**Оглавление.**

1. Введение 3

1.1. Основная цель: 3

1.2. Дополнительные цели: 3

1.3. Целевая аудитория: 3

1.4. Область применения: 4

2. Функциональные требования 5

2.1. Дизайн: 5

2.2. Экран приветствия: 7

2.3. Экран авторизации/регистрации: 7

2.4. Основной экран: 7

2.5. Профиль пользователя: 8

2.5.1. Общие требования: 8

2.5.2. Изменение пароля: 8

2.6. Отслеживание заказов: 8

2.6.1. История заказов: 8

2.6.2. Детальная информация о заказе: 8

2.7. Скидки: 8

2.7.1. Отображение скидок: 8

2.7.2. Раздел «Discounts»: 9

3. Технические требования 10

3.1. Платформы: 10

3.2. Языки программирования: 10

3.3. База данных: 10

3.4. API: 10

3.5. Push-уведомления: 10

4. Требования к интерфейсу пользователя 12

4.1. Интуитивно понятный интерфейс: 12

4.2. Адаптивность: 12

4.3. Дизайн: 12

4.4. Требования к отдельным экранам: 12

4.4.1. Экран приветствия: 12

4.4.2. Экран авторизации/регистрации: 12

4.4.3. Основной экран: 12

4.4.4. Экран списка заказов: 12

4.4.5. Экран деталей заказа: 13

4.4.6. Экран скидок: 13

5. Требования к безопасности 14

5.1. Авторизация: 14

5.2. Защита данных: 14

6. Требования к тестированию 15

6.1. Юнит-тестирование: 15

6.2. Интеграционное тестирование: 15

6.3. Системное тестирование: 15

6.4. Пользовательское тестирование: 16

7. Документация 18

7.1. Техническое задание: 18

7.2. Спецификация API: 18

7.3. Руководство пользователя: 18

7.4 Требования к документации: 18

8. Словарь 19

9. Сроки и этапы разработки 21

10. Критерии приемки 22

10.1. Функциональные требования: 22

10.2. Нефункциональные требования: 22

10.3. Тестирование: 22

10.4. Документация: 23

# Введение

## Основная цель:

Разработать нативное мобильное приложение, которое предоставляет клиентам компании удобный и интуитивно понятный инструмент для отслеживания своих заказов компьютерной техники, управления ими, получения актуальной информации о скидках и акциях, а также оперативных уведомлений о любых изменениях в их заказах.

### Дополнительные цели:

* Повышение лояльности клиентов за счет улучшения сервиса и предоставления удобного инструмента для самообслуживания.
* Снижение нагрузки на службу поддержки клиентов за счет предоставления пользователям возможности самостоятельно получать ответы на большинство вопросов о своих заказах.
* Увеличение продаж за счет информирования клиентов о скидках и акциях.
* Сбор данных о поведении пользователей в приложении для последующего улучшения функциональности и пользовательского опыта.
* Создание позитивного имиджа компании как современной и клиентоориентированной.

## Целевая аудитория:

* Описание:
  + Физические лица, заказывающие компьютерную технику онлайн или в розничных магазинах компании.
  + Корпоративные клиенты, заказывающие компьютерную технику для своих сотрудников.
* Возраст: 18-65 лет (основная группа).
* Уровень владения мобильными устройствами: от среднего до продвинутого.
* Предпочтительные операционные системы: Android.
* Потребности целевой аудитории:
* Оперативное получение информации о статусе заказа.
* Удобный доступ к истории заказов.
* Возможность быстро связаться со службой поддержки при необходимости.
* Своевременное получение информации о скидках и акциях на интересующие товары.
* Персонализированные уведомления, соответствующие их потребностям и интересам.

## Область применения:

Приложение будет использоваться для:

* Просмотр истории заказов.
* Оформления новых заказов.
* Просмотра товаров в каталоге.
* Получения детальной информации о каждом заказе.
* Просмотра доступных и примененных скидок.
* Получения push-уведомлений.

# Функциональные требования

## Дизайн:

Дизайн приложения должен соответствовать макету (рис. 1, 2, 3, 4):

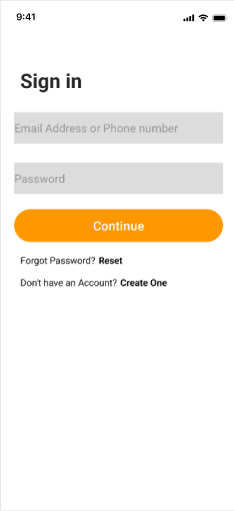
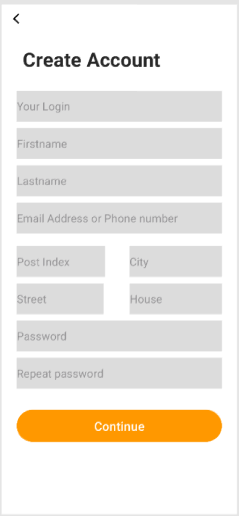
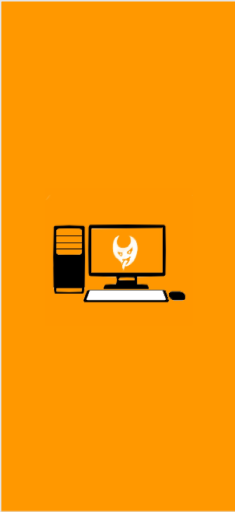


Рисунок 1 - экран загрузки, авторизации, регистрации

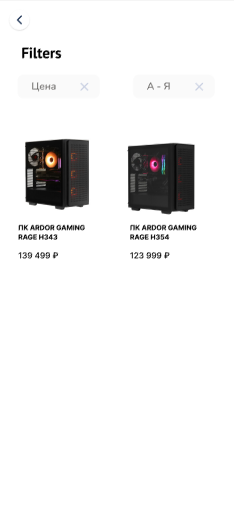
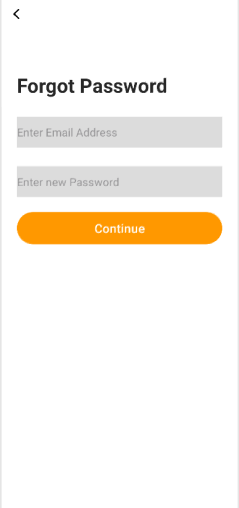


Рисунок 2 - экран восстановления пароля, главной страницы, фильтров

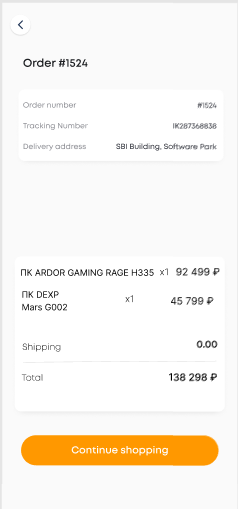
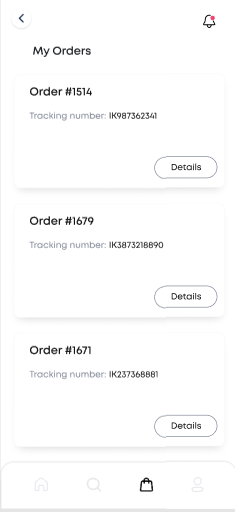
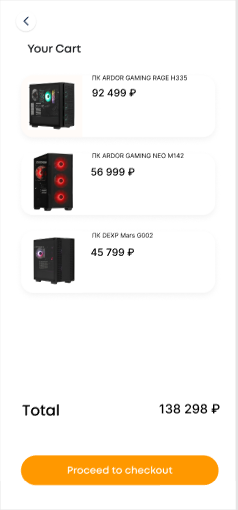


Рисунок 3 - экран корзины, заказа

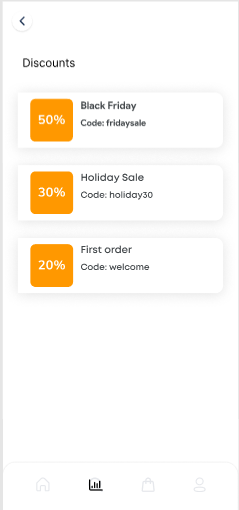
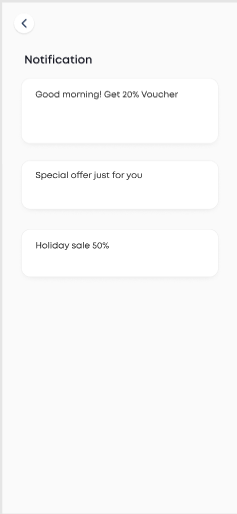


Рисунок 4 - экран уведомлений, скидок

## Экран приветствия:

* Экран должен отображать логотип компании.
* Если пользователь не авторизован, после отображения экрана приветствия система должна автоматически перенаправить пользователя на экран авторизации/регистрации.
* Если пользователь уже авторизован, система должна автоматически перенаправить пользователя на основной экран приложения.

## Экран авторизации/регистрации:

* Экран должен содержать поле для ввода логина (email или номер телефона).
* Экран должен содержать поле для ввода пароля.
* Поля должны быть подписаны.
* Экран должен содержать кнопку «Continue».
* При нажатии на кнопку «Continue» система должна попытаться авторизовать пользователя с введенным логином и паролем.
* Экран должен содержать ссылку «Forgot password?».
* При нажатии на ссылку «Reset password» должен происходить переход к функциональности восстановления пароля.
* Экран должен содержать ссылку «Create account»
* При нажатии на ссылку «Create account» должен происходить переход к экрану регистрации.
* Система должна выполнять валидацию введенных данных.
* Проверка формата email должна быть реализована.
* Проверка сложности пароля должна быть реализована.
* Система должна отображать сообщения об ошибках валидации.
* Если логин или пароль неверны, система должна отображать понятное сообщение об ошибке.
* Сообщение об ошибке должно быть отображено в понятном месте.

## Основной экран:

* + 1. Навигационное меню:
* Основной экран должен содержать навигационное меню.
* Навигационное меню должно обеспечивать навигацию к разделам: «My orders», «Discounts», «Categories».
  + 1. Уведомления о новых событиях:
* На основном экране должны отображаться уведомления о новых событиях.
* Уведомления должны быть информативными и понятными.

## Профиль пользователя:

### 2.5.1. Общие требования:

* Безопасность: информация о пользователе хранится в безопасном режиме, доступ к данным ограничен.
* Интерфейс: экран профиля должен быть интуитивно понятным и удобным для пользователя.

### 2.5.2. Изменение пароля:

* Пользователь должен иметь возможность изменить свой пароль.
* Для изменения пароля должна быть предусмотрена проверка текущего пароля.
* Пользователь должен ввести новый пароль и подтвердить его
* Должна быть реализована проверка сложности нового пароля (минимальная длина, наличие символов).
* После успешного изменения пароля должна быть отображена информация об успешной смене пароля.

## 2.6. Отслеживание заказов:

### 2.6.1. История заказов:

* Отображение списка заказов.
* Для каждого заказа отображается:
  + Номер заказа.
  + Трек номер заказа.
  + Кнопка перехода на детали заказа.

### 2.6.2. Детальная информация о заказе:

* Номер заказа.
* Трек номер заказа.
* Список товаров в заказе с указанием:
  + Наименование товара.
  + Цены за единицу.
* Информация о скидках.
* Итоговая сумма заказа.

## 2.7. Скидки:

### 2.7.1. Отображение скидок:

* На в специальном разделе «Discounts» должен отображаться список всех актуальных скидок, доступных клиенту.
* Для каждой скидки должны отображаться следующие данные:
  + Название скидки.
  + Описание скидки.
  + Процент скидки.

### 2.7.2. Раздел «Discounts»:

В приложении должен быть раздел, содержащий список всех доступных пользователю скидок. Это могут быть как общие скидки, так и персональные, предоставленные конкретному пользователю.

Для каждой скидки должны отображаться те же данные, что и в разделе «Отображение скидок».

# Технические требования

## 3.1. Платформы:

Android:

* Поддержка актуальной и нескольких предыдущих версий Android (например, Android 13, Android 12, Android 11). Необходимо учитывать фрагментацию Android устройств.
* Адаптация интерфейса под различные размеры экранов и разрешения устройств Android.

## 3.2. Языки программирования:

Android: Android Studio Java, C#.

## 3.3. База данных:

* Реляционная база данных (RDBMS): MySQL. Подходит для хранения структурированных данных о заказах, товарах, пользователях. Обеспечивает ACID транзакции.
* Требования к базе данных:
  + Масштабируемость (возможность увеличения производительности и емкости базы данных по мере роста нагрузки).
  + Надежность (обеспечение сохранности данных и отказоустойчивости).
  + Безопасность (защита данных от несанкционированного доступа.).
  + Оптимизация запросов (эффективное выполнение запросов для быстрого доступа к данным).

## 3.4. API:

* Тип API: RESTful API
* Формат данных: JSON
* Требования к API:
  + Аутентификация и авторизация (защита API от несанкционированного доступа).
  + Версионирование (обеспечение обратной совместимости при изменении API.).
  + Документация (подробная документация API для облегчения интеграции).
  + Обработка ошибок (стандартизированные коды ошибок и сообщения для облегчения отладки.).

## 3.5. Push-уведомления:

* Требования к push-уведомлениям:
  + Надежность (гарантия доставки уведомлений пользователю).
  + Персонализация (возможность персонализации уведомлений).
  + Приоритезация (возможность установки приоритета уведомлений).

# Требования к интерфейсу пользователя

## 4.1. Интуитивно понятный интерфейс:

* Минимальное количество шагов для выполнения основных задач.
* Использование общепринятых иконок и символов.
* Логичная и предсказуемая навигация.
* Предоставление контекстной помощи и подсказок (при необходимости).

## 4.2. Адаптивность:

* Корректное отображение на различных размерах и разрешениях экранов мобильных устройств.
* Быстрая загрузка страниц и элементов интерфейса.
* Плавная анимация и переходы.
* Минимизация задержек и зависаний.

## 4.3. Дизайн:

* Соответствие фирменному стилю компании (цвета, шрифты, логотип).
* Согласованность элементов интерфейса (единый стиль кнопок, форм, списков).
* Использование визуальных акцентов для выделения важных элементов.
* Приятная и современная визуальная эстетика.

## 4.4. Требования к отдельным экранам:

### 4.4.1. Экран приветствия:

* Отображение логотипа компании.
* Автоматический переход к экрану авторизации/регистрации.

### 4.4.2. Экран авторизации/регистрации:

* Поля для ввода логина (email или номер телефона) и пароля.
* Кнопка «Continue».
* Ссылка «Forgot password?».
* Ссылка «Create account».
* Валидация введенных данных.
* Сообщение об ошибке при неправильном вводе данных.

### 4.4.3. Основной экран:

* Навигационное меню или панель вкладок для перехода к другим разделам приложения («My orders», «Discounts», «Categories»).
* Возможность поиска товаров по фильтрам.
* Уведомления о новых событиях.

### 4.4.4. Экран списка заказов:

* Отображение заказов в виде списка с кратким описанием (номер заказа, трек номер заказа).
* Возможность просмотра деталей заказа при нажатии на кнопку «Details».

### 4.4.5. Экран деталей заказа:

* Отображение полной информации о заказе.
* Четкое и структурированное представление информации.

### 4.4.6. Экран скидок:

* Отображение списка доступных скидок с кратким описанием (название, размер скидки, описание).
* Возможность активировать скидку.

# Требования к безопасности

## 5.1. Авторизация:

* Идентификация: Пользователь должен идентифицироваться по уникальному логину (номер телефона, email) и паролю.
* Безопасность пароля:
  + Пароли должны храниться в базе данных в хешированном виде. Минимальная длина пароля – 8 символов.
  + При регистрации предлагать пользователю оценить надёжность пароля и рекомендовать более сложные варианты.
* Обработка забытого пароля: реализовать безопасную процедуру восстановления пароля.
* Использовать только те данные, которые необходимы для функционирования приложения. Тщательно проверять и фильтровать данные, полученные от сторонних сервисов.

## 5.2. Защита данных:

* Шифрование конфиденциальных данных: шифровать все конфиденциальные данные, хранящиеся на устройстве (например, токены авторизации, личную информацию пользователя, данные банковских карт). Использовать надежные алгоритмы шифрования.
* Безопасное хранилище ключей: использовать KeyStore (Android) для хранения ключей шифрования. Эти хранилища обеспечивают аппаратную защиту ключей.
* Очистка данных из памяти: удалять конфиденциальные данные из памяти после использования. Не допускать хранения конфиденциальных данных в логах или других временных файлах.
* Защита от резервного копирования: рассмотреть возможность отключения резервного копирования конфиденциальных данных или шифрования резервных копий. Не хранить ключи шифрования в резервных копиях.
* Валидация пользовательского ввода: валидировать все данные, поступающие от пользователя, на стороне клиента и на стороне сервера. Проверять тип данных, длину, формат и диапазон значений.
* Обработка ошибок: обрабатывать ошибки таким образом, чтобы не раскрывать конфиденциальную информацию. Вести подробные логи ошибок для отладки и анализа проблем.

# Требования к тестированию

## 6.1. Юнит-тестирование:

* Цель: проверка корректности работы отдельных модулей, классов и функций приложения. Юнит-тесты должны быть быстрыми и изолированными.
* Объекты тестирования:
  + Классы и функции, реализующие бизнес-логику.
  + Компоненты пользовательского интерфейса.
  + Методы работы с базой данных.
  + Функции обработки данных, полученных из API.
* Покрытие кода: стремиться к максимально возможному покрытию кода юнит-тестами (не менее 80%).
* Примеры тестовых сценариев:
* Проверка правильности вычисления стоимости заказа.
* Проверка корректности форматирования данных.
* Проверка обработки исключений.
* Проверка валидации пользовательского ввода.
* Проверка логики авторизации пользователя.

## 6.2. Интеграционное тестирование:

* Цель: проверка взаимодействия различных компонентов приложения между собой. Интеграционные тесты проверяют корректность передачи данных между модулями и правильность работы системы в целом.
* Объекты тестирования:
  + Взаимодействие между компонентами пользовательского интерфейса.
  + Взаимодействие между приложением и базой данных.
  + Взаимодействие между приложением и API.
  + Процессы, включающие несколько модулей.
* Примеры тестовых сценариев:
  + Проверка корректности отображения списка заказов, полученных из API.
  + Проверка сохранения данных заказа в базе данных.
  + Проверка корректности работы процесса оформления заказа, включающего несколько этапов.
  + Проверка корректности работы системы push-уведомлений.
  + Проверка интеграции с платежным шлюзом.

## 6.3. Системное тестирование:

* Цель: проверка соответствия приложения всем требованиям технического задания и ожиданиям пользователей. Системные тесты проверяют работоспособность приложения в целом, а также его интеграцию с окружением.
* Объекты тестирования:
  + Все функции и возможности приложения.
  + Интерфейс пользователя.
  + Производительность.
  + Безопасность.
  + Удобство использования.
  + Совместимость с различными устройствами и операционными системами.
* Типы системного тестирования:
  + Функциональное тестирование: проверка соответствия функций приложения требованиям.
  + Тестирование производительности: проверка скорости работы приложения, времени отклика и устойчивости к нагрузкам.
  + Тестирование безопасности: проверка защиты приложения от несанкционированного доступа и утечек данных.
* Инструменты: ручное тестирование
* Примеры тестовых сценариев:
  + Проверка успешной авторизации пользователя с правильными учетными данными.
  + Проверка корректного отображения детальной информации о заказе.
  + Проверка отправки push-уведомления при изменении статуса заказа.
  + Проверка быстродействия приложения при большом количестве активных пользователей.
  + Проверка устойчивости приложения к DDoS-атакам.
  + Проверка работоспособности приложения на устройствах с различными разрешениями экрана.

## 6.4. Пользовательское тестирование:

* Цель: получение обратной связи от реальных пользователей о приложении. Пользовательское тестирование позволяет выявить проблемы с удобством использования, функциональностью и соответствием ожиданиям пользователей.
* Участники: группа пользователей, представляющая целевую аудиторию приложения.
* Процесс: пользователям предоставляется доступ к приложению, и они выполняют заранее определенные задачи или используют приложение в свободной форме. Пользователи сообщают о найденных проблемах и предоставляют свои отзывы.
* Критерии успешности:
  + Большинство пользователей успешно выполняют основные задачи.
  + Пользователи положительно оценивают удобство использования приложения.
  + Выявлено и исправлено большинство критических проблем.
* Методы сбора обратной связи:
* Опросы
* Интервью
* Анализ данных использования приложения
* Тепловые карты
* Примеры тестовых сценариев:
* Пользователь пытается найти информацию о своем заказе.
* Пользователь пытается оформить новый заказ.
* Пользователь пытается изменить свои личные данные.
* Пользователь пытается воспользоваться функцией поиска.
* Пользователь пытается обратиться в службу поддержки.

# Документация

## 7.1. Техническое задание:

Данный документ. Описывает цели проекта, функциональные и нефункциональные требования, технические требования, требования к безопасности, требования к тестированию и другие важные аспекты проекта.

## 7.2. Спецификация API:

Подробное описание API, используемого для взаимодействия мобильного приложения с backend-системой, включая эндпоинты, параметры, форматы данных и коды ошибок. Документация по API должна быть актуальной.

## 7.3. Руководство пользователя:

Документ, содержащий описание основных функций приложения и инструкции по его использованию. Руководство пользователя должно быть понятным для нетехнических пользователей.

## 7.4 Требования к документации:

* Актуальность: документация должна быть актуальной и отражать текущее состояние приложения. Регулярно обновлять документацию по мере внесения изменений в приложение.
* Полнота: документация должна содержать всю необходимую информацию для понимания и использования приложения.
* Читаемость: Документация должна быть написана понятным языком и хорошо структурирована.
* Формат: выбрать стандартный формат для документации.
* Контроль версий: использовать систему контроля версий (Git) для управления документацией.
* Доступность: документация должна быть доступна всем заинтересованным сторонам.

# Словарь

**Приложение** – программное обеспечение, которое разрабатывается и является конечным продуктом.

**Пользователь** – лицо, использующее приложение.

**Авторизация** – процесс подтверждения личности пользователя.

**Регистрация** – процесс создания учетной записи пользователя.

**Функциональность** – возможности, которые предоставляет приложение.

**Экран** – отдельный интерфейсный элемент, отображающий информацию и позволяющий пользователю взаимодействовать с приложением.

**Элемент интерфейса** – компонент, используемый на экране для взаимодействия с пользователем (например, кнопка, поле ввода, список).

**API** – интерфейс программирования приложений, позволяющий взаимодействовать с другими системами или компонентами.

**База данных** – система хранения и управления данными.

**Бэкенд** – серверная часть приложения, отвечающая за обработку данных, логику работы и взаимодействие с базой данных.

**Фронтенд** – Клиентская часть приложения, реализующая пользовательский интерфейс (отображение, взаимодействие с пользователем).

**Логин** – уникальное имя пользователя, используемое для входа в систему.

**Пароль** – секретный код, используемый для подтверждения личности пользователя.

**Валидация** – проверка данных, введенных пользователем, на соответствие заданным требованиям.

**UI/UX** – пользовательский интерфейс и пользовательский опыт. Дизайн и удобство использования приложения.

**HTTPS** – защищенный протокол передачи данных.

**Тестирование** – процесс проверки работы приложения для выявления ошибок.

**Баг** – ошибка в работе приложения.

**Релиз** – версия приложения, доступная для пользователей.

**Каталог товаров** – список товаров, доступных для продажи.

**Товар** – единица продукции, предлагаемая для продажи

**Карточка товара** – экран с подробной информацией о товаре.

**Корзина** – место, где пользователь может просмотреть и изменить выбранные товары перед оформлением заказа.

**Заказ** – заявка на покупку товаров.

**История заказов** – список предыдущих заказов пользователя.

**Статус заказа** – текущее состояние заказа (например, “Принят”, “В обработке”, “Отправлен”, “Доставлен”).

**Доставка** – процесс доставки товаров покупателю.

**Способ оплаты** – способ оплаты заказа.

**Скидка** – уменьшение цены товара.

**Профиль** – раздел с информацией о пользователе.

**Уведомление** – сообщение, информирующее пользователя о каком-либо событии.

**Спецификация** – подробное описание требований к функциональности и характеристикам приложения.

**Модуль** – отдельная, логически связанная часть приложения.

**REST API** – архитектурный стиль API.

**JSON** – формат обмена данными.

# Сроки и этапы разработки

|  |  |
| --- | --- |
| ***Дата*** | ***Виды выполненных работ*** |
| 28.04.25 | Вводный инструктаж. Распределение ролей. Планирование разработки. |
| 29.04.25 | Декомпозиция системы и поведения. Моделирование бизнес-логики. Моделирование интерфейса. |
| 30.04.25 | Реализация логики приложения для работы с веб-сервисом. Документирование результатов проекта. |
| 1.05.25 | Обработка сетевых запросов. Парсинг данных. |
| 2.05.25 | Тестирование разработки базы данных и приложения |
| 3.05.25 | Подготовка и защита портфолио, дневника и отчета по учебной практике. |

# Критерии приемки

## 10.1. Функциональные требования:

* Полнота реализации: все функции, описанные в техническом задании и функциональной спецификации, должны быть реализованы в полном объеме.
* Корректность работы: все функции должны работать корректно и соответствовать ожидаемому поведению. Не должно быть ошибок или сбоев.
* Тестовое покрытие: все ключевые функции приложения должны быть протестированы с использованием юнит-тестов, интеграционных тестов и системных тестов. Уровень покрытия кода юнит-тестами должен составлять не менее 80%.
* Соответствие UI/UX дизайну: интерфейс приложения должен соответствовать спецификации UI/UX и фирменному стилю компании.

## 10.2. Нефункциональные требования:

* Безопасность:
  + Отсутствие уязвимостей: приложение не должно содержать известных уязвимостей.
  + Безопасное хранение данных: конфиденциальные данные должны быть надежно защищены.
  + Соответствие стандартам безопасности: приложение должно соответствовать требованиям безопасности, указанным в техническом задании.
* Удобство использования:
  + Интуитивно понятный интерфейс: приложение должно быть простым и понятным в использовании.
  + Отсутствие ошибок UI/UX: интерфейс приложения не должен содержать ошибок и несоответствий.
* Совместимость: приложение должно быть совместимо с указанными в техническом задании версиями операционных систем и устройствами.
* Надежность: приложение должно работать стабильно и не должно вылетать или зависать.

## 10.3. Тестирование:

* Успешное прохождение всех этапов тестирования: приложение должно успешно пройти юнит-тестирование, интеграционное тестирование, системное тестирование и пользовательское тестирование.
* Устранение всех критических ошибок: все критические ошибки, выявленные в процессе тестирования, должны быть устранены.
* Устранение большинства значимых ошибок: большинство значимых ошибок, выявленных в процессе тестирования, должны быть устранены.
* Подтверждение исправления ошибок: все исправленные ошибки должны быть перепроверены и подтверждено их устранение.
* Отчет о тестировании: предоставлен полный отчет о тестировании, содержащий результаты всех этапов тестирования.

## 10.4. Документация:

* Наличие всей необходимой документации: должна быть предоставлена вся документация, указанная в техническом задании.
* Актуальность документации: документация должна быть актуальной и соответствовать текущей версии приложения.
* Полнота документации: документация должна содержать всю необходимую информацию для понимания и использования приложения.
* Качество документации: документация должна быть написана понятным языком и хорошо структурирована.