

Добрый вечер уважаемые члены комиссии. Меня зовут Шашкин Юрий Анатольевич и я представляю вашему вниманию выпускную квалификационную работу бакалавра по теме «Система контроля знаний языка запросов к базе данных».

Для начала давайте рассмотрим, кому данная система будет полезна, и кто является ее целевой аудиторией. Система представляет потенциальную ценность для всех, кому требуется тем или иным образом проверять знания языка sql, будь то менеджеру для оценки знаний кандидата на техническом собеседовании, или преподавателю для контроля знаний учащихся. Внедрение системы позволит упростить создание тестов и проверки результатов тестирования, а само тестирование перенести в онлайн. Кроме того, система проверяет именно результаты выполнения скриптов, а не сами скрипты, ведь правильных решений может быть много.

Первым этапом работы над данным продуктом был анализ аналогов разрабатываемой системы. На рынке представлено не так много систем, которые выполняют аналогичные функции. Наиболее подходящим под описание оказался англоязычный портал codewars, который содержит функционал для обучения программированию и проведения собеседований.

В результате проведенного анализа были выделены основные функциональные особенности, которые требуются системе. На слайде представлен только функционал менеджера с максимальным приоритетом реализации, остальные функции можно посмотреть в конце первой главы расчетно-пояснительной записки. На слайде вы видите отличительные особенности разрабатываемой системы, которые выделены красным, а именно возможности: загружать картинки в условие задания, создавать объектов базы данных с помощью файла со скриптами, напрямую подключаться к БД и другие.

Для анализа возможной автоматизации предметной области разработана функциональная диаграмма IDEF0. На данном слайде представлена начальная диаграмма с точки зрения ресурс-менеджера.

Далее мы видим последовательную детализацию бизнес-процессов и выделение основных шагов тестирования знаний: просмотр существующих заданий, создание нового задания, удовлетворяющего требованиям к вакансии, формирование комплекта заданий, отправка данного комплекта, проведение тестирования, которое происходит, по сути, без участия менеджера, и получение результатов после проверки задания системой

Создание нового задания является самым сложным шагом в системе. Первым делом менеджеру необходимо создать новую схему БД написав скрипт на языке SQL. Этот скрипт выполняется системой и создает все необходимые объекты БД необходимые для постановки задачи. Впоследствии эту схему можно будет модифицировать и использовать повторно. После этого требуется написать условие задания, в которое чаще всего входит схема БД в графическом виде и эталонное решение. Осталось только ввести некоторые настройки, а именно: указать время на выполнение задания, указать метки для задания (например, сложность задания) и сохранить итоговый вариант.

Создание комплекта заданий состоит в выборе заданий для включения в комплект и расположения их в правильном порядке.

Получение результатов состоит из нескольких этапов. Первый – сверка решения кандидата с эталонным решением путем выполнения обоих запросов и проверкой, что результаты этих запросов идентичны. Второй – сбор статистики по комплекту заданий и сохранение результатов.

На основе бизнес-процессов была получена схема базы данных, которая представлена на слайде. По данной схеме видно, какие сущности присутствуют в базе данных и какие связи между ними.

В системе существует несколько сложных алгоритмов, которые были разобраны в рамках расчетно-пояснительной записки. В рамках доклада рассмотрим алгоритм создания объектов БД. Существует два основных процесса – создание новой схемы и новых объектов в ней и модификация старой. При создании новой схемы будут созданы пользователи, под которыми будут выполняться скрипты. Один – для создания объектов, другой – для выполнения тестовых заданий.

У пользователя есть возможность загрузить SQL файл со скриптами для создания объектов базы данных или подключиться к БД напрямую и создавать или модифицировать объекты вручную.

После создания всех объектов схемы пользователь может создавать задания.

После проработки бизнес-процессов были спроектированы формы интерфейса. На данном слайде представлена форма с комплектами заданий и отправкой одного из них кандидату.

Здесь представлена форма для написания задания. В левой части экрана представлено поле для написания условия задания, в правой – для написания эталонного решения. Внизу экрана у менеджера есть возможность загрузить изображение, которое будет приложено к заданию.

Подготовительным этапом проектирования классов системы является проектирование пакетов. Так как диаграмма классов получилась достаточно громоздкой, то в рамках доклада рассмотрим диаграмму пакетов. На данной диаграмме можно увидеть, что в системе присутствуют два больших пакета – контроллеры и модели. В моделях находятся объекты, которые связывают БД и систему. Контроллеры – занимаются обработкой данных и их подготовкой к выводу на экран пользователю. Большая часть пакетов выполняют функции, которые могут быть определены по названию. Отдельно был выделен пакет для соединения с БД, который используется для создания объектов базы данных и запуска решений.

Технологический стек получился не сильно большим: JAVA и JAVA EE для серверной части разработки, MS SQL Server 2016 в качестве СУБД для хранения данных, React js для работы с формами интерфейса, библиотека Hibernate для работы с базой данных и библиотека ACE для формирования блоков интерфейса, в которые будет вводиться sql код решения.

Далее вы видите пример кода, написанного для работы в некоторых технологических компонентах.

Тестирование было разделено на несколько этапов: автоматическое тестирование, тестирование безопасности, тестирование интерфейса пользователя и тестирование компонентов SQL.

В ходе автоматического тестирования были написаны программные тесты для проверки невозможности доступа неавторизованного пользователя к формам менеджера, что данные корректно сохраняются в БД и функциональность классов работает ожидаемо.

Тестирование безопасности оказалось очень важным, так как система в некоторых местах выполняет запуск SQL скриптов непосредственно на сервере без каких-либо проверок. Тестирование проводилось с помощью команды `DROP TABLE USERS`, что подразумевает полное удаление таблицы пользователей из системы. Как видно из данного слайда, система успешно прошла проверку безопасности.

Тестирование интерфейса пользователя заключалось в показе системы контрольной группе из 3 пользователей и сбора из отзывов о работе системы. Кроме того, пользователи пытались вводить некорректные данные на различных этапах и их действия никак не отразились на работоспособности системы.

Тестирование компонентов SQL заключалось в использовании различных конструкций языка и его процедурных расширений в рамках эталонного запроса и запроса кандидата. Как видно из результатов тестирования, процедурные расширения языков не всегда корректно обрабатывают. Это является не самой критичной ошибкой, так как система рассчитана именно на несложные задания, которые содержат именно «чистый» SQL. Данная проблема может быть решена в будущем, если перевести проверку правильности запросов на хранимые процедуры.

В качестве перспектив развития могут быть выделены два основных направления – интеграция с системами электронного обучения, в частности moodle, что позволит значительно расширить целевую аудиторию системы до учебных заведений, и развитие других видов тестирования, среди которых задания с выбором ответа и теоретические задания с развернутым ответом.

Спасибо за внимание!