**Аналитическая записка**

«Информационная система для распределения заявок по ремонту в общежитии»

1. **Наименование проекта**

Информационная система для распределения заявок по ремонту в общежитии предназначена для приема заявок от студента в общежитии и распределения этих заявок по специалистам.

1. **Назначение проекта**

Назначение данной информационной системы заключается в создании автоматизированной системы приема заявок по ремонту в общежитии и распределения этих заявок по нужным специалистам, что значительно ускорит и облегчит выполнение рабочих в общежитии.

1. **Описание взаимодействия с потенциальным пользователем**

Пользовательские истории представлены в таблице 1.

Таблица 1 - пользовательские истории

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Роль** | **Действие** | **Цель** |
| **1** | Студент | Создать заявку на ремонт чего либо. | Обеспечить возможность создания и описания заявки для студента. |
| **2** | Сотрудник | Принять или отклонить заявку на ремонт чего либо. | Обеспечить возможность принять или отклонить заявку. |
| **3** | Сотрудник | Обновляет статус заявки на «Завершено» после выполнения ремонта | Обеспечить возможность изменения статуса заявки для сотрудника. |
| **4** | Администратор | Дать доступ к системе для создания заявок на ремонт. | Дать доступ студентом проживающим в общежитии. |
| **5** | Администратор | Дать доступ к системе для принятия или отклонения заявок на ремонт. | Дать доступ сотруднику обеспечивающему ремонт в общежитии. |
| **6** | Администратор | Создать в ручную заявку на ремонт в общежитии. | Обеспечить возможность создания и описания заявки для администратора. |

Пользовательские сценарии представлены в таблице 2.

Таблица 2 – пользовательские сценарии

|  |  |  |
| --- | --- | --- |
| **№** | **Пользователь** | **Система** |
| **1** | Студент создает и описывает заявку на ремонт в общежитии. | Добавляет заявку в базу данных и отправляет сотруднику который работает в данном корпусе. |
| **2** | Работник принимает заявку на ремонт в корпусе. | Меняет статус заявки и привязывает ФИО сотрудника к статусу заявки в базе данных. |
| **3** | Работник отклоняет заявку на ремонт в корпусе. | Отправляет заявку другому сотруднику который работает в данном или ближайшем корпусе. |
| **4** | Работник выполняет работу и меняет статус заявки на «Завершено». | Меняет статус заявки и привязывает на «Завершено» в базе данных. |
| **5** | Администратор меняет описание заявок | Меняет описание заявки в базе данных |

Диаграмма Use-Case представлена на рисунке 1.

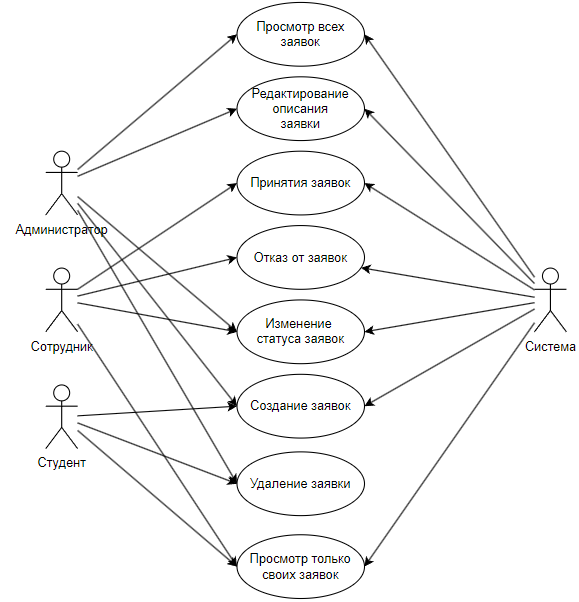


Рисунок 1 – Диаграмма Use-Case

Диаграмма Use-Case для информационной системы для распределения заявок по ремонту в общежитии отображает взаимодействие тремя основными категориями пользователей: Администратор, Сотрудник и Студент. Каждый из акторов имеет доступ к различным функциям системы, в зависимости от своей роли.

Акторы**:**

1. Администратор – обладает полными правами на управление системой, может просматривать, добавлять, изменять и удалять заявки.
2. Сотрудник – имеет доступ к просмотру только его заявок, а также может изменять статус заявки.
3. Студент – может создавать, а также редактировать комментарии к заявке.
4. **Основной функционал (функциональные требования)**

Функциональные требования представлены в таблице 3.

Таблица 3 – Функциональные требования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Название функции** | **Входные параметры** | **Выходные параметры** | **Описание функции** |
| **1** | Подача заявки. | Описание поломки в общежитии. | Сообщение «Ваша заявка принята». | Студент или администратор подает заявку на ремонт чего либо в общежитии. |
| **2** | Принятие заявки. | Нет. | Описание поломки в общежитии. | Сотрудник принимает заявку на ремонт в общежитии. |
| **3** | Отклонение заявки | Нет. | Сообщение «Заявка отклонена» | Сотрудник отклоняет заявку на ремонт в общежитии. |
| **4** | Просмотр заявки. | Нет. | Описание поломки в общежитии. | Функция отображается сотрудникам и администратору. |
| **5** | Редактирование заявки. | Измененное описание поломки в общежитии. | Подтверждение редактирования. | Студент или администратор редактируют заявку. |
| **6** | Изменение статуса заявки | Новый статус заявки. | Измененный статус заявки. | Сотрудник или администратор меняет статус заявки зависимости от выполненной работы . |
| **7** | Удаление заявки. | Нет. | Удаленная заявка. | Администратор удаляет заявку. |
| **8** | Простор всех заявок | Нет. | Список всех заявок. | Администратор может просматривать все заявки. |
| **9** | Просмотр только своих заявок. | Нет. | Список своих заявок. | Сотрудник и студент могут просматривать заявки в которых они принимали участие. |

1. **Возможные аналоги**

**Телеграм-бот «OkdeskBot»**

OkdeskBot - телеграмм бот для службы поддержки, приёма и обработки заявок.



Рисунок 2 - OkdeskBot

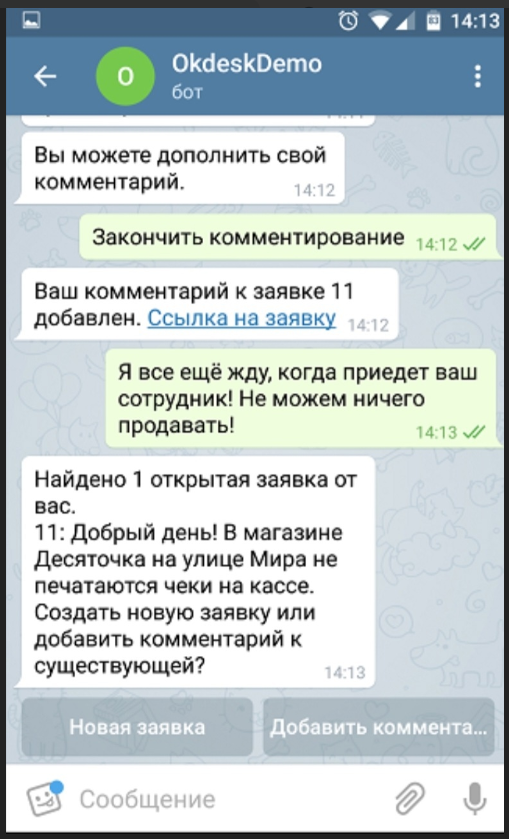


Рисунок 3 - OkdeskBot

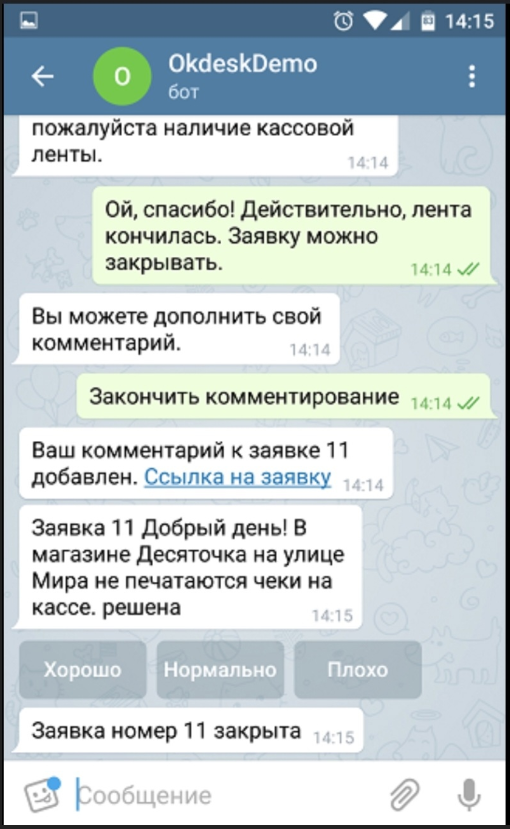


Рисунок 4 - OkdeskBot

Из достоинств можно выделить удобство использования. OkdeskBot предлагает простой и интуитивно понятный интерфейс, что делает его удобным в использовании для всех пользователей, так же быстрое получение ответов, благодаря автоматическому обработчику заявок, OkdeskBot способен быстро и эффективно отвечать на вопросы пользователей.

Из недостатков можно выделить, ограничение функционала, так как OkdeskBot предлагает базовый функционал для работы с заявками, что может ограничить возможности управления более сложными задачами.

**Телеграмм бот «Аpple.Service»**

«Аpple.Service» **-** телеграмм бот для сбора заявок на ремонт техники Apple.

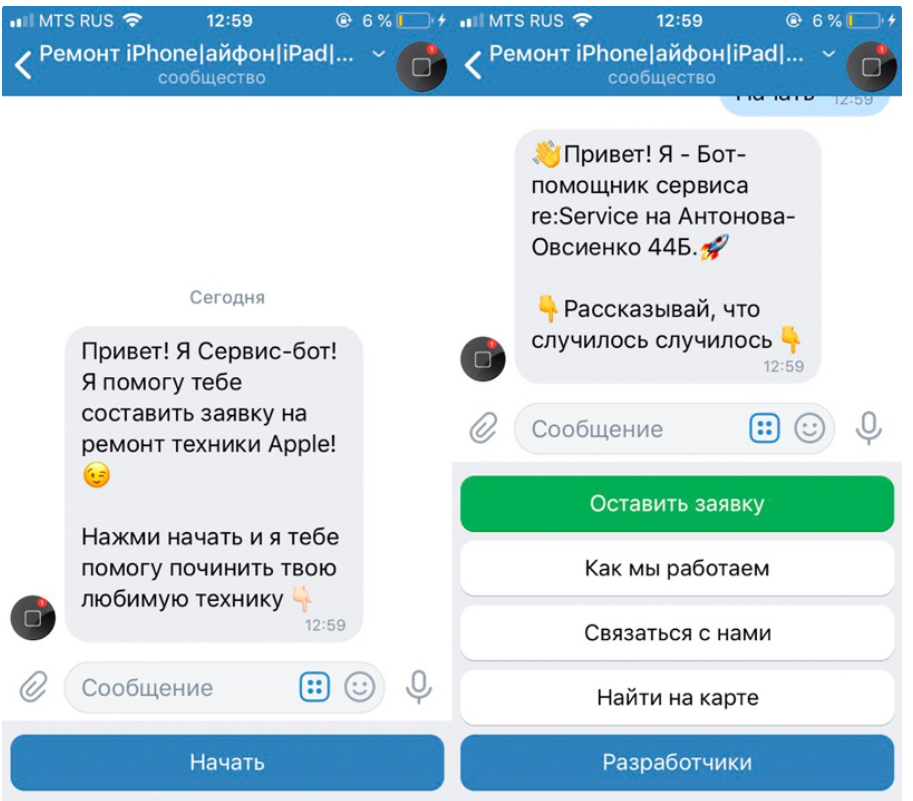


Рисунок 5 - Apple.Serice

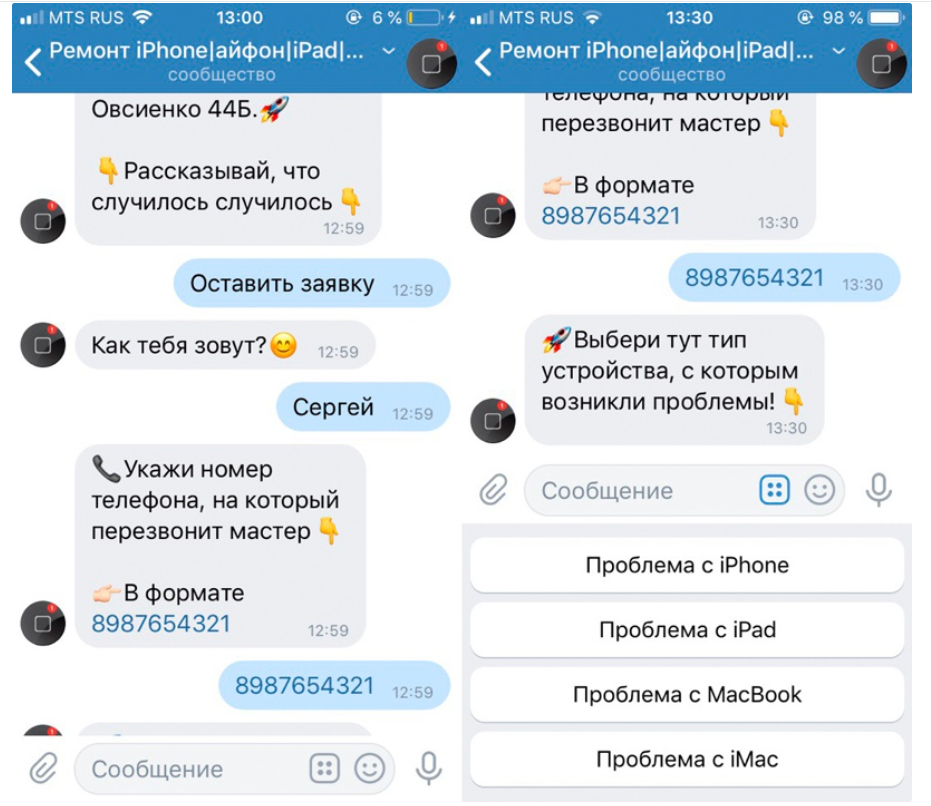


Рисунок 6 - Apple.Serice

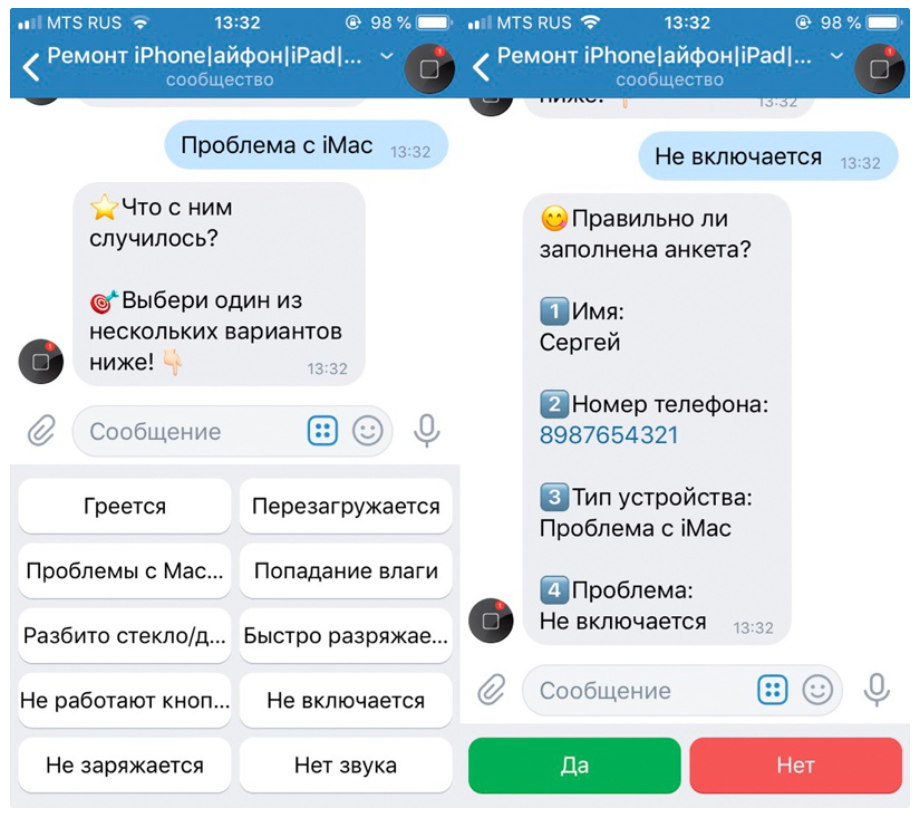


Рисунок 7 - Apple.Serice

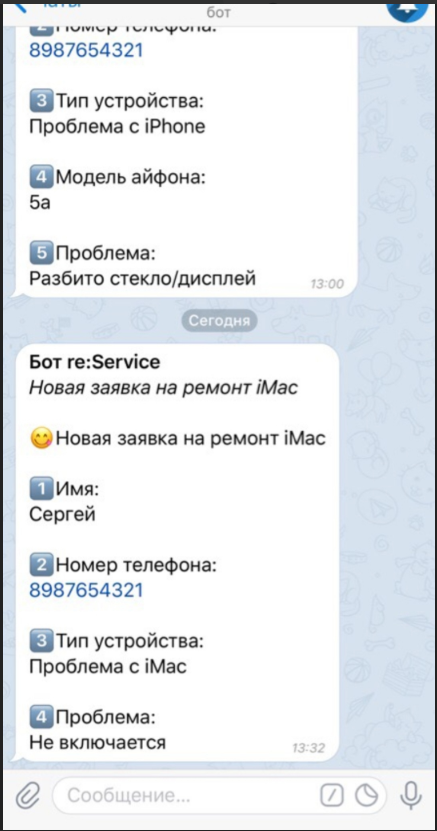


Рисунок 8 - Apple.Serice

Из достоинств бота можно выделить простоту использования. Бот позволяет пользователям легко и быстро подать заявку на ремонт техники Apple. Все необходимые поля присутствуют, и нет необходимости дополнительно взаимодействовать с оператором, также он автоматически принимает заявки, что позволяет оперативно начать их обработку.

Из минусов бот предназначен только для сбора заявок на ремонт техники Apple, и не предоставляет другие дополнительные услуги или возможности.

1. **Предлагаемые к использованию технологии и модели**

**Бэкенд:**

* Python: язык программирования, известный своей простотой и богатой экосистемой библиотек, что делает его идеальным для разработки серверной части приложения.

**База данных:**

* PostgreSQL: реляционная база данных, обеспечивающая надежное хранение данных и поддержку сложных запросов.