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. На данном слайде можно увидеть примерный алгоритм работы менеджера при организации технического собеседования. Менеджер разрабатывает задание, потом готовит его решение, тратит время на проведение тестирования, а обычно это около часа, пока человек решает данное задание, затем вручную проверяет данное задание и только после этого менеджер может делать выводы и том, подходит сотрудник на эту должность или нет.

После внедрения системы проведение собеседования можно будет перенести в онлайн формат, что позволит, по сути, избавиться от шага с непосредственным участием менеджера в проведении собеседования, так как кандидат будет проходить тестирование самостоятельно. Тем самым мы экономим огромное количество времени менеджеру, которое он может потратить на проекты компании. Кроме того, система позволяет свободно переиспользовать задания для отправки другим кандидатам.

С точки зрения кандидата, процесс собеседования затягивается из-за того, что ему приходится тратить время на дорогу до компании, ожидание менеджера. Кроме того, бывает достаточно сложно назначить собеседование на такое время, которое бы устраивало и менеджера, и кандидата.

Перевод технического собеседования в онлайн позволит кандидату выполнять задание в любое удобное ему время и пользоваться любыми удобными ему источниками, что максимально приблизит выполнение тестового задания к рабочей обстановке.

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. Особенно хотелось бы отметить отличительные особенности разрабатываемой системы, а именно возможности: загружать картинки в условие задания, создавать объектов базы данных с помощью файла со скриптами, напрямую подключаться к БД, переиспользовать созданные объекты и высылать задания кандидату по электронной почте.

Для кандидата столь широкий функционал не требуется. Для него требуется лишь удобный интерфейс для решения задания, возможности навигации по заданиям и чтения условия задания.

Для понимания бизнес-процессов, которые будут проходить в системе была разработана функциональная диаграмма IDEF0. На данном слайде представлена начальная диаграмма с точки зрения ресурс-менеджера. Для проведения тестирования знаний языка SQL требуется понимание, какие навыки нужны для вакансии. Кроме того, требуется информация о кандидате, уже прошедшем первое собеседование с отделом кадров, информация об объектах и заданиях в системе и практический опыт для составления качественных вопросов.

При первой детализации можно выделить несколько основных шагов тестирования: просмотр существующих заданий, создание нового задания, удовлетворяющего требованиям к вакансии, формирование комплекта заданий, отправка данного комплекта, проведение тестирование, которое происходит, по сути, без участия менеджера, и получение результатов после проверки задания системой

Просмотр существующих заданий заключается в формировании понимания о том, какие задания можно использовать заново после предыдущих собеседований, а какие требуется составить для оценки знаний кандидата.

Создание нового задания является самым сложным шагом в системе. Первым делом менеджеру необходимо выбрать схему и добавить в нее новые объекты или создать полностью новую схему. После этого требуется написать условие задания и эталонное решение. В результате, получаем почти готовое задание, котором требуется проставить настройки для завершения процесса, а именно: указать время на выполнение задания, указать метки для задания (например, сложность задания) и сохранить итоговый вариант задания.

Особенность создания объектов базы данных состоит в том, что менеджер может как переиспользовать старые объекты, так и создать новые путем загрузки sql файла в систему. Кроме того, имеется возможность точечного исправления объектов, так как менеджер получает логин с паролем для подключения к базе данных.

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 поиском строк, которые есть в эталонном решении, но нет в решении кандидата. Далее идет 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 отсутствуют связи много ко многим, для этого были введены специальные таблички, в которых хранятся данных связи (например, таблицы TasksOfPacks и TaskTags).

В системе существует несколько сложных алгоритмов, которые были разобраны в рамках расчетно-пояснительной записки. В рамках доклада рассмотрим алгоритм создания объектов БД. В начале, пользователь выбирает СУБД и схему в ней, либо вводит название новой схемы, если существующие не удовлетворяют его условиям. Если выбрана новая схема, то создадим новую схему и пользователей, под которыми будут выполняться скрипты. Один – для создания объектов, другой – для выполнения тестовых заданий.

Далее выдаем пользователю логин и пароль от пользователя для создания объектов. Если пользователю недостаточно объектов, которые уже присутствуют, или была создана новая схема, то даем пользователю возможность загрузки sql файла для создания объектов с последующей возможностью точечного изменения данных объектов с помощью прямого подключения к базе данных.

В результате, показываем пользователю полный список объектов и, если ему этих объектов достаточно, то переходим к созданию непосредственно условия задания.

Далее были спроектированы формы интерфейса. Для примера, 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 были выделены пакеты для соединения с БД, который используется для создания объектов базы данных и запуска решений, а также пакет rest - в него была выделена функциональность, обращение к которой происходит через REST API.

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.

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, что неавторизованный пользователь не имеет доступа 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 слайда6 система успешно прошла проверку безопасности.

Тестирование интерфейса пользователя заключалось в показе системы контрольный 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 отрабатывают. Это является не самой критичной ошибкой, так как система рассчитана именно на несложные задания, которые содержат именно 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 системой электронного обучения 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!