Функциональный профиль для beActive

Андрей Коновалов, 073

1 beActive

beActive - это менеджер личного расписания. Пользователь может создать аккаунт и манипулировать собственным расписанием, например добавлять или удалять события. Расписание пользователей хранится на централизованном сервере.

Сервер осуществляет две функции: реализует API для работы с расписанием пользователия и предоставляет пользователю стандартный веб-клиент для работы с этим API (веб-сайт). У сторонних разработчиков должна быть воможность реализовать собственный клиент (например мобильное приложение) для взаимодействия с сервером. Это требование порождает необходимость открытости и доступности API для взаимодействия с сервером.

Требования к открытости системы:

- открытые форматы взамодействия клиента с сервером;
- переносимость сервера;
- легкая управляемость сервера.

2 Функциональный профиль

В качестве модели для функционального профиля была выбрана модель MUSIC/CCITT.

2.1 Внешняя среда

Между сервером и клиентами требуется сетевое соединение (описано в разделе коммуникации).

2.2 Платформа

Для запуска серверной части системы требуется операционная система, совместимая с

- интерпретатором языка Python;

- стандартом базы данных, описанном в разделе базы данных и информация;
- стандартом сетевого соединения, описанном в разделе коммуникации.

Для запуска клиентской части системы требуется операционная система, совместимая с

- веб-браузером, совместимым со стандартами, описанными в разделе пользовательский интерфейс;
- стандартом сетевого соединения, описанном в разделе коммуникации.

2.3 Информация

Вся пользовательская информация хранится на сервере. Для хранения информации используются базы данных поддерживающие открытый стандарт SQL:2011 (ISO/IEC 9075:2011), совместимые с реализацией операционной системы платформы.

2.4 Коммуникация

Для коммуникации между сервером и клиентами используется TLS соединение (стандарт описан в RFC 5246). Для шифрования передаваемых данных используются алгоритмы, совместимые со стандартом TLS соединения. Данные по TLS соединению передаются с использованием протокола HTTP (описан в RFC 7230, 7231, 7232, 7233, 7234, 7235), в соответствии со стандартом HTTPS (HTTP over TLS, описан в RFC 2818).

Для взаимодействия клиента с API сервера, используется формат JSON (RFC 7159) поверх HTTPS.

Для коммуникации серверного приложения с базой данных используется стандарт, описанный в реализации базы данных.

2.5 Пользовательский интерфейс

В качестве стандартного пользовательского интерфейса используется веб-сайт, использующий HTML (RFC 2854), CSS (RFC 2318) и JavaScript (RFC 4329). Сторонним разработчикам предоставляется возможность реализовать собственный интерфейс для взаимодействия с сервером, никаких ограничений на который не накладывается.

2.6 Управление

Контроль сервера осуществляется вручную администратором посредством инструментов операционной системы (стандарт IEEE Std 1003.2-1992).

Авторизация пользователей осуществляется с использованием стандарта OAuth 2.0.

3 Реализация

Для запуска серверной части системы используется операционная система Ubuntu 14.04. В качестве интерпретатора используется CPython, а в качестве базы данных - MySQL 5.6. Серверное приложение реализовано на языке Python, с использованием фрейворка Django. HTTPS соединение реализуется с помощью библиотеки OpenSSL, как часть веб сервера Nginx.

Для взаимодействия со стандартным пользовательским интерфейсом используется любой из современных браузеров (Chrome, Firefox, ...).