**Отчет о выполнении лабораторной работы №1**

Выполнил: студент группы ЗБ-ПИ20-2, Петренко Артемий.

В качестве документации для работы будет использоваться требования к API, описанные мной в рамках курсовой работе на 3 курсе. Приведу фрагмент, который будет использоваться для лабораторной работы.

**Текст требований:**

От серверной части разрабатываемого программного решения требуется выполнение нескольких ключевых задач:

* регистрация пользователя;
* авторизация пользователя;
* получение списка доступных тестов для заданного пользователя;
* получение теста по его id;
* добавление теста;
* удаление теста;
* сохранение результата прохождения теста;
* просмотр всех результатов прохождения теста;

При этом все функции API, кроме регистрации пользователя должны быть доступны только авторизованным пользователям. Это необходимо для того, чтобы никто посторонний не мог повлиять на информацию, предоставляемую API для пользователей. Для зарегистрированных пользователей необходимо предусмотреть две роли: ученик и учитель. Часть функционала API должна быть доступна только для пользователей с ролью учитель, это такие функции как: создание, удаление и просмотр всех результатов прохождения теста.

Для доступа к каждой из требуемых функций необходимо предусмотреть адрес, или как это чаще называется в терминологии API маршрут, обращаясь к которому клиент будет получать ту или иную информацию. Определим маршруты, которые необходимо реализовать для решения поставленных задач:

Таблица 1 - Маршруты, разрабатываемого API

|  |  |  |
| --- | --- | --- |
| Адрес и вид запроса | Функциональное назначение | Описание данных |
| /registration  POST | Регистрация пользователя | Вход: POST с формой, содержащей информацию, описывающую нового пользователя.  Выход: статус выполнения - успешно или код ошибки. |
| /token  POST | Получение токена авторизации | Вход: авторотационные данные пользователя  Выход: уникальный токен, используемый для подтверждения авторизации пользователя |
| /verification  GET | Подтверждение адреса электронной почты, используемого при регистрации | Вход: уникальный код, присылаемый на email пользователя  Выход: статус выполнения – успешно или код ошибки |
| /tests  GET | Получение информации о всех доступных тестах | Вход: токен пользователя  Выход: получения списка всех тестов, доступных для текущего пользователя |
| /test/{id}  GET | Получение информации о тесте | Вход: идентификатор требуемого теста  Выход: список заданий, содержащихся в тесте |
| /question/{id}  GET | Получении информации о задании | Вход: идентификатор требуемого задания  Выход: информация, представляющая задание теста: вопрос и варианты ответа |
| /question  POST | Добавление нового задания в систему | Вход: информация, необходимая для представления задания  Выход: статус выполнения – успешно или код ошибки |
| /test  POST | Добавление нового теста | Вход: информация, необходимая для представления теста  Выход: статус выполнения – успешно или код ошибки |
| /test/{id} | Удаление теста | Вход: идентификатор, удаляемого теста  Выход: статус выполнения – успешно или нет |

**Цель работы:** Целью данной лабораторной работы является знакомство с порядком тестирования документации описывающий требования к разработке программного обеспечения.

**Описание тестируемой документации:** Документация описывает требования к разрабатываемому API, представлен набор функциональных требований, описывающих действия, которые должен будет способен выполнять реализованный сервис.

В документации представлена таблица, подробно описывающая маршруты API, которые необходимо реализовать, какой тип должен иметь запрос (POST\GET) и какие данные он будет принимать и возвращать.

**Описание критериев качества тестируемой документации:** Документация можно отнести к категории, выполненной в соответствие с внутренними правилами команды (в данном случае университета, так как она является частью курсовой работы). Оценим качество документации в соответствии с набором стандартов ISO/IEC 9126-2, 9126-3, 9126-4:

* Внутреннее качество программного обеспечения можно оценить, как удовлетворительное, так как соблюдается логика описания функциональных требований, представлен достаточный объем информации для разработки API в читаемом виде.
* Внешнее качество не является удовлетворительным, так как документация и программное средство ее описывающее реализовано без учета нормативов ГОСТ и ISO.
* Качество процесса разработки ПС по данной документации удовлетворительно, так как функциональные требования описаны достаточно подробно для реализации API.

**Описание и обоснование метода тестирования документации:**

Для данного случая стоит применить метод тестирования документации именуемый рецензированием, так как не имеется тест-кейсов или функциональных набросков для осуществления других типов тестирования. Основным материалом, с которым можно работать является сам текст документации, и на его основании и будем делать выводы.

**Список несоответствий в документации критериям качества с указанием номеров пунктов исходной документации, цитированием части документации и конкретными рекомендациями к исправлению:**

Документация обладает достаточной для реализации ПС полнотой, но могла бы иметь и более полное и подробное описание типов данных и значений, которые должно получать API, например:

|  |  |  |
| --- | --- | --- |
| /test  POST | Добавление нового теста | Вход: информация, необходимая для представления теста  Выход: статус выполнения – успешно или код ошибки |

Таким образом в документации задаются требования к реализации маршрута API, несмотря на то что проект подразумевал реализацию на языке программирования с динамической типизаций, для полноты предоставления было бы правильно описать подробно структуру данных, которой представляется «тест». В контексте документации под этим понимается набор из вопросов с ответами, на которые будет отвечать участник. Если бы в документации было написано:

Тест представляет собой Вопрос (ссылка не текстовый тип данных длинной до 256 символов), Ответы (ссылка на n число структур, являющихся строковым типом данных длинной в 128 символов.

В таком случае документация имела бы уже более высокий уровень полноты описания, а если добавить обозначение типов данных строк и числе в соответствии с форматами ISO, например, указания стандарта кодирования строк или представления чисел, то документация имела бы лучший показатель стандартизации.

Для полноты описания требований также необходимо указывать с какой СУБД должно взаимодействовать API, с указанием запросов, которые необходимо реализовать, это необходимо для того чтобы требования к реализации БД не противоречили требованиям к API. Данные требования могут казаться избыточными для работы, реализуемой целиком и полностью одним человеком, но могут быть полезными для дальнейшего расширения функционала ПС и поддержки проекта в будущем другими специалистами.

**Выводы по работе:** в рамках данной лабораторной работы познакомились с порядком тестирования документации описывающий требования к разработке программного обеспечения. Сделал выводы о написании требований к API, реализованного в рамках курсовой работы.