оплатить»: <ol style="list-style-type: none"> 1) Валидация данных 2) Создание Order в БД (status=pending) 3) Оплата услуги 4) Создание пользователя в Nextcloud → 5) Перенаправление на 2.2
2.2	Просмотр информации об услуге	Отображение текущего активного тарифа, использованно го хранилища, состояния платежей	<ul style="list-style-type: none"> Все основные поля предоставляют ся системой BILLmanager по административ ной настройке Поля username и password для входа в nextcloud (только для чтения) 	N/A (только чтение, данные из API Nextcloud и БД)	<ol style="list-style-type: none"> 1) Отображение данных пользователя

2.4.4 Состояния экранов

Диаграмма состояний для экрана 1.1 представлена на рисунке 8.

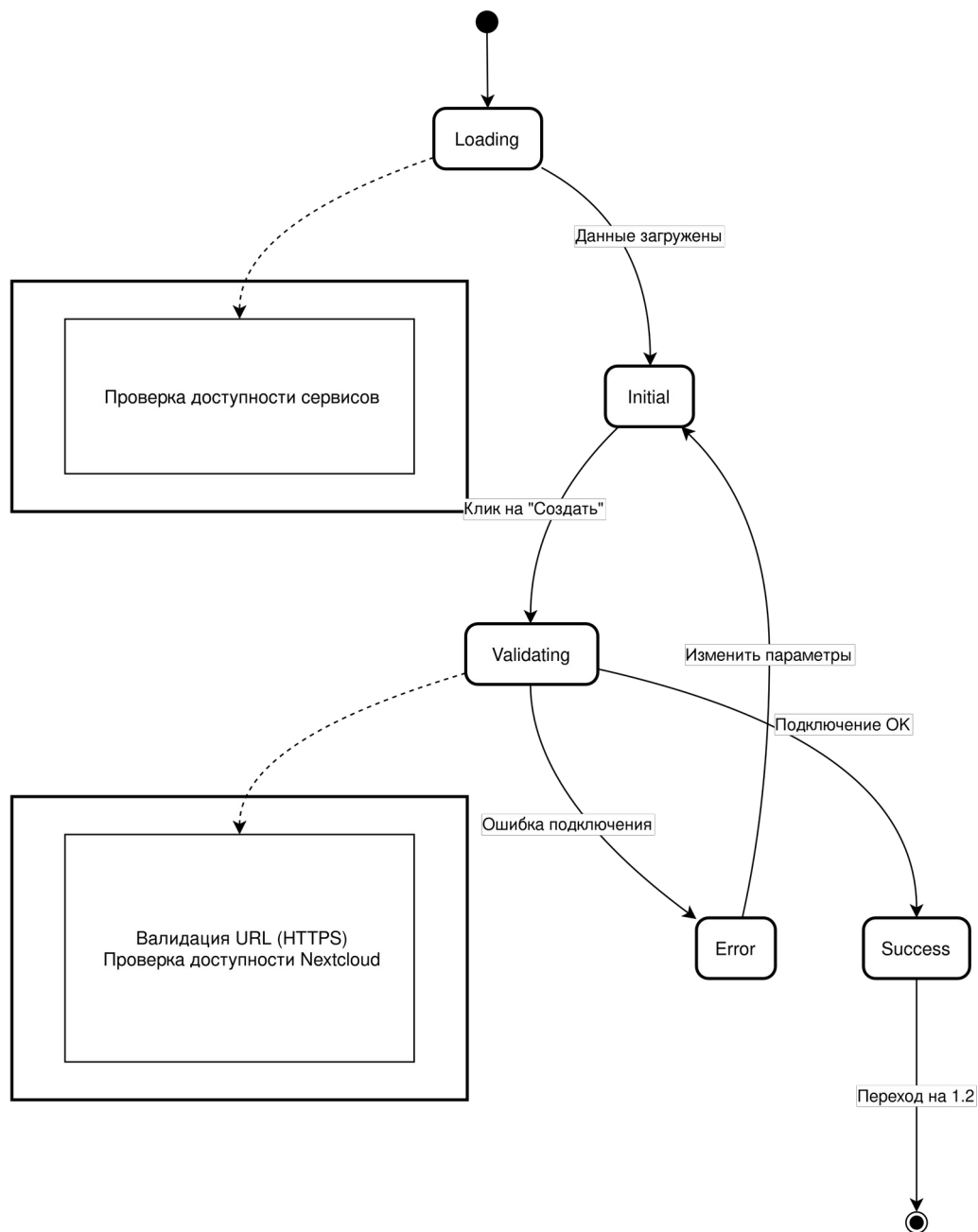


Рисунок 8 - Диаграмма состояний для экрана 1.1

Диаграмма состояний для экрана 1.2 отображена на рисунке 9.

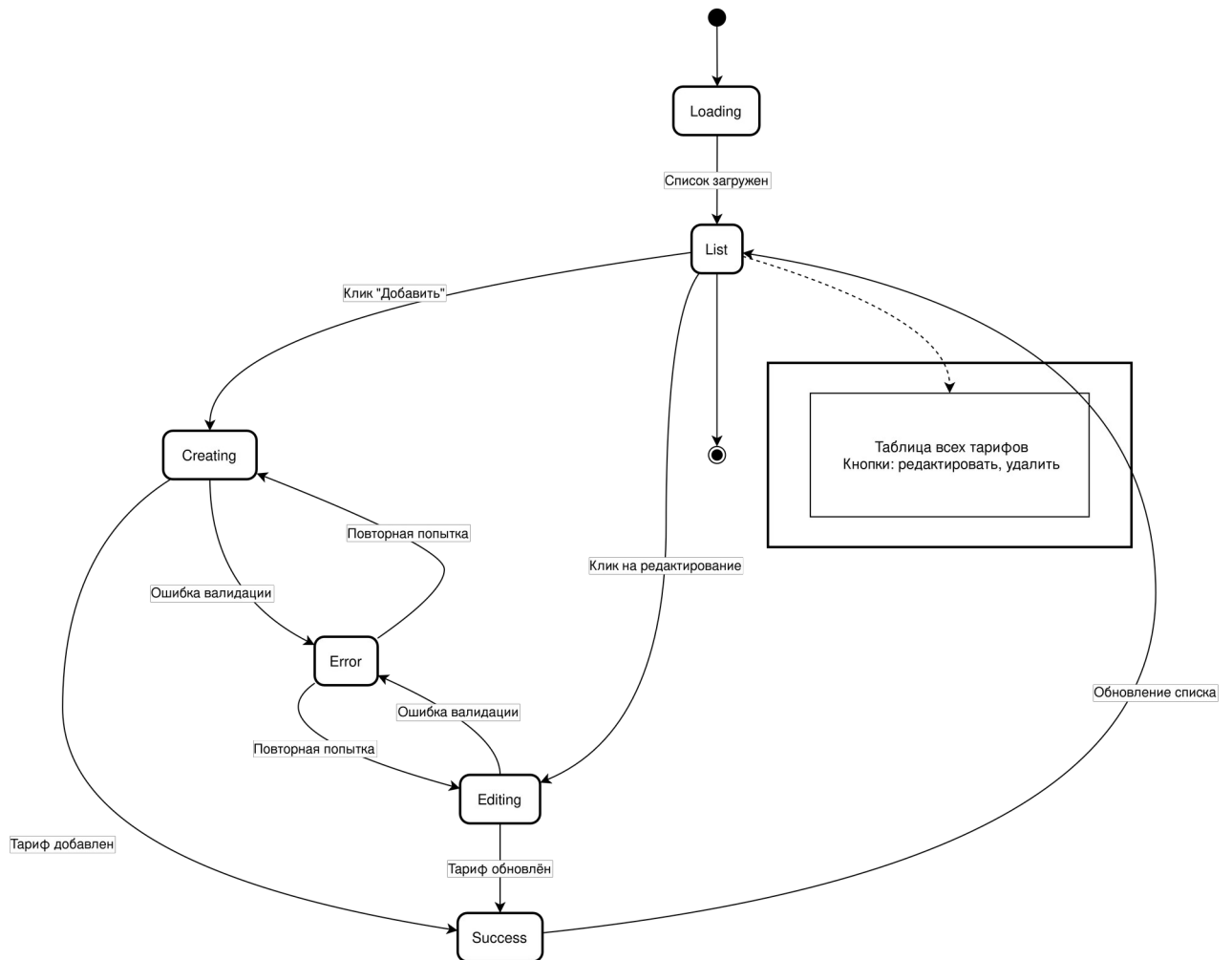


Рисунок 9 - Диаграмма состояний для экрана 1.2

Диаграмму состояний для экрана 2.1 можно увидеть на рисунке 10.

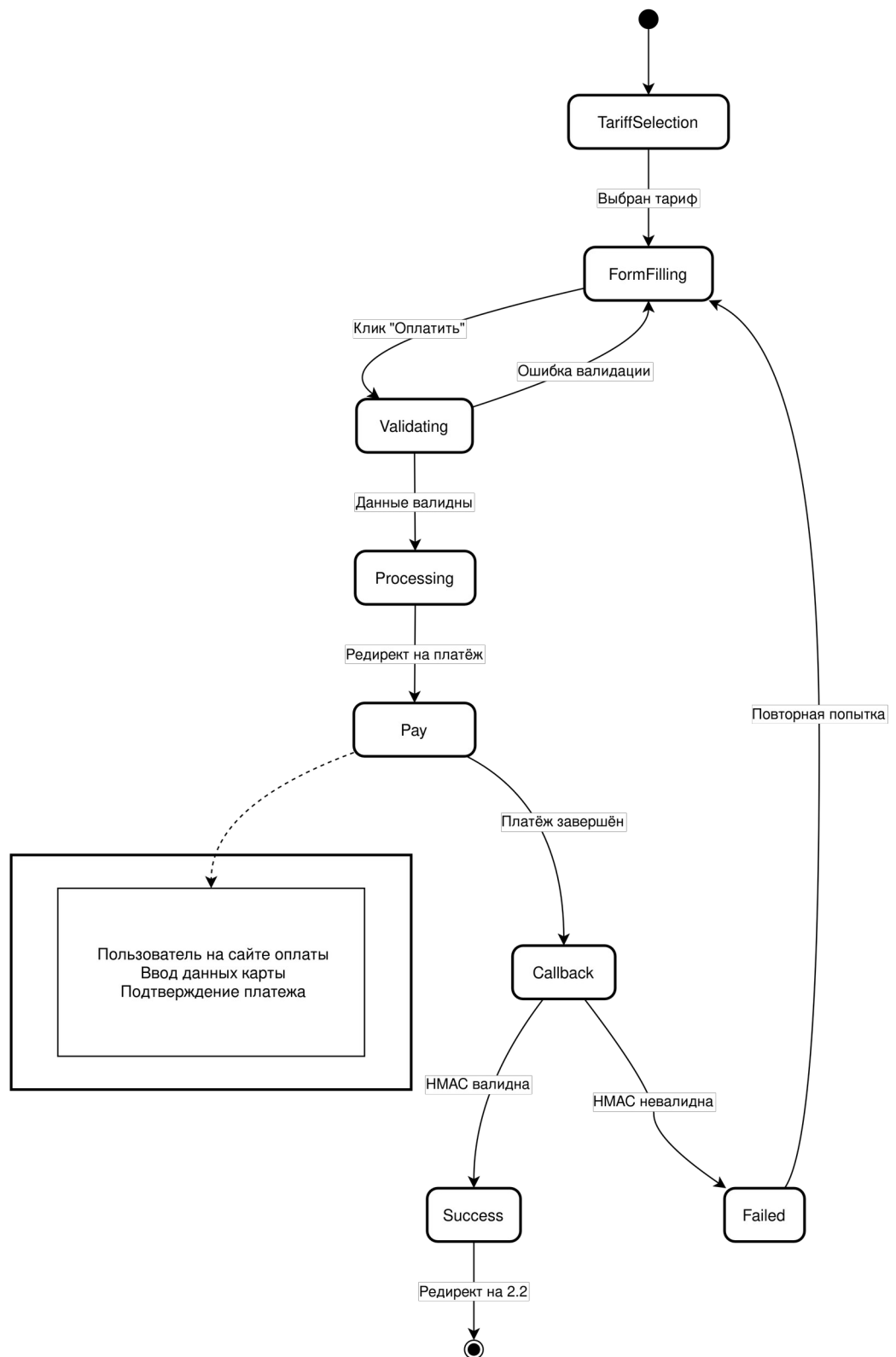


Рисунок 10 - Диаграмма состояний для экрана 2.1

Диаграмма состояний для экрана 2.2 отображена на рисунке 11.

2.4.5 Описание макетов экранов

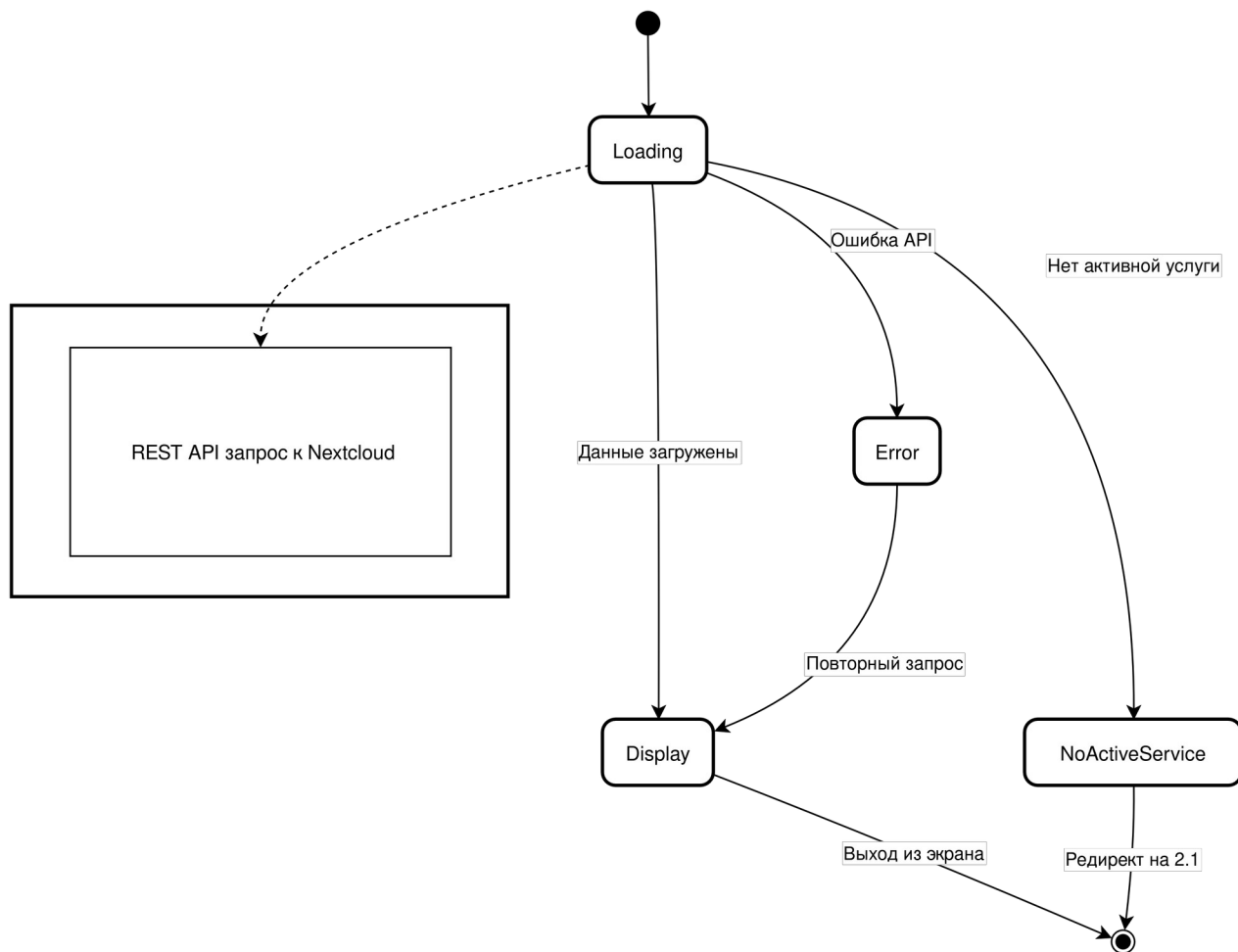


Рисунок 11 - Диаграмма состояний экрана 2.2
Макеты всех описанных выше экранов показаны на рисунках 12-16.

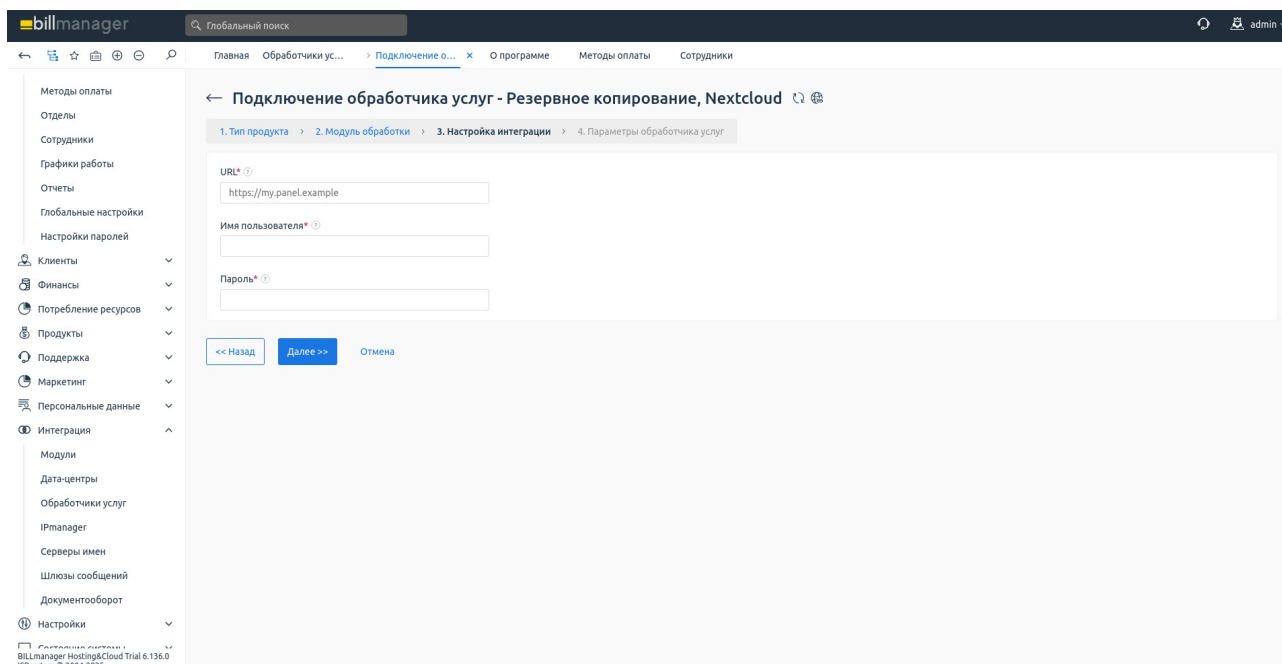


Рисунок 12 - Макет экрана создания модуля

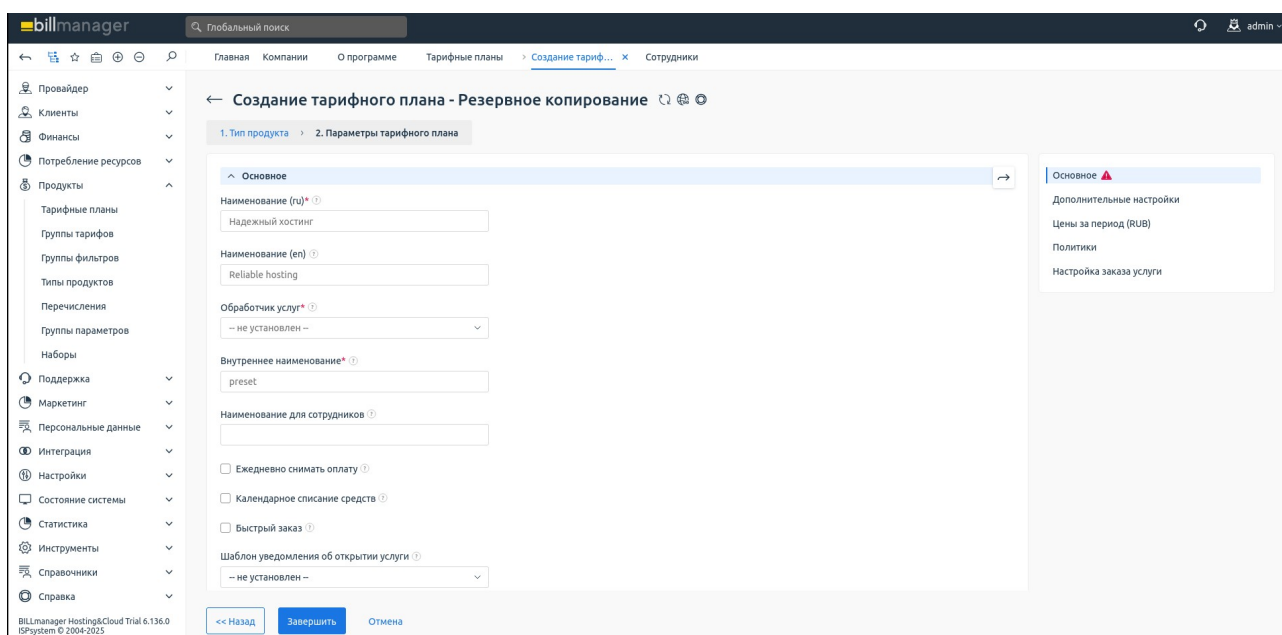


Рисунок 13 - Макет экрана создания тарифа

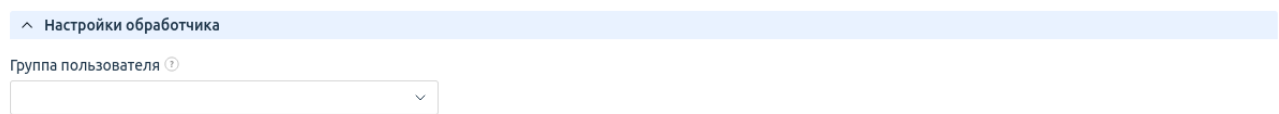


Рисунок 14 - Раздел настройки группы пользователя

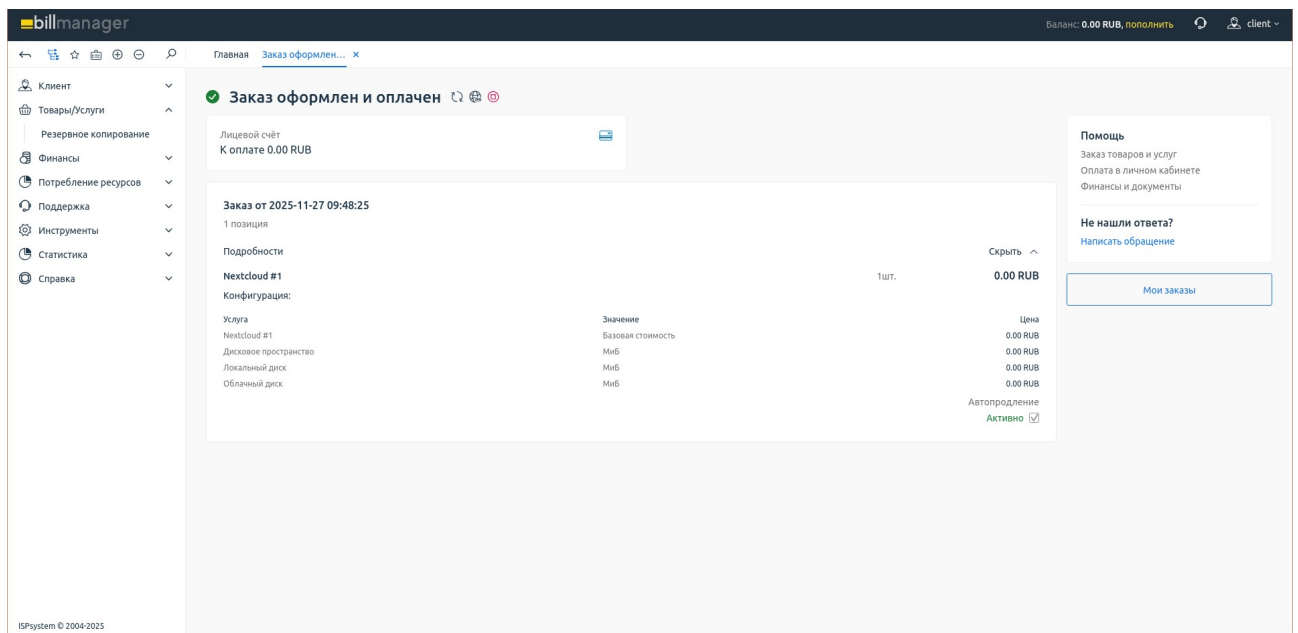


Рисунок 15 - Макет экрана покупки услуги

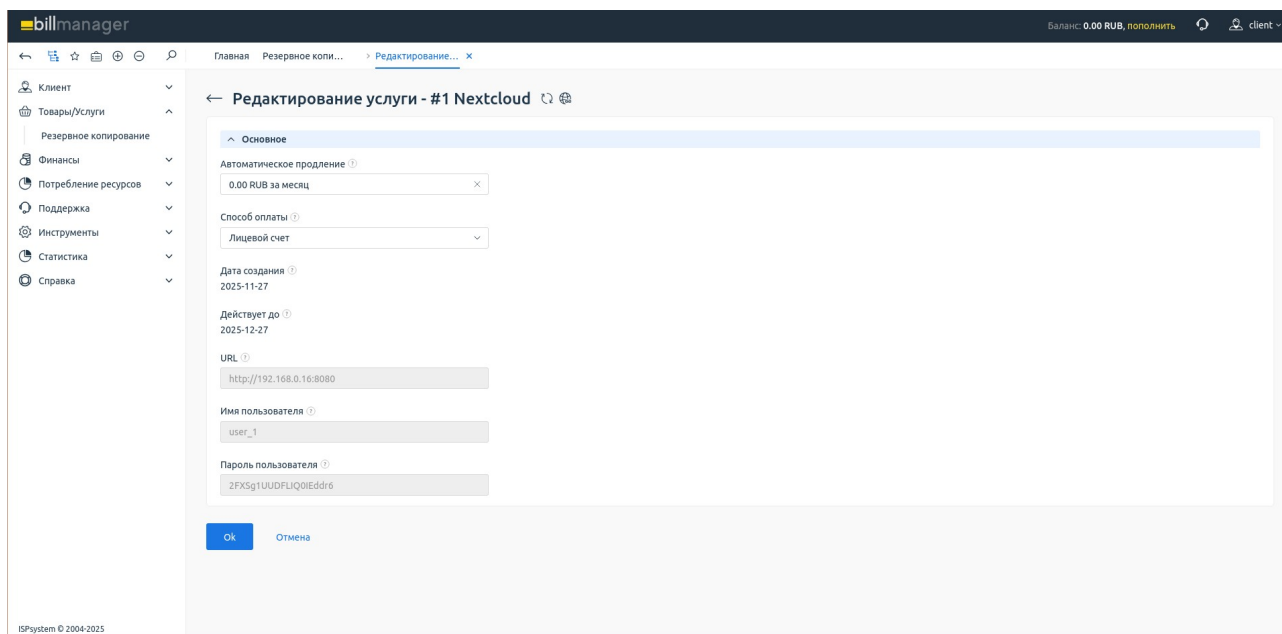


Рисунок 16 - Макет экрана просмотра данных заказа

2.4.6 Карта переходов между экранами

Полная карта переходов между экранами отображена н рисунке 17.

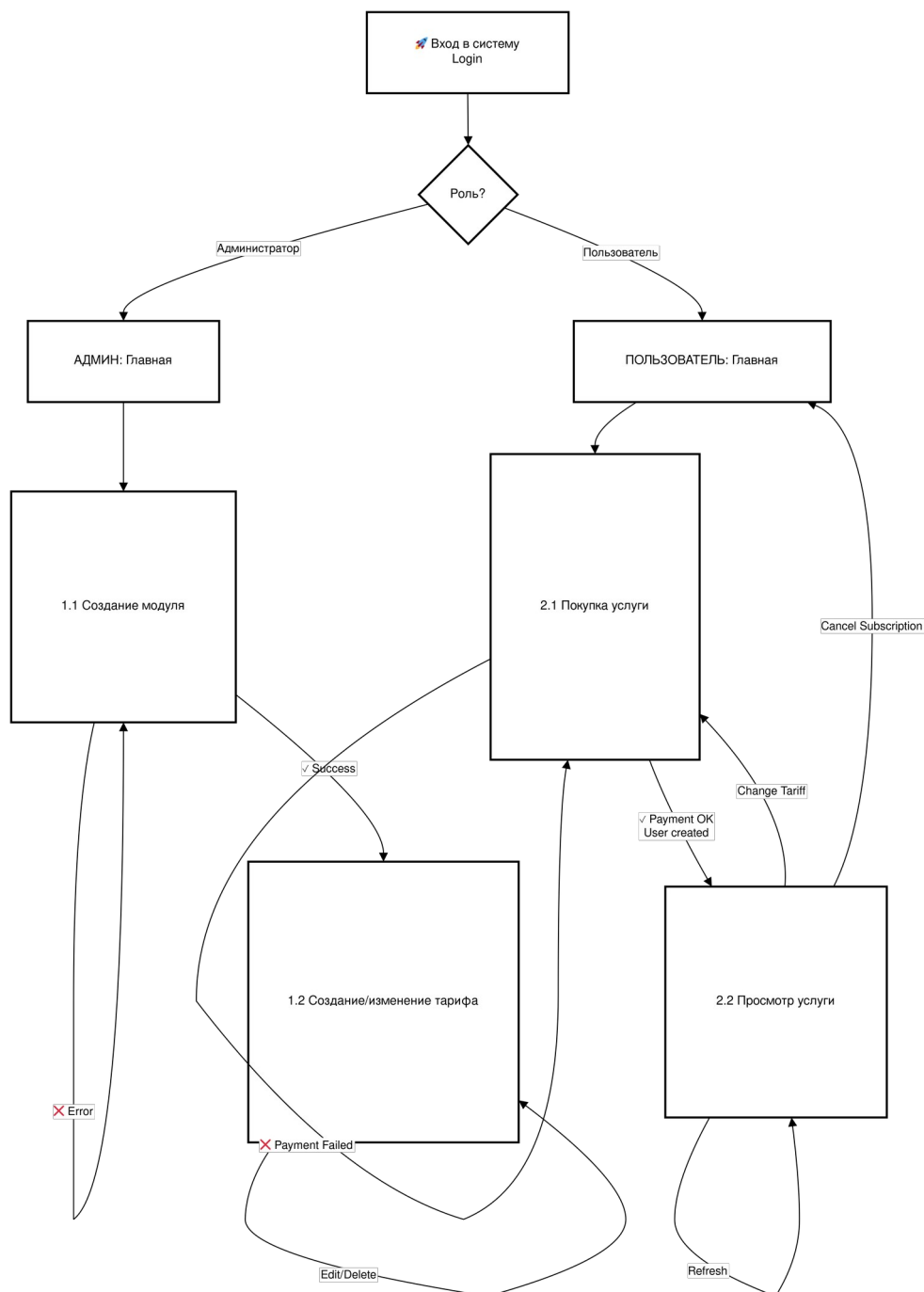


Рисунок 17 - Полная карта переходов между экранами

Заключение

Основные результаты работы

В процессе выполнения курсовой работы была проведена полная разработка программного модуля bill-nextcloud для интеграции биллинговой системы BILLmanager с облачными платформами Nextcloud и ownCloud. Результаты работы продемонстрировали, что поставленная цель достижима и актуальна для IT-провайдеров и хостинг-компаний.

Достигнутые задачи

1. Анализ предметной области и выявление проблемы

Проведён детальный анализ рынка IT-услуг, который показал:

- Тренд на переход от единоразовых покупок к модели SaaS
- Отсутствие готовых решений для интеграции BILLmanager с Nextcloud на рынке СНГ и Восточной Европы
- Наличие конкурентных решений (HostBill, WHMCS, Blesta), но только в своих экосистемах
- Значительный спрос на автоматизацию управления облачными услугами хранения

Выявленная проблема: биллинг-система BILLmanager и платформа Nextcloud/ownCloud существуют изолированно, требуя ручного вмешательства при продаже, изменении и управлении услугой. Это приводит к низкой операционной эффективности, риску ошибок и невозможности масштабирования.

2. Разработка архитектуры модуля

Спроектирована модульная архитектура, включающая:

- **Точка входа** (pmnextcloud.py) — регистрируется в BILLmanager как обработчик услуг
- **Обработчики команд** (open, close, suspend, resume, stat, set_param, check_connection) — каждый отвечает за определённый этап жизненного цикла услуги
- **REST API клиент** (NextcloudAPIClient) — взаимодействие с OCS Provisioning API Nextcloud через безопасные HTTPS соединения
- **Утилиты** — логирование, валидация, повторные попытки, управление ошибками

Архитектура спроектирована с учётом:

- Масштабируемости (поддержка нескольких серверов Nextcloud)
- Отказоустойчивости (механизм повторных попыток, логирование)

- Безопасности (шифрование учётных данных, HMAC-SHA1 для подписей, HTTPS соединения)

- Совместимости (работа с Python 3.8+, различные ОС)

3. Спецификация и моделирование бизнес-процессов

Использованы современные подходы моделирования:

- **BPMN 2.0** для описания бизнес-процессов (AS-IS → TO-BE)

- **UML диаграммы** для архитектуры компонентов

- **ER-диаграммы** для логической модели базы данных

- **Диаграммы состояний** (Mermaid) для описания переходов экранов

- **Sequence диаграммы** для взаимодействия между системами

Эти модели обеспечивают:

- Полное понимание всех процессов для техническим и деловым сторонам

- Ясность взаимодействия между BILLmanager, WebPay и Nextcloud

- Возможность автоматизированного тестирования и валидации

4. Разработка пользовательского интерфейса (Этап 4)

Спроектирована полная система из 4 экранов, организованных в 2 раздела:

Раздел 1: Административные функции

- Экран 1.1 — Создание модуля: конфигурация параметров подключения к WebPay и Nextcloud

- Экран 1.2 — Создание/изменение тарифа: управление тарифными планами с указанием стоимости и объёма хранилища

Раздел 2: Пользовательские функции

- Экран 2.1 — Покупка услуги: выбор тарифа, ввод данных, оплата через WebPay

- Экран 2.2 — Просмотр информации об услуге: отображение активного тарифа, использованного хранилища, статуса подписки

Для каждого экрана определены:

- Поля ввода и правила валидации

- Возможные состояния (Loading, Initial, Success, Error)

- Всплывающие уведомления (Alerts, Dialogs, Toasts)

- UI-контролы с идентификаторами для автоматизированного тестирования

- ASCII и Mermaid макеты

- Цветовая палитра и стили компонентов

5. Определение требований доступности и производительности

Обеспечено соответствие:

- **WCAG 2.1** уровень AA (контрастность, клавиатурная навигация, ARIA атрибуты)
- **Требования производительности:** Time to Interactive < 4 сек, API таймаут 10 сек
- **Адаптивность:** поддержка mobile-first дизайна, responsive breakpoints
- **Логирование и мониторинг:** интеграция с системой логирования BILLmanager, опциональная интеграция с Zabbix/Prometheus

6. Описание технической реализации

Определены:

- **Язык программирования:** Python 3.8+ (совместимость с BILLmanager)
- **Основные библиотеки:** requests, cryptography, json, xml.etree.ElementTree, hashlib
- **Инструменты:** Git, pytest, flake8, Docker
- **Таблицы БД:** описаны все сущности (account, item, pricelist, itemstat и др.)
- **Механизмы безопасности:** шифрование учётных данных, HMAC-SHA1 подписи, валидация SSL сертификатов

7. Анализ жизненного цикла услуги

Чётко определены все этапы:

1. **Активация** → создание пользователя в Nextcloud при оплате
2. **Изменение** → обновление квоты при апгрейде/даунгрейде тарифа
3. **Приостановка** → блокировка аккаунта при просрочке платежа
4. **Восстановление** → разблокировка при погашении задолженности
5. **Удаление** → удаление пользователя при окончательном удалении услуги

Каждый этап сопровождается логированием, обработкой ошибок и повторными попытками.

Практическая значимость и применение

Разработанное решение решает критическую проблему для IT-провайдеров:

Экономическая эффективность

- **Экономия времени:** Полная автоматизация процессов, которые ранее требовали ручного вмешательства
- **Снижение ошибок:** Исключение человеческого фактора при управлении аккаунтами

- **Масштабируемость:** Возможность обслуживать тысячи клиентов без увеличения штата

Улучшение клиентского опыта

- **Мгновенная активация:** Услуга предоставляется сразу после оплаты
- **Гибкость:** Клиент может самостоятельно выбирать, менять и отменять тарифы
- **Самообслуживание:** Минимизация взаимодействия с менеджерами через автоматизированный личный кабинет
- **Прозрачность:** Клиент видит текущее использование хранилища в реальном времени

Конкурентное преимущество

- **Уникальное предложение** на рынке СНГ и Восточной Европы
- **Соответствие тренду SaaS:** Предложение услуги по подписке с мгновенной активацией
- **Расширение портфеля услуг:** Возможность предлагать Nextcloud как готовый продукт, а не сложную техническую платформу

Список использованных источников

1. ISPSystem. Документация BILLmanager. Архитектура модулей и API [Электронный ресурс]. — Режим доступа: <https://www.ispsystem.ru/docs/bc/razrabotchiku/> (дата обращения: 24.11.2025).
2. ISPSystem. BILLmanager. Создание модулей обработки [Электронный ресурс]. — Режим доступа: <https://www.ispsystem.ru/docs/bc/razrabotchiku/sozдание-modulej/sozдание-modulej-obrabotki> (дата обращения: 24.11.2025).
3. ISPSystem. BILLmanager. Создание модулей платежных систем [Электронный ресурс]. — Режим доступа: <https://www.ispsystem.ru/docs/bc/razrabotchiku/sozдание-modulej/sozдание-modulej-plateyonyh-sistem> (дата обращения: 24.11.2025).
4. Nextcloud. Документация по REST API для управления пользователями [Электронный ресурс]. — Режим доступа: https://docs.nextcloud.com/server/latest/developer_manual/client_apis/OCS/ocs-api-overview.html (дата обращения: 24.11.2025).
5. Nextcloud. API для управления аккаунтами и квотами [Электронный ресурс]. — Режим доступа: https://docs.nextcloud.com/server/latest/developer_manual/client_apis/OCS/ocs-api-users.html (дата обращения: 24.11.2025).
6. GitHub. LC208/bill-nextcloud — модуль интеграции BILLmanager и Nextcloud [Электронный ресурс]. — Режим доступа: <https://github.com/LC208/bill-nextcloud/tree/main> (дата обращения: 24.11.2025).
7. RFC 2104. HMAC: Keyed-Hashing for Message Authentication [Электронный ресурс]. — Режим доступа: <https://tools.ietf.org/html/rfc2104> (дата обращения: 24.11.2025).
8. RFC 3875. The Common Gateway Interface (CGI) Version 1.1 [Электронный ресурс]. — Режим доступа: <https://tools.ietf.org/html/rfc3875> (дата обращения: 24.11.2025).
9. RFC 7230. Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing [Электронный ресурс]. — Режим доступа: <https://tools.ietf.org/html/rfc7230> (дата обращения: 24.11.2025).
10. RFC 5246. The TLS Protocol Version 1.2 [Электронный ресурс]. — Режим доступа: <https://tools.ietf.org/html/rfc5246> (дата обращения: 24.11.2025).
11. OWASP. Top 10 Web Application Security Risks [Электронный ресурс]. — Режим доступа: <https://owasp.org/www-project-top-ten/> (дата обращения: 24.11.2025).

12. Python Software Foundation. Python 3.8+ Documentation [Электронный ресурс]. — Режим доступа: <https://docs.python.org/3/> (дата обращения: 24.11.2025).
13. The Requests Library. Документация по библиотеке requests [Электронный ресурс]. — Режим доступа: <https://requests.readthedocs.io/> (дата обращения: 24.11.2025).
14. PyCA/cryptography. Документация по криптографической библиотеке [Электронный ресурс]. — Режим доступа: <https://cryptography.io/> (дата обращения: 24.11.2025).
15. MySQL Documentation. MySQL 5.7 Reference Manual [Электронный ресурс]. — Режим доступа: <https://dev.mysql.com/doc/refman/5.7/en/> (дата обращения: 24.11.2025).
16. PostgreSQL Documentation. PostgreSQL 10+ Documentation [Электронный ресурс]. — Режим доступа: <https://www.postgresql.org/docs/> (дата обращения: 22.11.2025).
17. HostBill Documentation. Nextcloud Module for HostBill [Электронный ресурс]. — Режим доступа: <https://hostbill.com/> (дата обращения: 24.11.2025).
18. WHMCS Documentation. Client Area and Module Development [Электронный ресурс]. — Режим доступа: <https://docs.whmcs.com/> (дата обращения: 24.11.2025).
19. Blesta Extensions. Nextcloud Provisioning Module [Электронный ресурс]. — Режим доступа: <https://marketplace.blesta.com/> (дата обращения: 24.11.2025).
20. WiseCP Extensions. Nextcloud Storage Module [Электронный ресурс]. — Режим доступа: <https://www.wisecp.com/> (дата обращения: 24.11.2025).
21. PuqCloud. Digital Product Nextcloud Integration [Электронный ресурс]. — Режим доступа: <https://puqcloud.com/> (дата обращения: 24.11.2025).
22. СТО 005–2020. Система менеджмента качества. Учебно-методическая деятельность. Оформление курсовых проектов (работ) и выпускных квалификационных работ технических направлений подготовки и специальностей [Электронный ресурс]. — Режим доступа: <https://www.istu.edu/local/modules/doc/download/41649> (дата обращения: 15.11.2025).