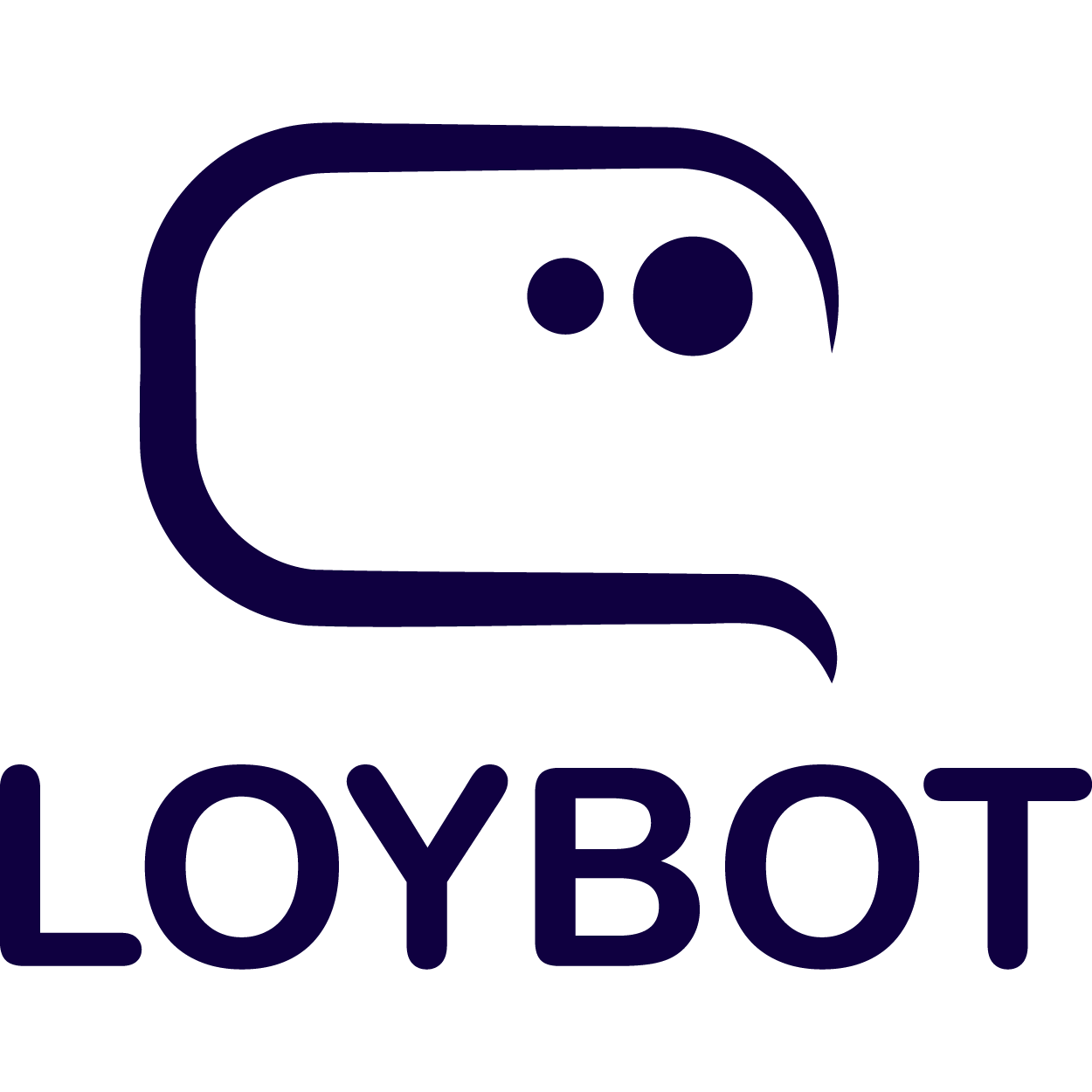
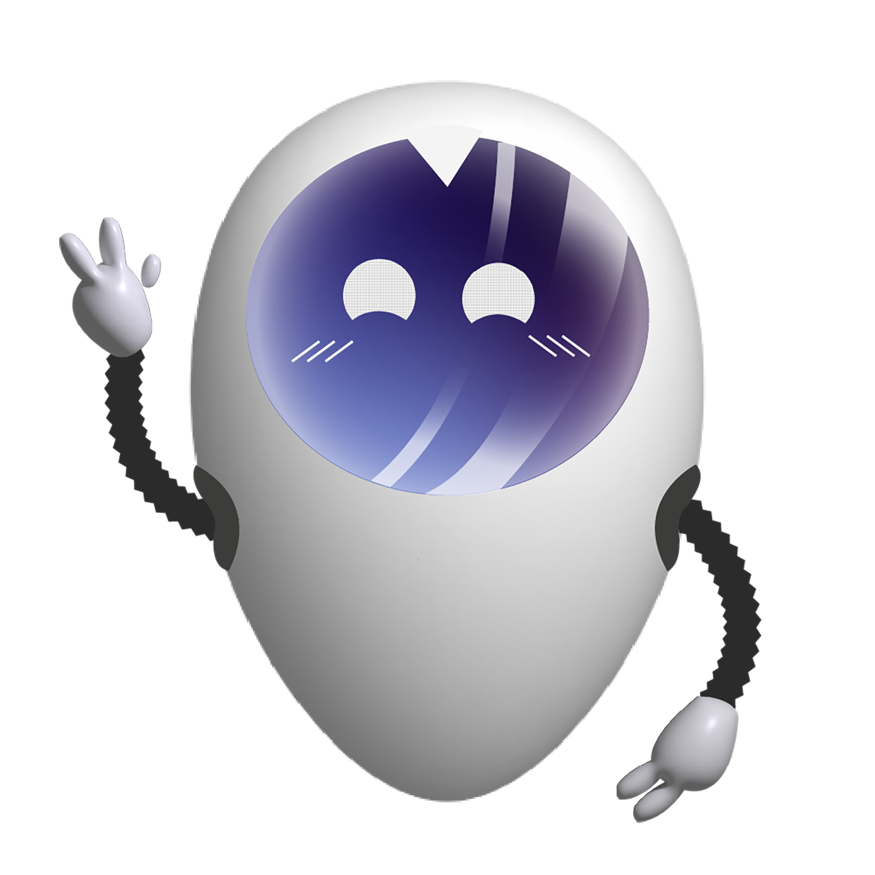
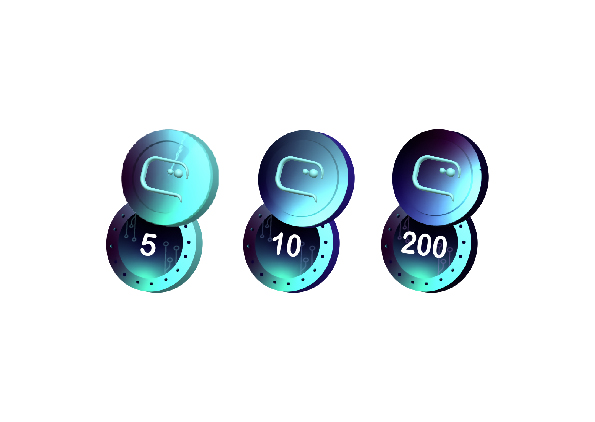
****

**ПРЕЗЕНТАЦИЯ ПРОЕКТА LoyBOT**



Гаркавый Алексей Павлович

Козлов Александр



СОДЕРЖАНИЕ

[Введение 2](#_Toc146275147)

[Процесс разработки (проблемы, решения, черновики, использованные технологии и инструменты). 2](#_Toc146275148)

[Предполагаемая итоговая стоимость разработки проекта. 3](#_Toc146275149)

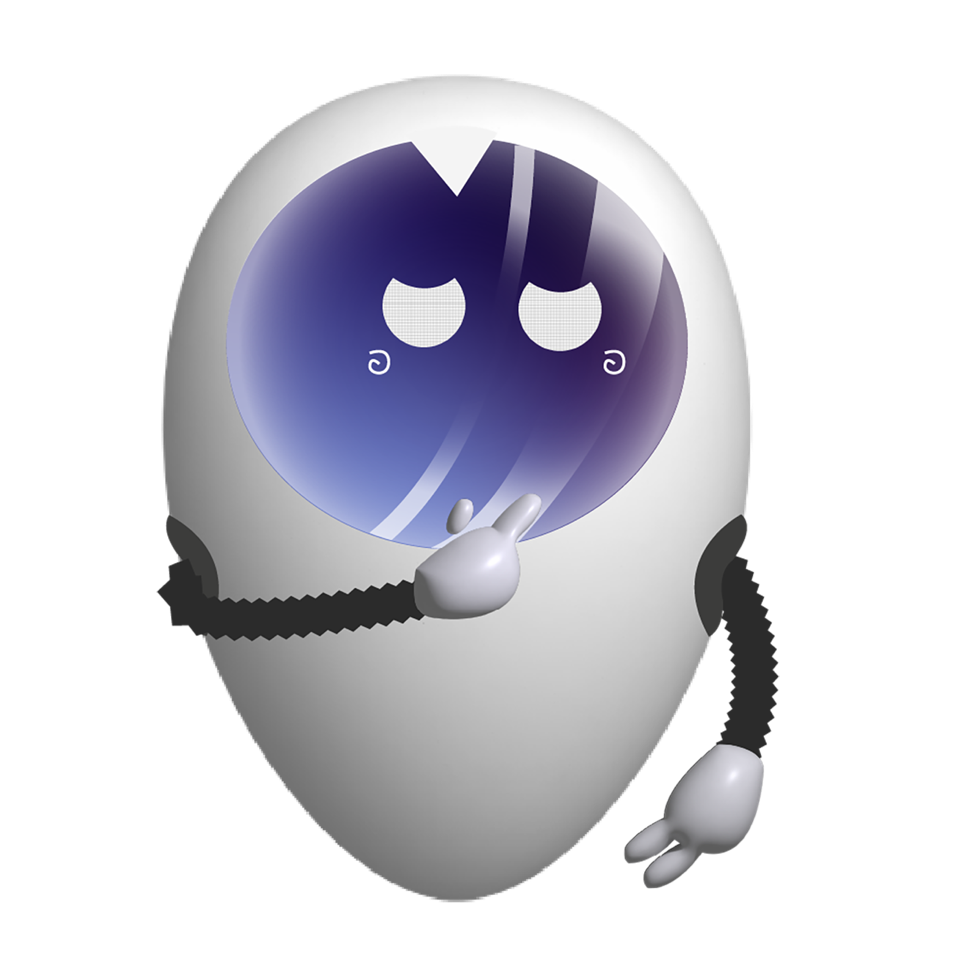
[Предполагаемая стоимость публикации и поддержки проекта в течении года. 3](#_Toc146275150)

[Список используемых библиотек и готовых решений. 3](#_Toc146275151)

[Лицензионная чистота проекта. 4](#_Toc146275152)

[Способность проекта к масштабированию. 4](#_Toc146275153)

[Рекомендации по развертыванию и использованию проекта. 5](#_Toc146275154)

****

# Введение

Целью составления данного документа является необходимость разработки системы учета использования и управления услугой клиентами компании с возможностью предоставления доступа к такой услуге по подписке.

# Процесс разработки (проблемы, решения, черновики, использованные технологии и инструменты).



1. Необходимо разработать и сверстать пользовательский интерфейс личного кабинета клиента и основного окна чата с ботом с учетом функционала системы.

Для реализации данной задачи были использованы следующие технологии:

HTML, CSS, SCSS, WebStorm.

1. Необходимо разработать веб-сервис с самостоятельно спроектированным программном интерфейсом (API), предоставляющий доступ к статистике и управлению услугой от имени клиента.

Для реализации данной задачи были использованы следующие технологии:

Java, Spring Boot, nginix, IntelliJ IDEA.

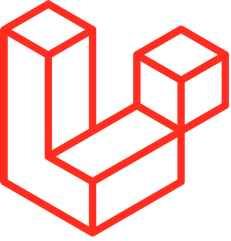


1. Необходимо разработать одностраничный информационный сайт компании, предоставляющий услугу доступа к боту по подписке.

Для реализации данной задачи были использованы следующие технологии:

OpenServer, WordPress плагин (Elementor), HTML, CSS, SCSS, WebStorm, MySQL.

1. Необходимо разработать монолитное веб-приложение для администраторов и менеджеров системы для организации работы с клиентами и управления выделяемыми ресурсами и их параметрами: биллинг, реестр клиентов, управление контейнерами и др.

Для реализации данной задачи были использованы следующие технологии:

Laravel, PHPStorm, PostgreSQL, PHP.

1. Необходимо разработать одностраничное веб-приложение (SPA), предоставляющее клиенту доступ в личный кабинет для получения статистики и управления услугой на основании разработанного ранее HTML макета (модуль A). Взаимодействие с сервером необходимо осуществлять посредством разработанного ранее API (модуль B). Для выполнения задания используйте API и верстку разработанную самостоятельно ранее.

Для реализации данной задачи были использованы следующие технологии:

Angular, WebStorm, PostgreSQL.

# Предполагаемая итоговая стоимость разработки проекта.

Оценочная стоимость данного проекта: 3000$

# Предполагаемая стоимость публикации и поддержки проекта в течении года.

Поддержка и сопровождение данного проекта оценивается в 1000$ в год.

# Список используемых библиотек и готовых решений.

**Angular:**

Material2 - это официальная библиотека компонентов Angular, в которой реализованы концепции Material Design от Google. Эти компоненты пользовательского интерфейса можно рассматривать как примеры кода, написанные в соответствии с рекомендациями команды разработчиков Angular.

NG-ZORRO - компоненты из библиотеки NG-ZORRO написаны на TypeScript с полной типизацией. Цель этого проекта - дать разработчикам высококлассные компоненты для создания пользовательских интерфейсов в стиле Ant Design.

**Laravel:**

Kotakin - превращает веб-приложение на основе Laravel в решение для обмена файлами, которое позволяет легко обмениваться файлами и папками с другими.

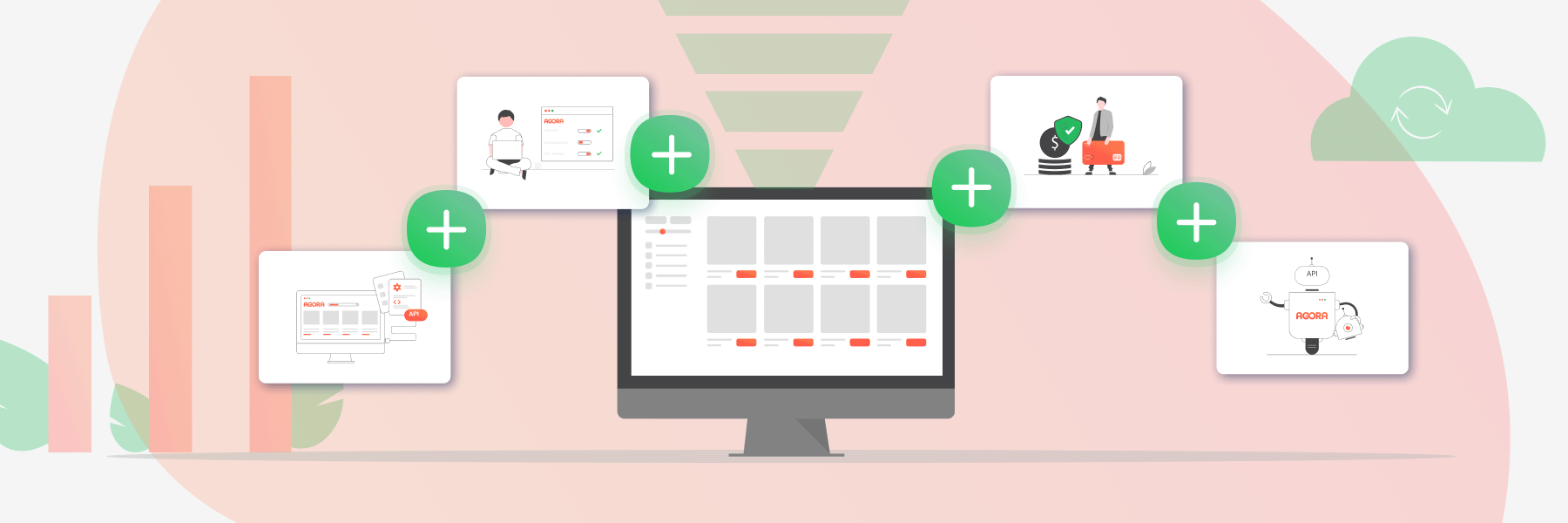
API RESTful - возможность генерировать ключи API для клиентских приложений, встроенные функции CRUD, а также основные функции GET, PUT, POST, DELETE и базовую проверку подлинности.

# Лицензионная чистота проекта.

Сегодня вопрос использования пиратского программного обеспечения стоит очень остро. И дело не в том, что это просто неэтично, но в большинстве случаев очень опасно, поэтому данной программный продукт не содержит лицензионных ограничений и является абсолютно независимым в плане лицензий.

# Способность проекта к масштабированию.

Понимание типов масштабируемости имеет решающее значение для создания программного обеспечения, которое может расти вместе с потребностями рынка. Существует два основных типа масштабируемости: вертикальная и горизонтальная.

Данный программный продукт готов к вертикальной и горизонтальной масштабируемости. Вертикальная масштабируемость может быть достигнута благодаря обновлению серверного оборудования, например добавление оперативной памяти, более быстрых процессоров или твердотельных накопителей и оптимизация программного обеспечения для более эффективного использования системных ресурсов, например повышение производительности запросов или реализация стратегий кэширования. Горизонтальная масштабируемость может быть достигнута благодаря добавлению физических или виртуальных машин, которые работают параллельно для увеличения вычислительной мощности и емкости.

# Рекомендации по развертыванию и использованию проекта.

В исходниках проекта предоставлены все инструкции по развертыванию проекта на удаленном сервере.