Общество с ограниченной ответственностью «Слива Групп»

ИНН 6678147202 · ОГРН 1269600025331

Программа для ЭВМ

«Rubble»

Веб-платформа автоматизации логистики вывоза отходов

ОПИСАНИЕ ПРОЦЕССОВ ЖИЗНЕННОГО ЦИКЛА

Правообладатель: ООО «Слива Групп»

г. Екатеринбург

2026

Содержание

1. Общие положения.....3
2. Проектирование.....3
3. Разработка.....3
4. Тестирование.....4
5. Поставка.....4
6. Эксплуатация.....4
7. Документирование.....5
8. Обучение персонала.....5
9. Поддержка версий и доработка.....5
10. Устранение сбоев.....6
11. Персонал разработчика.....6
12. Адреса и инфраструктура.....6
13. Служба поддержки.....7

1. Общие положения

Настоящий документ описывает процессы жизненного цикла программы для ЭВМ «Rubble» (далее — программа), применяемые правообладателем — ООО «Слива Групп»: проектирование, разработку, тестирование, поставку, эксплуатацию, документирование, обучение персонала, поддержку версий и доработку, устранение сбоев, а также сведения о персонале и инфраструктуре разработки и поддержки.

Программа разрабатывается правообладателем самостоятельно. Исходный код хранится в системе контроля версий Git на собственном сервере GitLab правообладателя (gitlab.sliva-group.ru); сборка, проверка и поставка выполняются средствами GitLab CI на серверах, расположенных на территории Российской Федерации.

2. Проектирование

– Сбор требований ведётся по обращениям заказчиков и пользователей, анализу отраслевых процессов обращения с отходами и требований законодательства к учёту и отчётности.

– Требования оформляются в виде задач (issues) в GitLab с указанием затрагиваемой части программы (серверная часть, веб-клиент, мобильное приложение).

– Значимые изменения планируются по методике OpenSpec: для каждого изменения готовятся предложение, проектное решение, спецификация требований и перечень задач; спецификации хранятся в репозитории вместе с кодом.

– Контракт взаимодействия клиентских приложений с серверной частью описывается в спецификации OpenAPI, которая формируется из кода серверной части и является единым источником истины для клиентов.

– Архитектурные решения фиксируются в репозитории и согласуются ведущими разработчиками.

3. Разработка

– Серверная часть: Go 1.27, Gin, PostgreSQL, Bun, миграции goose, объектное хранилище S3; исполняемый файл собирается командой go build.

– Веб-клиент: React 19, TypeScript в строгом режиме, Vite, RTK Query, Ant Design.

– Мобильное приложение водителя: Expo, React Native, TypeScript.

– Код ведётся в монорепозитории; каждое изменение выполняется в отдельной ветке и вливается в основную ветку через запрос на слияние (merge request) после рецензирования кода другим разработчиком и успешного прохождения автоматических проверок.

– Сообщения коммитов оформляются по соглашению Conventional Commits и проверяются автоматически; перед коммитом выполняются форматирование и статический анализ кода.

4. Тестирование

– Модульное тестирование серверной части и клиентских приложений.

– Интеграционное тестирование программного интерфейса серверной части с СУБД PostgreSQL, запускаемой в контейнерах (Testcontainers).

– Компонентное тестирование пользовательского интерфейса веб-клиента и мобильного приложения (Vitest, Jest, Testing Library).

– Проверка используемых сторонних библиотек на известные уязвимости по базе OSV.

– Статический анализ и проверка типов (линтеры, компилятор TypeScript, go vet).

– Все проверки запускаются автоматически в GitLab CI для каждого запроса на слияние; изменение не может быть влито при непрошедших проверках.

– Перед выпуском в эксплуатацию версия разворачивается на тестовом стенде и проверяется вручную по ключевым сценариям.

5. Поставка

– Программа поставляется по модели SaaS: пользователь получает доступ через веб-браузер по адресу https://logistics.sliva-group.ru по учётной записи, созданной администратором.

– Компоненты собираются в GitLab CI в контейнерные образы и публикуются в собственном реестре образов правообладателя registry.sliva-group.ru.

– Развёртывание выполняется средствами Docker Compose: тестовый стенд обновляется автоматически, выпуск в промышленную эксплуатацию инициирует ответственный сотрудник правообладателя.

– Мобильное приложение водителя собирается для Android и iOS и распространяется: [заполнить: канал распространения мобильного приложения].

– По отдельному договору программа может быть развёрнута на инфраструктуре заказчика по документу «Инструкция по установке».

6. Эксплуатация

– Серверная часть работает в контейнерах на серверах правообладателя; защищённое соединение HTTPS обеспечивается обратным прокси Traefik с сертификатами Let's Encrypt.

– Мониторинг ресурсов и состояния сервисов — Prometheus, node-exporter, Grafana с настроенными оповещениями; централизованный сбор журналов — Loki и Alloy; сбор ошибок клиентских приложений — Bugsink.

– Резервное копирование базы данных выполняется ежечасно с шифрованием и выгрузкой в отдельное объектное хранилище.

– Учётные записи пользователей создаются администратором; права доступа разграничиваются по ролям и организациям.

7. Документирование

- Пользовательская документация публикуется на сайте программы https://landing.logistics.sliva-group.ru/documentation и обновляется при выпуске версий, затрагивающих работу пользователей.
- Техническая документация (описание архитектуры, правила разработки, спецификации OpenSpec, спецификация OpenAPI, инструкции по развёртыванию) хранится в репозитории вместе с кодом и изменяется в тех же запросах на слияние, что и код.
- Документация программного интерфейса доступна в работающем экземпляре по адресу /swagger.

8. Обучение персонала

- Обучение пользователей заказчика: дистанционные демонстрации и консультации по ролям (отходообразователь, регулятор, перевозчик, водитель, утилизатор), пользовательская документация на сайте программы, консультации службы поддержки.
- Обучение персонала правообладателя: вводный курс для новых разработчиков по архитектуре и правилам разработки, закреплённым в репозитории, наставничество и рецензирование кода.

9. Поддержка версий и доработка

- Версии программы нумеруются по схеме семантического версионирования X.Y.Z; выпуск фиксируется тегом в системе контроля версий.
- Доработки выполняются по обращениям заказчиков и плану развития, проходят полный цикл: проектирование, разработка, тестирование, поставка.
- Поскольку программа предоставляется как сервис, обновления применяются централизованно и становятся доступны всем пользователям без действий с их стороны.
- Изменения схемы базы данных выполняются версионируемыми миграциями, применяемыми автоматически при обновлении.

10. Устранение сбоев

1) Сбой выявляется средствами мониторинга и оповещения либо по обращению пользователя в службу поддержки.
2) Обращение регистрируется как задача в GitLab, ему присваивается приоритет.
3) Ответственный специалист анализирует журналы, метрики и отчёты об ошибках, локализует причину.
4) Исправление разрабатывается и проходит рецензирование и автоматические проверки в обычном порядке; при критическом сбое — в приоритетном порядке.
5) Исправленная версия разворачивается; при необходимости выполняется откат к предыдущей версии образов или восстановление данных из резервной копии.

6) Пользователь, сообщивший о сбое, информируется о результате.

11. Персонал разработчика

Параметр	Сведения
Общая численность персонала, участвующего в разработке и поддержке	[заполнить: численность, чел.]
Разработчики серверной части	[заполнить: численность и квалификация]
Разработчики веб-клиента и мобильного приложения	[заполнить: численность и квалификация]
Тестирование	[заполнить: численность и квалификация]
Специалисты службы поддержки	[заполнить: численность и квалификация]
Руководитель разработки	[заполнить: ФИО, квалификация]

Требуемая квалификация разработчиков: высшее профессиональное образование в области информационных технологий, знание языков Go и TypeScript, опыт работы с PostgreSQL, Docker, системой контроля версий Git.

12. Адреса и инфраструктура

Объект	Адрес
Юридический адрес правообладателя	620091, Свердловская область, г. Екатеринбург, Полимерный пер., д. 9, кв. 482
Фактический адрес места нахождения разработчиков	[заполнить: фактический почтовый адрес разработчиков]
Фактический адрес службы поддержки	[заполнить: фактический почтовый адрес службы поддержки]
Инфраструктура разработки (GitLab gitlab.sliva-group.ru, GitLab Runner, реестр образов registry.sliva-group.ru)	[заполнить: наименование провайдера и почтовый адрес дата-центра]
Инфраструктура эксплуатации (серверы экземпляра https://logistics.sliva-group.ru)	[заполнить: наименование провайдера и почтовый адрес дата-центра]

Вся инфраструктура разработки и эксплуатации программы размещена на территории Российской Федерации.

13. Служба поддержки

Параметр	Значение
Электронная почта	contact@sliva-group.ru
Телефон	[заполнить: телефон службы поддержки]
Сайт	https://sliva-group.ru, https://landing.logistics.sliva-group.ru
Режим работы	[заполнить: дни и часы работы службы поддержки, часовой пояс]

Параметр	Значение
Срок реакции на обращение	[заполнить: срок первичного ответа]

Служба поддержки принимает обращения по вопросам работы с программой, сообщения о сбоях и предложения по доработке, консультирует пользователей и передаёт обращения разработчикам.