

ОПИСАНИЕ ТЕХНИЧЕСКОЙ АРХИТЕКТУРЫ
программного обеспечения
"Система бронирования автомобилей bookingcar.ru"

ПРАВООБЛАДАТЕЛЬ:
ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ
«МЕДИА-РЕШЕНИЯ»
ИНН 7204205305

Список сокращений и обозначений

API – (англ. Application Programming Interface) программный интерфейс приложения

HTTP – (англ. HyperText Transfer Protocol) протокол прикладного уровня передачи данных

БД – база данных

ПО – программное обеспечение

СУБД – система управления базами данных

1. Общие сведения

Документ содержит описание технической архитектуры программного обеспечения "Система бронирования автомобилей bookingcar.ru" (далее ПО).. ПО представляет собой интернет-ресурс, который помогает пользователю получать информацию по аренде автомобилей в городах России с помощью контента предоставленного Пользователями и Администрацией Сайта.

Проектирование и описание сценариев использования системы Сценарии по ролям в представлении вариантов использования.

Выделим общие сценарии, доступные всем пользователям:

1. Просмотр контента
2. Поиск и фильтрация автомобилей
3. Каталог автомобилей
4. Бронирование автомобилей

Сценарии пользователя с ролью «Администратор системы»:

1. Войти в панель администратора
2. Редактировать профиль
3. Добавить, редактировать каталог автомобилей
4. Создать Пользователя-Организацию
5. Модерировать аккаунт Пользователя-Организации
6. Добавить, редактировать любой контент системы
7. Посмотреть и редактировать заявки на бронирование автомобилей

Сценарии пользователя с ролью «Пользователь-Организация»:

1. Войти в систему
2. Редактировать свой профиль
3. Посмотреть, редактировать автомобили
4. Посмотреть, обработать входящие заявки

2. Структура программного обеспечения

2.1. Взаимодействие компонентов

ПО состоит из следующих компонентов:

«**Веб-сервис**» – выполняет функции авторизации пользователя-организации в ПО, добавление и редактирование автомобилей Организации;

«**Веб-интерфейс**» – предоставляет функционал для ввода данных всех пользователей, просмотра каталога автомобилей, поиска и фильтрации автомобилей, бронирования автомобилей;

«**Библиотека интеграции**» – реализует подключение к базе данных, выполняет поиск объектов, файлов мультимедиа и документов, загружает файлы для просмотра/тестирования;

«**База данных**» – выполняет функции по хранению информации с результатами поиска для последующего быстрого доступа.

2.2. Функциональная структура

ПО включает следующие функциональные модули:

«**Ядро системы**» – представляет собой веб-сервис, в котором реализован веб сервер, обеспечивающий подключение пользователей по протоколу HTTP/HTTPS. Веб-сервер состоит из следующих базовых классов:

- Cache – класс реализующий кеширование и хранение пользовательских данных

- Database – класс работы с базой данных

- Session – класс работы с сессиями пользователей

- Stat – класс работы статистики

- Template – класс обработки визуальной части компонентов

- User – класс работы с пользователями системы

«**Модуль веб-интерфейса**» – формирует пользовательские формы, обеспечивающие выполнение следующих действий пользователя:

- авторизация;

- ввод данных пользователя;

- поиск и фильтрация автомобилей;

- бронирование автомобилей;

«**Модуль авторизации пользователей в ПО**» – реализует следующий функционал:

- проверка зарегистрированных пользователей;

- ведение записи журнала действий пользователей. Вся информация хранится в базе данных;

«**Модуль просмотра каталога автомобилей**» – отображает автомобили удовлетворяющие условиям фильтрации. Модуль загружает автомобили из кэша, куда они загружаются по запросу из базы данных;

«**Модуль бронирования**» – сохраняет результаты заявки на бронирование автомобиля в базе данных для последующего отображения их пользователям-Организациям, сохраняет данные об автомобиле и датах бронирования.

3. Структура базы данных программного обеспечения

ПО поддерживает работу со следующими системами управления базами данных (СУБД):

- MariaDb 10.1

4. Применяемые технологии разработки программного обеспечения

Основной сервис ПО реализован как SAAS приложение на сервере Правообладателя.

Логика приложения реализована на PHP 7.4.

Логика хранения данных приложения реализована MariaDb.

Логика обработки клиент-серверного взаимодействия реализована Apache.

Логика кеширования реализована на Memcache.

Логика шаблонизатора реализована на Smarty 4.2.