**«УТВЕРЖДАЮ»**

Руководитель

<Компания-заказчик>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

**«УТВЕРЖДАЮ»**

Руководитель

<Компания-исполнитель>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

**Техническое задание**

На разработку сайта для компании Practica

**«СОГЛАСОВАНО»**

<Представители заказчика>

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

**«СОГЛАСОВАНО»**

<Представители исполнителя>

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

Брянск 2024

**Содержание**

[История изменений 3](#_Toc169966400)

[Название проекта 4](#_Toc169966401)

[Цель проекта 4](#_Toc169966402)

[1. Функциональные требования 4](#_Toc169966403)

[2. Администраторская часть 5](#_Toc169966404)

[3. Технические требования 5](#_Toc169966405)

[4. Не функциональные требования 6](#_Toc169966406)

[5. Фазы разработки 7](#_Toc169966407)

[6. Критерии приемки 7](#_Toc169966408)

[7. Команда 7](#_Toc169966409)

[Приложение А 8](#_Toc169966410)

# История изменений

|  |  |  |  |
| --- | --- | --- | --- |
| Версия | Дата | Автор | Примечание (суть изменений) |
| 1.0 | 20.06.2024 | Коткин А. А. | Первоначальная версия ТЗ |
| 1.1 | 21.06.2024 | Жирнов М. О. | Добавление архитектуры сервиса бронирования (приложение А) |
| 1.2 | 22.06.2024 | Коткин A. A.  Жирнов М. О. | Редактирование документации ТЗ |

|  |  |
| --- | --- |
| **Техническое задание: Сервис для бронирования столиков в ресторане** | |
| Название проекта | Сервис для бронирования столиков в ресторане |
| Цель проекта | 1. Разработать веб-сервис, который позволит клиентам бронировать столики в ресторане онлайн. 2. Обеспечить удобный и эффективный процесс бронирования для клиентов и персонала ресторана. |
| 1. Функциональные требования: | |
| 1.1.Клиентская часть | Создание аккаунта с email/пароль |
| 1.2.Ресторан | 1. Выбор ресторана 2. Просмотр информации о ресторане: описание, меню, фотографии, отзывы |
| 1.3.Бронирование столика | 1. Выбор даты и времени бронирования. 2. Выбор количества персон, типа столика (внутренний/наружный, вид). 3. Возможность добавления особых пожеланий (например, некурящий столик). |
| 1.4.Отмена бронирования | Возможность отмены бронирования |
| 1.5.Управление бронированиями | Просмотр истории бронирований, отмена бронирования |
| 1.6.Отзывы | Возможность оставлять отзывы о ресторане |
| 2. Администраторская часть: | |
| 2.1.Управление рестораном | 1. Добавление/редактирование информации о ресторане (название, описание, меню, фотографии, время работы). 2. Управление столами (количество, типы, расположение). 3. Установка цен на блюда. |
| 2.2.Управление бронированиями | 1. Просмотр списка бронирований, изменение статуса бронирования (подтверждено/отменено). 2. Отправка уведомлений клиентам о подтверждении/отмене бронирования |
| 2.3.Аналитика | 1. Статистика по бронированиям (количество, время, тип столика). 2. Отчеты о популярности блюд, пиковых часах работы. |
| 3. Технические требования: | |
| 3.1.Язык программирования | 1. Python (Django/Flask), 2. Node.js (Express), 3. Java (Spring Boot), 4. PHP (Laravel). |
| 3.2.База данных | 1. MySQL, 2. MongoDB |
| 3.3.Фреймворк | Выбор фреймворка зависит от выбранного языка программирования |
| 3.4.Web-сервер | 1. Nginx, 2. Apache, 3. uWSGI |
| 3.5.Интерфейс | 1. HTML, 2. CSS, 3. JavaScript |
| 3.6.API | RESTful API для взаимодействия между клиентской и серверной частью |
| 3.7.Безопасность | Хранение паролей в зашифрованном виде. |
| 3.8.Тестирование | 1. Unit-тесты для серверной части. 2. Функциональное тестирование клиентской части |
| 4. Не функциональные требования: | |
| 4.1.Надежность | Сервис должен быть доступен 24/7 |
| 4.2.Производительность | Сервис должен обрабатывать запросы быстро и эффективно |
| 4.3.Масштабируемость | Сервис должен быть готов к увеличению нагрузки |
| 4.4.Безопасность | Сервис должен быть защищен от атак злоумышленников |
| 4.5.Удобство использования | Сервис должен быть интуитивно понятным и простым в использовании |
| 4.6.Доступность | Сервис должен быть доступен с любых устройств (компьютеры, смартфоны, планшеты) |
| 5. Фазы разработки: | |
| 5.1.Анализ и планирование | Определение функциональных и технических требований |
| 5.2.Проектирование | Разработка архитектуры приложения, выбор технологий |
| 5.3.Разработка | Написание кода, создание интерфейса |
| 5.4.Тестирование | Проведение различных типов тестирования |
| 5.5.Развертывание | Запуск сервиса на сервере |
| 5.6.Обслуживание | Мониторинг работы сервиса, исправление ошибок, добавление новых функций |
| 6. Критерии приемки: | |
| 1. Сервис работает стабильно и без ошибок. 2. Сервис обладает необходимым функционалом. 3. Сервис удобен в использовании. 4. Сервис соответствует требованиям безопасности. | |
| 7. Команда**:** | |
| 7.1. Разработчики (frontend, backend) | 2 человека |
| 7.2. Тестировщики | 1 человек |
| 7.3. Дизайнеры | 2 человека |

Наглядное представление работы сервиса со стороны пользователя представлено на диаграмме деятельности(рисунок 1, Приложение А).

# Приложение А

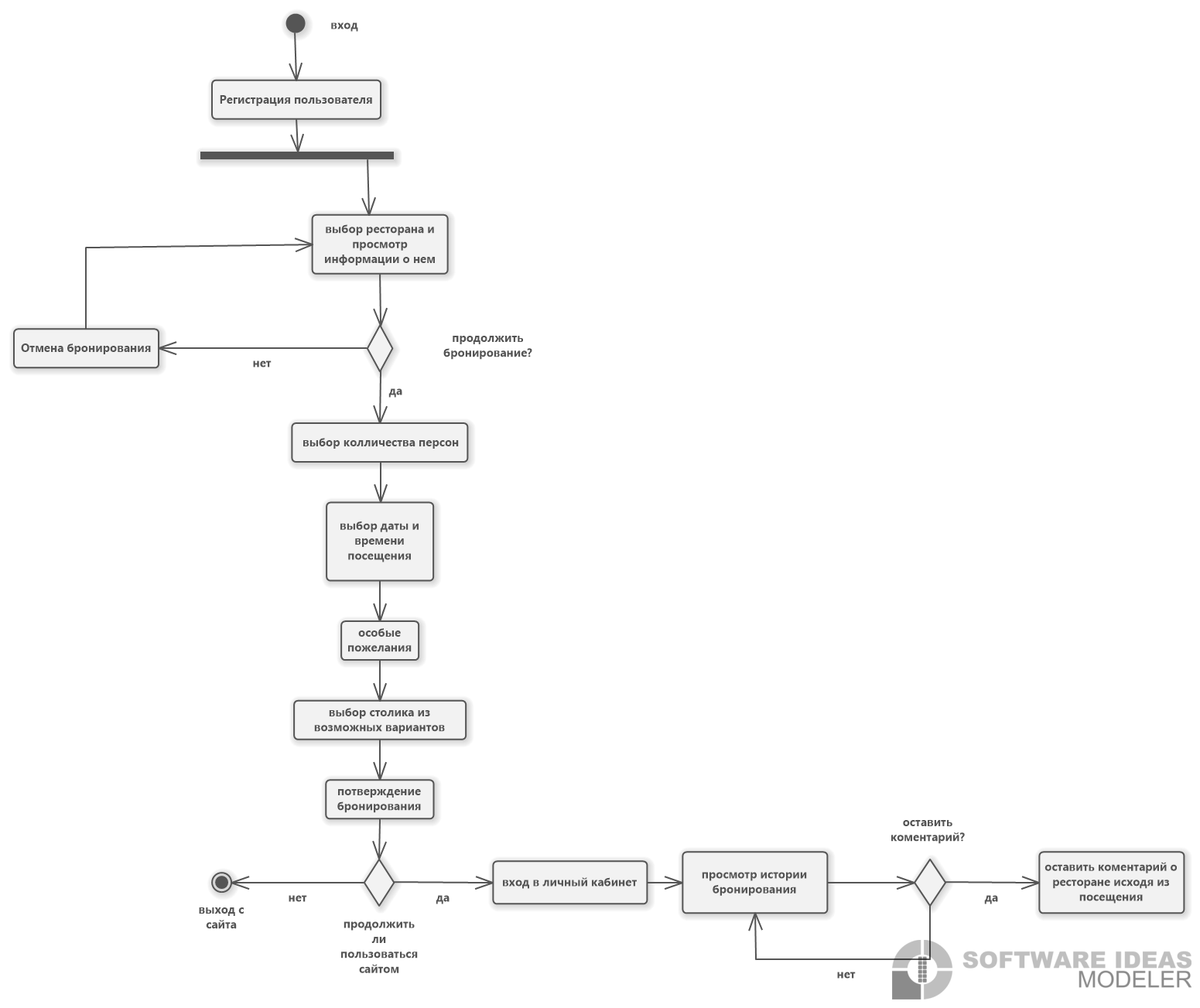


Рисунок 1 - Диаграмма деятельности пользователя сервисом бронирования