

Используется

Техническое задание NekoShop

Разработка онлайн магазина в мессенджере Telegram с
использованием Python

4-proxy
21.08.2024

Содержание

Оглавление

Содержание	1
1. Особенности терминологии и проектных решений	3
1.1. Особое положение по терминам:	3
1.2. Особое положение по проектным решениям:	3
1.3. Общее положение для технического задания проекта:	4
2. Общие сведения	5
2.1. Полное наименование проекта и его условное обозначение:	5
2.2. Наименование разработчика и сведения о нём при их наличии:	5
2.3. Нормативно-технические документы, регламентирующие создание продукта:	5
3. Цели и назначение проекта	6
3.1. Цели создания проекта:	6
3.2. Назначение проекта:	6
4. Характеристика объекта автоматизации	7
4.1. Основные сведения об объекте автоматизации:	7
4.2. Условия эксплуатации объекта автоматизации:	7
4.3. Характеристика окружающей среды:	7
5. Требования к проекту	8
5.1. Требования к компонентам структуры базы данных:	8
5.1.1. Требования к структуре таблиц базы данных:	8
5.1.2. Требования к назначению таблиц базы данных:	22
5.1.3. Визуальное представление схемы базы данных:	27
5.2. Требования к функциональным характеристикам:	27
5.2.1. Функциональные требования к интерфейсу клавиатур в Telegram:	27
5.2.1.1. Функциональные требования на уровне обычного пользователя:	27
5.2.1.2. Функциональные требования на уровне продавца:	31
5.2.1.3. Функциональные требования на уровне администратора:	33
5.2.2. Иные функциональные требования:	38
5.3. Требования к информационной и программной совместимости:	39
5.3.1. Требования к языкам программирования и программным средствам:	39
5.3.1.1. Требования к языкам программирования:	39
5.3.1.2. Требования к программным средствам:	39
5.3.2. Требования к исходному коду:	39
5.3.3. Требования к элементам интерфейса в Telegram:	40

5.3.4.1. Вид клавиатур для обычного пользователя:.....	40
5.3.4.2. Вид клавиатур для продавца:	41
5.3.4.3. Вид клавиатур для администратора:	41
6. Стадии и этапы разработки.....	44
Приложения.....	46
1. Стандарт документирования	46
2. Семантическое версионирование	46
3. ГОСТ 34.602-2020	47
4. ГОСТ 19.201-78.....	47

1. Особенности терминологии и проектных решений

1.1. Особое положение по терминам:

- Проект – Совокупность программного кода, документации, баз данных, изображений и других файлов, включённых в процесс разработки;
- Система – Под системой подразумевается набор программного кода, встроенных алгоритмов и рабочих сценариев, которые обеспечивают функциональность и работоспособность продукта;
- Продукт – Конечный объект разработки, программа, представляющая собой Telegram магазин;
- Администратор – Лицо, влияющее на структуру и работу продукта или же участвующее в разработке проекта;
- Продавец – Лицо, имеющее права на *CRUD*¹ операции для собственного товара в каталоге продукта;
- Пользователь – Базовый клиент конечного продукта, то есть пользователь Telegram (покупатель);

1.2. Особое положение по проектным решениям:

1. Для указания версии проекта и отдельных пакетов/модулей будет использоваться *семантическое версионирование*² (подробнее в разделе “Приложения” пункт: [2. Семантическое версионирование](#));

¹ **CRUD** - создание (*create*), чтение (*read*), модификация (*update*), удаление (*delete*).

² **Семантическое версионирование** (*Semantic Versioning*) - спецификация о том, как присваивать версии релизам программного обеспечения.

2. При работе с проектной документацией в Microsoft Word будет применяться документ “Стандарт документирования - Рекомендации по форматированию документации проектов в Microsoft Word”. (подробнее в разделе “Приложения” пункт: [1. Стандарт документирования](#));

3. Структура настоящего технического задания для проекта “NekoShop” основана на рекомендациях из “ГОСТ 34.602-2020” и “ГОСТ 19.201-78”. (подробнее в разделе “Приложения” пункт: [3. ГОСТ 34.602-2020](#), [4. ГОСТ 19.201-78](#));

4. Приоритет в оформлении настоящего технического задания отдаётся документу “Стандарт документирования”. (подробнее в разделе “Приложения” пункт: [1. Стандарт документирования](#)).

То есть в случае конфликтов с другими документами, касательно оформления элементов содержания предпочтение отдается “Стандарт документирования”;

1.3. Общее положение для технического задания проекта:

Поскольку проект "NekoShop" является индивидуальной и некоммерческой разработкой, техническое задание будет разрабатываться с целью общего проектирования архитектуры системы, создания дизайна и установления ясных намерений для разработчика/программиста.

Это означает, что содержание технического задания будет ограничено определёнными рамками, которые не будут избыточными и будут относиться к области разработки.

Так же структура и содержание настоящего технического задания может быть подвержено изменению в процессе разработки проекта;

2. Общие сведения

2.1. Полное наименование проекта и его условное обозначение:

Наименование проекта: “Разработка онлайн магазина в мессенджере Telegram с использованием Python”;

Условное обозначение: “NekoShop”;

2.2. Наименование разработчика и сведения о нём при их наличии:

Разработчик: 4proxy (4-proxy);

Способы связи:

- <https://github.com/4-proxy> (GitHub);
- 4proxy.cleanhouse@proton.me (Public Email);

2.3. Нормативно-технические документы, регламентирующие создание продукта:

Внешние НТД:

- [ГОСТ 34.602-2020](#);
- [ГОСТ 19.201-78](#);

Внутренние НТД:

- [Стандарт документирования](#);
- [Семантическое версионирование](#);

3. Цели и назначение проекта

3.1. Цели создания проекта:

Создание удобного интерфейса для пользователей: разработка удобного и интуитивно понятного интерфейса для пользователей, с целью обеспечения комфортного взаимодействия с конечным продуктом;

Обеспечение безопасности платежей: гарантирование безопасности платежей, что позволит пользователям совершать покупки онлайн с уверенностью в сохранности своих финансовых данных;

Простой контроль над конечным продуктом: обеспечение простого управления продуктом, чтобы позволить оперативно реагировать на ошибки, вносить улучшения и поддерживать качество функционирования;

Доступность расширения каталога: предоставление удобного интерфейса продавцам для обеспечения CRUD операций, что даст простой контроль над актуальными данными собственных товаров;

3.2. Назначение проекта:

Разработка и создание конечного продукта, представляющего из себя онлайн магазин по продаже виртуальных товаров (к примеру, платный контент в онлайн играх или платные подписки на сервисы) в мессенджере Telegram.

У продукта планируется маскот в роли *Neko*³, для обеспечения уникальной особенности. В следствии, подобное решение потребует определённый функционал по отображению и выделению данной особенности при взаимодействии пользователя с интерфейсом продукта;

³ **Neko** (*Catgirl*) – девушка, обладающая кошачьими ушками, хвостом или другими чертами семейства кошачьих, в остальном имеющая полностью человеческое тело. Вымышленный персонаж, часто встречающийся в аниме и манге, а также в различного рода комиксах и видеоиграх.

4. Характеристика объекта автоматизации

4.1. Основные сведения об объекте автоматизации:

Название: Магазин в мессенджере Telegram;

Тип объекта: Онлайн-магазин;

Предмет деятельности: Продажа товаров и услуг через мессенджер Telegram;

Основные функции: Предоставление информации о товарах, приём заказов, обработка платежей, организация доставки, обратная связь, управление клиентской базой;

4.2. Условия эксплуатации объекта автоматизации:

Требования к оборудованию: Выход/доступ к сети интернет;

Режим работы: Онлайн, круглосуточно;

Поддержка пользователей: Обратная связь, обработка заказов в реальном времени, простой и удобный интерфейс;

Безопасность: Защита данных клиентов, безопасные платёжные системы;

4.3. Характеристика окружающей среды:

Целевая аудитория: Пользователи Telegram, интересующиеся приобретением предоставляемых в продукте товаров или услуг;

Тенденции развития: Возможность внедрения новых технологий/функций для улучшения пользовательского опыта и эффективности работы системы, создание экосистемы, включающей филиал онлайн магазина в реализации веб-сайта;

5. Требования к проекту

5.1. Требования к компонентам структуры базы данных:

5.1.1. Требования к структуре таблиц базы данных:

Таблица 5.1.

Таблица полей таблицы “Category”

Наименование	Тип хранимого значения	Флаги
id	INT	PK; NN; AI;
name	VARCHAR(60)	NN; UQ;
description	VARCHAR(160)	NN;
is_blocked	TINYINT(1); DEFAULT 0	NN;

Таблица 5.2.

Таблица полей таблицы “Service”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
category_id (FK)	INT	NN;
name	VARCHAR(60)	NN; UQ;
description	VARCHAR(255)	NN;
image	BLOB	NN;
is_blocked	TINYINT(1); DEFAULT 0	NN;

Таблица 5.3.

Таблица полей таблицы “Product”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
owner_id (FK)	INT	NN;
service_id (FK)	INT	NN;
title	VARCHAR(60)	NN;
description	VARCHAR(255)	NN;
image	BLOB	NN;
price	DECIMAL(6,2)	NN;

Окончание таблицы 5.3.

is_blocked	TINYINT(1); DEFAULT 0	NN;
is_infinite	TINYINT(1); DEFAULT 0	NN;
quantity	INT; DEFAULT 0	NN;

Таблица 5.4.

Таблица полей таблицы “Order”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
product_id (FK)	INT	NN;
customer_id (FK)	INT	NN;
status	ENUM('open', 'pending', 'canceled', 'completed'); DEFAULT 'open'	NN;
total_price	DECIMAL(6,2)	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
updated_at	DATETIME; DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	
closed_at	DATETIME	

Таблица 5.5.

Таблица полей таблицы “User”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
phone_number	VARCHAR(20)	NN; UQ;
email	VARCHAR(255)	UQ;
is_email_notification	TINYINT(1); DEFAULT 0	NN;
is_banned	TINYINT(1); DEFAULT 0	NN;
is_admin	TINYINT(1); DEFAULT 0	NN;
is_seller	TINYINT(1); DEFAULT 0	NN;

Окончание таблицы 5.5.

interface_language_code (FK)	VARCHAR(3); DEFAULT 'eng'	NN;
---------------------------------	------------------------------	-----

Таблица 5.6

Таблица полей таблицы “SupportedInterfaceLanguage”

Наименование	Тип хранимого значения	Флаги
code (PK)	VARCHAR(3)	PK; NN;
name	VARCHAR(60)	NN;

Таблица 5.7.

Таблица полей таблицы “Ticket”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
creator_id (FK)	INT	NN;
responsible_admin_id (FK)	INT	
creator_type	ENUM('user', 'seller'); DEFAULT 'user'	NN;
status	ENUM('open', 'pending', 'resolved', 'canceled'); DEFAULT 'open'	NN;
subject	VARCHAR(100)	NN;
description	TEXT	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
updated_at	DATETIME; DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	
resolved_at	DATETIME	
canceled_reason	TEXT	

Таблица 5.8.

Таблица полей таблицы “TicketSolution”

Наименование	Тип хранимого значения	Флаги
ticket_id (PK)(FK)	INT	PK; NN;
solution	TEXT	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
updated_at	DATETIME; DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	

Таблица 5.9.

Таблица полей таблицы “RequestBlockProduct”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
product_id (FK)	INT	NN;
responsible_admin_id (FK)	INT	
status	ENUM('open', 'pending', 'closed', 'canceled'); DEFAULT 'open'	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
updated_at	DATETIME; DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	
closed_at	DATETIME	
canceled_reason	TEXT	

Таблица 5.10.

Таблица полей таблицы “RequestUnblockProduct”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
product_id (FK)	INT	NN;
responsible_admin_id (FK)	INT	
status	ENUM('open', 'pending', 'closed', 'canceled'); DEFAULT 'open'	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
updated_at	DATETIME; DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	
closed_at	DATETIME	
canceled_reason	TEXT	

Таблица 5.11.

Таблица полей таблицы “RequestDeleteProduct”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
product_id (FK)	INT	NN;
responsible_admin_id (FK)	INT	
status	ENUM('open', 'pending', 'closed', 'canceled'); DEFAULT 'open'	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
updated_at	DATETIME; DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	
closed_at	DATETIME	

Окончание таблицы 5.11.

canceled_reason	TEXT	
-----------------	------	--

Таблица 5.12.

Таблица полей таблицы “RequestEditProduct”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
product_id (FK)	INT	NN;
responsible_admin_id (FK)	INT	
status	ENUM('open', 'pending', 'closed', 'canceled'); DEFAULT 'open'	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
updated_at	DATETIME; DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	
closed_at	DATETIME	
canceled_reason	TEXT	

Таблица 5.13.

Таблица полей таблицы “EditProductData”

Наименование	Тип хранимого значения	Флаги
request_id (PK)(FK)	INT	PK; NN;
new_title	VARCHAR(60)	
new_description	VARCHAR(255)	
new_image	BLOB	
new_price	DECIMAL(6,2)	
is_infinite	TINYINT(1)	
quantity	INT	

Таблица 5.14.

Таблица полей таблицы “RequestAddProduct”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
service_id (FK)	INT	NN;
owner_id (FK)	INT	NN;
responsible_admin_id (FK)	INT	
status	ENUM('open', 'pending', 'closed', 'canceled'); DEFAULT 'open'	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
updated_at	DATETIME; DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	
closed_at	DATETIME	
canceled_reason	TEXT	

Таблица 5.15.

Таблица полей таблицы “AddProductData”

Наименование	Тип хранимого значения	Флаги
request_id (PK)(FK)	INT	PK; NN;
title	VARCHAR(60)	NN;
description	VARCHAR(255)	NN;
image	BLOB	NN;
price	DECIMAL(6,2)	NN;
is_infinite	TINYINT(1); DEFAULT 0	NN;
quantity	INT; DEFAULT 0	NN;

Таблица 5.16.

Таблица полей таблицы “AdminPrivilege”

Наименование	Тип хранимого значения	Флаги
admin_id (PK)(FK)	INT	PK; NN;
is_technical_support	TINYINT(1); DEFAULT 0	NN;
is_request_management	TINYINT(1); DEFAULT 0	NN;
is_category_management	TINYINT(1); DEFAULT 0	NN;
is_service_management	TINYINT(1); DEFAULT 0	NN;
is_product_management	TINYINT(1); DEFAULT 0	NN;
is_delete_product	TINYINT(1); DEFAULT 0	NN;
is_seller_management	TINYINT(1); DEFAULT 0	NN;
is_admin_management	TINYINT(1); DEFAULT 0	NN;
is_user_ban	TINYINT(1); DEFAULT 0	NN;
is_user_unban	TINYINT(1); DEFAULT 0	NN;

Таблица 5.17.

Таблица полей таблицы “AdminRequestAddAdmin”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
responsible_admin_id (FK)	INT	NN;
added_admin_user_id (FK)	INT	NN;
status	ENUM('open', 'pending', 'closed'); DEFAULT 'open'	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
closed_at	DATETIME	

Таблица 5.18.

Таблица полей таблицы “AdminRequestAddSeller”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
responsible_admin_id (FK)	INT	NN;
added_seller_user_id (FK)	INT	NN;

Окончание таблицы 5.18.

status	ENUM('open', 'pending', 'closed'); DEFAULT 'open'	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
closed_at	DATETIME	

Таблица 5.19.

Таблица полей таблицы “AdminRequestKickAdmin”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
responsible_admin_id (FK)	INT	NN;
kickable_admin_user_id (FK)	INT	NN;
status	ENUM('open', 'pending', 'closed'); DEFAULT 'open'	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
closed_at	DATETIME	

Таблица 5.20.

Таблица полей таблицы “AdminRequestKickSeller”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
responsible_admin_id (FK)	INT	NN;
kickable_seller_user_id (FK)	INT	NN;
status	ENUM('open', 'pending', 'closed'); DEFAULT 'open'	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
closed_at	DATETIME	

Таблица 5.21.

Таблица полей таблицы “AdminRequestBanUser”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
responsible_admin_id (FK)	INT	NN;
banned_user_id (FK)	INT	NN;
status	ENUM('open', 'pending', 'closed'); DEFAULT 'open'	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
closed_at	DATETIME	

Таблица 5.22.

Таблица полей таблицы “AdminRequestUnbanUser”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
responsible_admin_id (FK)	INT	NN;
unbanned_user_id (FK)	INT	NN;
status	ENUM('open', 'pending', 'closed'); DEFAULT 'open'	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
closed_at	DATETIME	

Таблица 5.23.

Таблица полей таблицы “AdminRequestBlockCategory”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
category_id (FK)	INT	NN;
responsible_admin_id (FK)	INT	NN;
status	ENUM('open', 'pending', 'closed'); DEFAULT 'open'	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;

Окончание таблицы 5.23.

closed_at	DATETIME	
-----------	----------	--

Таблица 5.24.

Таблица полей таблицы “AdminRequestUnblockCategory”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
category_id (FK)	INT	NN;
responsible_admin_id (FK)	INT	NN;
status	ENUM('open', 'pending', 'closed'); DEFAULT 'open'	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
closed_at	DATETIME	

Таблица 5.25.

Таблица полей таблицы “AdminRequestEditCategory”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
category_id (FK)	INT	NN;
responsible_admin_id (FK)	INT	NN;
status	ENUM('open', 'pending', 'closed'); DEFAULT 'open'	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
closed_at	DATETIME	

Таблица 5.26.

Таблица полей таблицы “AdminEditCategoryData”

Наименование	Тип хранимого значения	Флаги
request_id (PK)(FK)	INT	PK; NN;
new_name	VARCHAR(60)	
new_description	VARCHAR(160)	

Таблица 5.27.

Таблица полей таблицы “AdminRequestAddCategory”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
responsible_admin_id (FK)	INT	NN;
status	ENUM('open', 'pending', 'closed'); DEFAULT 'open'	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
closed_at	DATETIME	

Таблица 5.28.

Таблица полей таблицы “AdminAddCategoryData”

Наименование	Тип хранимого значения	Флаги
request_id (PK)(FK)	INT	PK; NN;
new_name	VARCHAR(60)	NN;
new_description	VARCHAR(160)	NN;

Таблица 5.29.

Таблица полей таблицы “AdminRequestBlockService”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
service_id (FK)	INT	NN;
responsible_admin_id (FK)	INT	NN;
status	ENUM('open', 'pending', 'closed'); DEFAULT 'open'	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
closed_at	DATETIME	

Таблица 5.30.

Таблица полей таблицы “AdminRequestUnblockService”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;

service_id (FK)	INT	NN;
responsible_admin_id (FK)	INT	NN;
status	ENUM('open', 'pending', 'closed'); DEFAULT 'open'	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
closed_at	DATETIME	

Таблица 5.31.

Таблица полей таблицы “AdminRequestEditService”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
service_id (FK)	INT	NN;
responsible_admin_id (FK)	INT	NN;
status	ENUM('open', 'pending', 'closed'); DEFAULT 'open'	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
closed_at	DATETIME	

Таблица 5.32.

Таблица полей таблицы “AdminEditServiceData”

Наименование	Тип хранимого значения	Флаги
request_id (PK)(FK)	INT	PK; NN;
new_name	VARCHAR(60)	
new_description	VARCHAR(160)	
new_image	BLOB	

Таблица 5.33.

Таблица полей таблицы “AdminRequestAddService”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
category_id (FK)	INT	NN;
responsible_admin_id (FK)	INT	NN;

Окончание таблицы 5.33.

status	ENUM('open', 'pending', 'closed'); DEFAULT 'open'	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
closed_at	DATETIME	

Таблица 5.34.

Таблица полей таблицы “AdminAddServiceData”

Наименование	Тип хранимого значения	Флаги
request_id (PK)(FK)	INT	PK; NN;
new_name	VARCHAR(60)	NN;
new_description	VARCHAR(160)	NN;
new_image	BLOB	NN;

Таблица 5.35.

Таблица полей таблицы “AdminRequestDeleteProduct”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
product_id (FK)	INT	NN;
responsible_admin_id (FK)	INT	NN;
status	ENUM('open', 'pending', 'closed'); DEFAULT 'open'	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
closed_at	DATETIME	

Таблица 5.36.

Таблица полей таблицы “AdminRequestBlockProduct”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
product_id (FK)	INT	NN;
responsible_admin_id (FK)	INT	NN;

Окончание таблицы 5.36.

status	ENUM('open', 'pending', 'closed'); DEFAULT 'open'	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
closed_at	DATETIME	

Таблица 5.37.

Таблица полей таблицы “AdminRequestUnblockProduct”

Наименование	Тип хранимого значения	Флаги
id (PK)	INT	PK; NN; AI;
product_id (FK)	INT	NN;
responsible_admin_id (FK)	INT	NN;
status	ENUM('open', 'pending', 'closed'); DEFAULT 'open'	NN;
created_at	DATETIME; DEFAULT CURRENT_TIMESTAMP	NN;
closed_at	DATETIME	

5.1.2. Требования к назначению таблиц базы данных:

Таблица 5.38.

Таблица описания назначения таблиц базы данных

Наименование таблицы	Описание назначения
Category	Таблица “Category” используется для управления актуальными категориями сервисов продукта. К примеру: “Game”, “Film”, “Music”
Service	Таблица “Service” используется для управления актуальными сервисами определённой категории продукта. К примеру, категория “Gift Card”: “Amazon”, “Xbox”, “Microsoft”, “Apple”.

Продолжение таблицы 5.38.

Product	Таблица “Product” используется для управления актуальными товарами определённого сервиса продукта. К примеру, сервис “Steam wallet”: “Refill your wallet for 5\$”, “Refill your wallet for 10\$”, “Refill your wallet for 25\$”
User	Таблица “User” используется для управления всеми учётными записями продукта. Не разделяя на отдельные таблицы привилегии учётной записи. Под привилегиями подразумевается статус учётной записи, к примеру: “Admin”, “Seller”.
Order	Таблица “Order” используется для управления заказами продукта в любом состоянии. Под состоянием заказа подразумевается его статус: ожидает выполнения, отменён, возвращён, завершён.
SupportedInterfaceLanguage	Таблица “SupportedInterfaceLanguage” используется для управления актуальными языковыми локализациями интерфейса продукта.
Ticket	Таблица “Ticket” используется для управления ticket-ами (запросами) в техническую поддержку продукта.
TicketSolution	Таблица “TicketSolution” используется для управления решениями (ответами) на ticket-ы отправленные в техническую поддержку.
RequestBlockProduct	Таблица “RequestBlockProduct” используется для управления запросами на блокировку товара в каталоге продукта.

Продолжение таблицы 5.38.

RequestUnblockProduct	Таблица “RequestUnblockProduct” используется для управления запросами на разблокировку товара в каталоге продукта.
RequestDeleteProduct	Таблица “RequestDeleteProduct” используется для управления запросами на удаление товара в каталоге продукта.
RequestEditProduct	Таблица “RequestEditProduct” используется для управления запросами на изменение данных товара в каталоге продукта.
EditProductData	Таблица “EditProductData” используется для хранения данных запроса на изменение данных товара в каталоге продукта.
RequestAddProduct	Таблица “RequestAddProduct” используется для управления запросами на добавление нового товара в каталог продукта.
AddProductData	Таблица “AddProductData” используется для хранения данных запроса на добавление нового товара в каталог продукта.
AdminPrivilege	Таблица “AdminPrivilege” используется для описания доступных привилегий/разрешений для каждого отдельного админа.
AdminRequestBanUser	Таблица “AdminRequestBanUser” используется для хранения запросов от администраторов на блокировку пользователей.
AdminRequestUnbanUser	Таблица “AdminRequestUnbanUser” используется для хранения запросов от администраторов на разблокировку пользователей.

Продолжение таблицы 5.38.

AdminRequestAddAdmin	Таблица “AdminRequestAddAdmin” используется для хранения запросов от администраторов на повышение указанной учётной записи до администратора.
AdminRequestKickAdmin	Таблица “AdminRequestKickAdmin” используется для хранения запросов от администраторов на понижение другой учётной записи администратора до обычного пользователя.
AdminRequestAddSeller	Таблица “AdminRequestAddSeller” используется для хранения запросов от администраторов на повышение указанной учётной записи до продавца.
AdminRequestKickSeller	Таблица “AdminRequestKickSeller” используется для хранения запросов от администраторов на понижение учётной записи продавца до обычного пользователя.
AdminRequestBlockCategory	Таблица “AdminRequestBlockCategory” используется для хранения запросов от администраторов на блокировку категории.
AdminRequestUnblockCategory	Таблица “AdminRequestUnblockCategory” используется для хранения запросов от администраторов на разблокировку категории.
AdminRequestEditCategory	Таблица “AdminRequestEditCategory” используется для хранения запросов от администраторов на редактирование категории.
AdminEditCategoryData	Таблица “AdminEditCategoryData” используется для хранения данных запроса от администратора на изменение категории.

Продолжение таблицы 5.38.

AdminRequestAddCategory	Таблица “AdminRequestAddCategory” используется для хранения запросов от администраторов на добавление категории.
AdminAddCategoryData	Таблица “AdminAddCategoryData” используется для хранения данных запроса от администратора на добавление категории.
AdminRequestBlockService	Таблица “AdminRequestBlockService” используется для хранения запросов от администраторов на блокировку сервиса категории продукта.
AdminRequestUnblockService	Таблица “AdminRequestUnblockService” используется для хранения запросов от администраторов на разблокировку сервиса категории продукта.
AdminRequestEditService	Таблица “AdminRequestEditService” используется для хранения запросов от администраторов на редактирование сервиса категории продукта.
AdminEditServiceData	Таблица “AdminEditServiceData” используется для хранения данных запроса от администратора на изменение сервиса категории продукта.
AdminRequestAddService	Таблица “AdminRequestAddService” используется для хранения запросов от администраторов на добавление сервиса в категорию продукта.
AdminAddServiceData	Таблица “AdminAddServiceData” используется для хранения данных запроса от администратора на добавление сервиса в категорию продукта.

Окончание таблицы 5.38.

AdminRequestBlockProduct	Таблица “AdminRequestBlockProduct” используется для хранения запросов от администраторов на блокировку товара определённого сервиса продукта.
AdminRequestUnblockProduct	Таблица “AdminRequestUnblockProduct” используется для хранения запросов от администраторов на разблокировку товара определённого сервиса продукта.
AdminRequestDeleteProduct	Таблица “AdminRequestDeleteProduct” используется для хранения запросов от администраторов на удаление товара из определённого сервиса продукта.

5.1.3. Визуальное представление схемы базы данных:

Визуальное представление, а также готовую модель для просмотра и идентификации связей между таблицами можно отыскать в репозитории проекта по ссылке: <https://github.com/4-proxy/NekoShop>.

Более точное расположение, определяется: /templates/database;

5.2. Требования к функциональным характеристикам:

5.2.1. Функциональные требования к интерфейсу клавиатур в Telegram:

Данный подраздел посвящён описанию функциональных требований настоящего технического задания к элементам текущего раздела “5. Требования к проекту”, подраздел: [5.3.3. Требования к элементам интерфейса в Telegram](#);

5.2.1.1. Функциональные требования на уровне обычного пользователя:

Таблица 5.39.

Таблица функциональных требований “Личный кабинет”

Функциональное требование	Подробное описание
Регистрация аккаунта в базе данных продукта	<p>Для первичной регистрации пользователь должен поделиться своим номером телефона учётной записи Telegram в чате с ботом.</p> <p>Бот проведёт проверку на дубликат номера в базе данных, если такой номер не зарегистрирован, то бот создаст новую запись и добавит указанный номер телефона в соответствующее поле.</p>
Привязка адреса электронной почты (email)	<p>При регистрации или использовании зарегистрированного аккаунта, должна быть доступна возможность привязать email к аккаунту.</p> <p>Бот проверит не зарегистрирован ли указанный email в базе данных, если нет, то бот отправит сообщение с кодом подтверждения на указанный email.</p> <p>В следствии, пользователю нужно будет отправить полученный код в чат с ботом для верификации.</p>
Авторизация по номеру телефона	<p>При начале нового чата с ботом пользователь должен иметь возможность авторизоваться, поделившись номером телефона учётной записи Telegram в чате.</p> <p>При условии, что аккаунт с указанным номером телефона зарегистрирован в базе данных.</p>

Окончание таблицы 5.39.

Авторизация по адресу электронной почты (email)	<p>При начале нового чата с ботом пользователь должен иметь возможность авторизоваться, получив и отправив код подтверждения с указанного email.</p> <p>При условии, что указанный пользователем email зарегистрирован за аккаунтом.</p>
Просмотр истории заказов	Пользователь может запросить свою историю заказов, после чего бот отправит записи о заказах пользователя в файле формата “.pdf” за последний год.
Удаление аккаунта	Пользователю должна предоставляться возможность удалить свою учётную запись из базы данных продукта. Для подтверждения удаления необходимо отправить текст "Подтверждаю".
Смена электронной почты (email)	Пользователь должен иметь возможность изменить привязанный email к аккаунту, указав новый email и отправив код подтверждения, полученный на неё в чат с ботом.
Просмотр ожидаемых заказов	При запросе информации об ожидаемых заказах (товар оплачен, но ещё не получен) бот сообщит информацию о заказах, которые ожидает пользователь.
Выбор языка интерфейса	Пользователь должен иметь возможность выбрать язык интерфейса продукта.

Таблица 5.40.

Таблица функциональных требований “Каталог”

Функциональное требование	Подробное описание
Отображение категорий сервисов	При переходе в каталог, система должна отображать все доступные и актуальные категории сервисов пользователю. Заблокированные категории не должны отображаться у обычного пользователя.
Отображение доступных сервисов выбранной категории	При выборе категории в каталоге, система должна отображать доступные и актуальные сервисы для выбора и приобретения товара в них. Заблокированные сервисы не должны подлежать отображению обычному пользователю.
Отображение доступных товаров выбранного сервиса	При выборе сервиса системе требуется отобразить доступные товары, а также актуальные данные о них пользователю.
Поиск товара или сервиса в каталоге	Пользователю должен быть доступен поиск товара или сервиса в каталоге продукта по ключевым словам. Пользователь будет вводить ключевые слова, которые будут использованы для поиска требуемого товара или сервиса в базе данных каталога.

Таблица 5.41.

Таблица функциональных требований “О нас”

Функциональное требование	Подробное описание
Отображение данных о продукте	Система предоставит и отобразит доступные данные о продукте пользователю в чате.

Таблица 5.42.

Таблица функциональных требований “Техническая поддержка”

Функциональное требование	Подробное описание
Сообщить об неполадке	У пользователя должна быть возможность уведомить техническую поддержку о неисправностях продукта в целом или его отдельных компонентов. Требуется соответствующий интерфейс, сопряжённый с формой для заполнения данных.
Просмотр часто задаваемых вопросов (FAQ)	Система должна иметь возможность предоставить актуальный список часто задаваемых вопросов в техническую поддержку и соответствующие ответы на них пользователю для дальнейшего просмотра.

5.2.1.2. Функциональные требования на уровне продавца:

Таблица 5.43.

Таблица функциональных требований “Каталог собственных товаров”

Функциональное требование	Подробное описание
CRUD операции для товаров	<p>Продавцу должен быть представлен удобный, понятный и простой интерфейс для обеспечения CRUD операций над товарами в его владении.</p> <p>Какие-либо изменения, касающиеся данных о товарах, будут осуществляться заполнением специальных форм продавцом, тем самым создавая запрос на определённую операцию для изменения товара.</p>

Окончание таблицы 5.43.

Просмотр активных запросов	Система должна иметь возможность предоставить просмотр активных запросов на изменение, а также просмотр заполненных форм, которые отправил/заполнил продавец.
Отмена запроса	У продавца должна быть возможность отменить активный запрос на изменение товара, в случае если он передумал.
Отредактировать запрос	У продавца должна быть возможность отредактировать отправленный запрос на изменение товара.
Блокировка товара	Система должна предоставлять продавцу функционал блокировки собственного товара от возможности приобретения со стороны обычных пользователей.
Разблокировка товара	Система должна предоставлять продавцу функционал для разблокировки собственного товара, заблокированного к приобретению.

Функциональные требования к обратной связи с технической поддержкой продукта соответствует функциональным требованиям на уровне обычного пользователя с тем замечанием, что любой запрос имеет поле с указанием роли отправителя. (подробнее в разделе “5. Требования к проекту”, подпункт: [5.2.1.1. Функциональные требования на уровне обычного пользователя](#), таблица 5.42.)

5.2.1.3. Функциональные требования на уровне администратора:

Таблица 5.44.

Таблица функциональных требований “Запросы на изменение товаров”

Функциональное требование	Подробное описание
Получение запроса на изменение товара	Администраторам требуется предоставить функционал для работы с запросами.
Одобрение запроса на изменение товаров	При одобрении запроса администратором система должна менять статус запроса на закрытый и помечать дату его закрытия.
Редактирование запроса на изменение товаров	При редактировании данных в полученных формах запроса, системе потребуется отправить изменённые формы обратно продавцу, для дальнейшего подтверждения с его стороны.
Отклонение запроса на изменение товаров	<p>При выборе отклонения со стороны администратора, он обязан указать причину, по которой запрос был отклонён. Данная причина должна быть закреплена в базе данных к текущему запросу, для дальнейшей отправки продавцу.</p> <p>При отклонении запроса, системе требуется уведомить продавца о том, что его запрос был отклонён. Прикрепив к сообщению с уведомлением, причину отказа.</p>

Таблица 5.45.

Таблица функциональных требований “Техническая поддержка”

Функциональное требование	Подробное описание
Получение ticket-а для технической поддержки	Система должна предоставить функционал администраторам для получения и работы над ticket-ом, отправленным в техническую поддержку.

Окончание таблицы 5.45

Просмотр ticket-ов	Администраторам требуется возможность получить случайный или определённый ticket используя его идентификатор.
Ticket продавца	Администратору должен быть доступен вариант просмотра ticket-а от продавца.
Принятие ticket-а	Если администратор принимает ticket, то система должна предоставить функционал для создания ответа, который будет передан отправителю запроса. Так же система отметит ticket как принятый на рассмотрение и укажет принявшего администратора. (Если ticket считается принятым, то другие администраторы не могут его получить)
Отклонение ticket-а	Если администратор отклоняет ticket, то ему требуется указать причину отклонения. Система изменит статус ticket-а на отменённый и оповестит отправителя прикрепив причину отклонения.

Таблица 5.46.

Таблица функциональных требований “Работа с каталогом”

Функциональное требование	Подробное описание
Добавление категории	Система должна предоставлять администратору форму для добавления новой категории сервисов в базу данных продукта.
Изменение категории	Администратору должна быть доступна возможность отредактировать данные категории сервисов в базе данных продукта.

Продолжение таблицы 5.46

Блокировка категории	Система должна предоставлять администратору функционал блокировки категории не позволяя получить доступ к сервисам заблокированной категории.
Разблокировка категории	Система должна предоставлять администратору функционал разблокировки заблокированной категории сервисов продукта.
Добавление сервиса	Система должна предоставлять администратору форму для добавления нового сервиса в выбранную категорию продукта.
Изменение сервиса	Администратору должна быть доступна возможность отредактировать данные сервиса выбранной категории в базе данных продукта.
Блокировка сервиса	Система должна предоставлять администратору функционал блокировки сервиса для его сокрытия от пользователей продукта.
Разблокировка сервиса	Система должна предоставлять администратору функционал разблокировки заблокированного сервиса, для возобновления доступа со стороны пользователей продукта.
Поиск товара	Система должна предоставлять функционал поиска товара в базе данных продукта по идентификатору записи, а также по ключевым словам.
Блокировка товара	Система должна предоставлять администратору функционал блокировки товара.

Окончание таблицы 5.46.

Разблокировка товара	Система должна предоставлять администратору функционал разблокировки заблокированного товара.
Изменение товара	Администратору должна быть доступна возможность отредактировать данные товара. Данный способ является ручным, в отличие от одобрения заполненной формы запроса на изменение товара от продавца.
Удаление товара	Администратору должна быть доступна возможность удалить товар из базы данных продукта.

Таблица 5.47.

Таблица функциональных требований “Работа с каталогом”

Функциональное требование	Подробное описание
Добавление администратора	Администратору с определёнными привилегиями должна быть доступна функция повышения учётной записи обычного пользователя до администратора.
Настройка привилегий администратора	Администратору с определёнными привилегиями должна быть доступна функция настройки привилегий других администраторов. Привилегии для настройки не переходят границы доступных администратора, иницирующего настройку.
Лишение прав администратора	Администратору с определёнными привилегиями должна быть доступна функция понижения учётной записи администратора до обычного пользователя, если уровень привилегий учётной записи не выше инициатора.

Окончание таблицы 5.47.

Добавление продавца	Администратору с определёнными привилегиями должна быть доступна функция повышения учётной записи обычного пользователя до продавца.
Лишение прав продавца	Администратору с определёнными привилегиями должна быть доступна функция понижения учётной записи продавца до обычного пользователя. Товары продавца получают статус блокировки и становятся недоступны для изменения.
Блокировка пользователя	Администратору с определёнными привилегиями должна быть доступна функция блокировки учётной записи в базе данных продукта при помощи идентификатора, номера телефона или email.

5.2.2. Иные функциональные требования:

Таблица 5.48.

Таблица функциональных требований к логированию продукта

Функциональное требование	Подробное описание
Уровни логирования	Логирование должно поддерживать различные уровни, такие как DEBUG, INFO, WARNING, ERROR и CRITICAL, чтобы была возможность фильтровать сообщения по важности.
Формат сообщений	Логи должны быть записаны в едином стандартизированном формате для обеспечения читаемости и простоты работы с ними.
Конфиденциальность данных	Логи не должны содержать конфиденциальную информацию, такую как пароли или личные данные.

Таблица 5.49.

Таблица функциональных требований к обработке состояний бота

Функциональное требование	Подробное описание
Выход бота в онлайн	При запуске продукта, система должна обрабатывать данное событие. После чего отправлять сообщение в личный чат владельца бота с соответствующим текстом о выходе в онлайн.
Выход бота в офлайн	При отключении продукта, система должна обрабатывать данное событие. После чего отправлять сообщение в личный чат владельца бота с соответствующим текстом о выходе в офлайн.

5.3. Требования к информационной и программной совместимости:

5.3.1. Требования к языкам программирования и программным средствам:

5.3.1.1. Требования к языкам программирования:

Основной язык программирования: Python 3.12.X;

Вспомогательные языки: PowerShell, SQL, Bash, Jenkinsfile, Dockerfile;

5.3.1.2. Требования к программным средствам:

Выбранные среды разработки: VSCode, PowerShell ISE, MySQL Workbench 8.0 CE;

Вспомогательные средства: MySQL Community Server 8.4.X (GPL), Docker Engine 27.1.X, Jenkins 2.462.X (LTS);

Основные используемые библиотеки и фреймворки: Aiogram 3.13.X, MySQL Connector/Python 9.0.X;

Платформы и окружение разработки: Windows 10, Windows Server 2019, Ubuntu Server 24.04.X (LTS), DockerImage - python:3.12-slim;

5.3.2. Требования к исходному коду:

Читаемость и структура кода: Исходный код должен быть написан понятно и четко, с использованием понятных имен переменных, функций и классов. Код должен быть структурирован с использованием отступов, комментариев и разделения на логические блоки;

Соблюдение стандартов кодирования: Исходный код должен соответствовать установленным стандартам кодирования для используемого языка, включая правила оформления кода, стилистику и соглашения об именовании;

Документирование: Исходный код должен сопровождаться подробной документацией, описывающей его функциональность, структуру, входные и выходные данные, а также спецификации API, если применимо;

Тестирование: Код должен быть написан с учетом возможности тестирования;

5.3.3. Требования к элементам интерфейса в Telegram:

Данный подраздел описывает требования к элементам интерфейса для проекта “NekoShop” в Telegram.

Поскольку продукт представлен в виде бота в чате Telegram, он будет использовать встроенные и предоставляемые элементы интерфейса Telegram для взаимодействия с пользователем, такие как клавиатура и кнопки.

В этом разделе используются специальные обозначения для клавиатур, такие как [...], что означает использование отдельной кнопки для клавиатуры, и [...]*, сообщающее об индивидуальной структуре и неопределённом количестве элементов, которыми являются отдельные кнопки;

5.3.4.1. Вид клавиатур для обычного пользователя:

- Новый чат (не авторизованный чат с ботом)
 - [Личный кабинет]
 - [Зарегистрироваться]
 - [Авторизоваться]
- Главная (авторизованный чат с ботом)
 - [Личный Кабинет]
 - [Удалить аккаунт]
 - [Сменить почту]
 - [История заказов]
 - [Заказы в ожидании]
 - [Язык интерфейса]

- [Каталог]
 - [Поиск]
 - [Категория]*
 - [Сервис]*
 - [Товар]*
- [О нас]
- [Техническая поддержка]
 - [Сообщить об неполадке]
 - [FAQ]

5.3.4.2. Вид клавиатур для продавца:

- Особый кабинет (авторизованный чат с ботом)
 - [Каталог собственных товаров]
 - [Посмотреть товары]
 - [Просмотреть активные запросы]
 - [Отредактировать запрос]
 - [Отменить запрос]
 - [Создать запрос]
 - [Добавить товар]
 - [Отредактировать товар]
 - [Удалить товар]
 - [Блокировка товара]
 - [Заблокировать]
 - [Разблокировать]
 - [Техническая поддержка]
 - [Сообщить об неполадке]
 - [FAQ]

5.3.4.3. Вид клавиатур для администратора:

- Особый кабинет (авторизованный чат с ботом)

- [Запросы на изменение товаров]
 - [Получить запрос продавца]
 - [Одобрить]
 - [Отредактировать запрос]
 - [Отклонить]
- [Техническая поддержка]
 - [Случайный Ticket]
 - [Принять]
 - [Отклонить]
 - [Ticket по идентификатору]
 - [Принять]
 - [Отклонить]
 - [Ticket от продавца]
 - [Принять]
 - [Отклонить]
- Панель администратора (с учётом привилегий)
 - [Работа с каталогом]
 - [Категории сервисов]
 - [Категория]*
 - [Добавить категорию]
 - [Отредактировать категорию]
 - [Блокировка категории]
 - [Заблокировать]
 - [Разблокировать]
 - [Сервисы категории]
 - [Сервис]*
 - [Добавить сервис]
 - [Отредактировать сервис]

- [Блокировка сервиса]
 - [Заблокировать]
 - [Разблокировать]
- [Товары]
 - [Поиск товара]
 - [По идентификатору]
 - [По ключевым словам]
 - [Блокировка товара]
 - [Заблокировать]
 - [Разблокировать]
 - [Удалить товар]
- [Управление пользователями]
 - [Администратор]
 - [Добавить администратора]
 - [Настроить привилегии]
 - [Удалить из администрации]
 - [Продавец]
 - [Добавить продавца]
 - [Удалить из продавцов]
 - [Заблокировать учётную запись]
 - [По идентификатору]
 - [По номеру телефона]
 - [По email]
 - [Разблокировать учётную запись]
 - [По идентификатору]
 - [По номеру телефона]
 - [По email]

6. Стадии и этапы разработки

1. Планирование

- Определение основной идеи проекта, целей и задач. Формирование краткого описания проекта и его возможностей;
- Поиск и ознакомление с требуемой информацией, а также документами для дальнейшего формирования набора документов, требуемых для составления первичной документации касательно разработки проекта;
- Формирование и составление основы технического задания, которое включает в себя сопутствующие действия, например: определение функциональных требований, анализ объекта автоматизации, анализ и формирование требований к программным средствам и т.д.;

2. Прототипирование

- Поиск, изучение и анализирование информации, программных средств и решений в программном коде, которые в дальнейшем могут быть использованы для разработки;
- Проектирование и создание первичных/тестовых прототипов и макетов для тестирования определённого функционала будущей системы;
- Анализ разработанных прототипов и их дальнейшее слияние для проверки отдельных функциональных областей будущей системы;
- Доработка технического задания;
- Проектирование архитектуры будущей системы на основе разработанных прототипов и сформированного технического задания;

3. Разработка

- Подготовка, установка и настройка компонентов рабочей области и рабочего окружения для дальнейшей разработки;
- Разработка базового основного функционала системы и создание первичной настройки;
- Формирование структуры и скелета системы;
- Разработка основного функционала и своевременное тестирование разработанных элементов кода;
- Составление документации проекта;
- Корректирование настроек системы и приведение их к завершённому виду;
- Разработка дополнительного функционала;
- Актуализация настроек системы;
- Актуализация настроек вспомогательных программных средств;
- Актуализация проектной документации;

4. Тестирование

- Анализ и формирование основных направлений, настроек и ситуаций для всестороннего тестирования системы;
- Подготовка требуемых программных решений для всестороннего тестирования;
- Проведение всестороннего тестирования путём ручного тестирования и тестирования с использованием программных решений;
- Анализ и оценка результатов тестирования, внесение соответствующих изменений и корректировок в проект;

Приложения

1. Стандарт документирования

- Полное наименование – “Стандарт документирования - Рекомендации по форматированию документации проектов в Microsoft Word”.
- Аннотация – Документ содержит рекомендации и правила для форматирования документов в “Microsoft Word”.
- Назначение - Используется для форматирования .docx документов проекта “NekoShop”.
- Ссылка на ресурс - 4-proxy/SpecificDocumentation
- Дополнительные свойства:
 - Дата публикации от – 19.06.2024
 - Версия – 1.0.0
 - Уникальный идентификатор – x04

2. Семантическое версионирование

- Полное наименование – “Семантическое Версионирование 2.0.0”.
- Аннотация – Документ содержит подход к управлению версиями программного обеспечения.
- Назначение - Используется для управления версиями исходного кода в проекте “NekoShop”.
- Ссылка на ресурс - 4-proxy/SpecificDocumentation
- Дополнительные свойства:
 - Версия – 2.0.0
 - Уникальный идентификатор – x01

3. ГОСТ 34.602-2020

- Полное наименование – “Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы”.
- Аннотация – Документ устанавливает требования к содержанию технического задания на создание автоматизированной системы.
- Назначение – Используется для формулирования структуры технического задания проекта “NekoShop”.
- Ссылка на ресурс - 4-proxy/SpecificDocumentation
- Дополнительные свойства:
 - Уникальный идентификатор – x03

4. ГОСТ 19.201-78

- Полное наименование – “Единая система программной документации. Техническое задание. Требования к содержанию и оформлению”
- Аннотация – Документ устанавливает порядок построения и оформления технического задания на разработку программы или программного изделия для вычислительных машин, комплексов и систем независимо от их назначения и области применения.
- Назначение – Используется для формулирования структуры технического задания проекта “NekoShop”.
- Ссылка на ресурс - 4-proxy/SpecificDocumentation
- Дополнительные свойства:
 - Уникальный идентификатор – x02