|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ***

***БАКАЛАВРА НА ТЕМУ:***

***Система контроля знаний языка запросов к базе данных***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-83Б |  |  | Ю.А. Шашкин |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Руководитель |  |  |  | М.М. Фомин |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Нормоконтролер |  |  |  |  |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
|  |  |  |  |  |

*2020 г.*

**РЕФЕРАТ**

SQL (Structured Query Language) - декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.

IDE (Integrated Development Environment) — система программных средств, используемая программистами для разработки программного обеспечения.

**СОДЕРЖАНИЕ**

[ОПРЕДЕЛЕНИЯ 8](#_Toc34003526)

[ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ 9](#_Toc34003527)

[ВВЕДЕНИЕ 10](#_Toc34003528)

[1 Анализ аналогов и выбор основных функциональных особенностей 11](#_Toc34003529)

[1.1 Выбор объекта анализа 11](#_Toc34003530)

[1.2 Анализ и тестирование функциональности для кандидата 13](#_Toc34003531)

[1.2.1 Блок задания 14](#_Toc34003532)

[1.2.2 Блок ввода решения 16](#_Toc34003533)

[1.2.3 Блок результатов 17](#_Toc34003534)

[1.2.4 Блок дополнительных функций 19](#_Toc34003535)

[1.3 Анализ и тестирование функциональности для менеджера 19](#_Toc34003536)

[1.3.1 Блок управления 20](#_Toc34003537)

[1.3.2 Блок информации о задании 21](#_Toc34003538)

[1.3.3 Блок задания 22](#_Toc34003539)

[1.3.4 Блок решения 24](#_Toc34003540)

[1.3.5 Блок тестов 26](#_Toc34003541)

[1.4 Выбор необходимых функций СКЗБД 29](#_Toc34003542)

[2 Проектирование и разработка системы 32](#_Toc34003543)

[2.1 Разработка функциональной спецификации 32](#_Toc34003544)

[2.2 Функциональная диаграмма IDEF0 32](#_Toc34003545)

[2.2.1 Функциональная модель с точки зрения ресурс-менеджера 32](#_Toc34003546)

[2.2.2 Функциональная модель с точки зрения кандидата 40](#_Toc34003547)

[2.3 Проектирование структуры базы данных 42](#_Toc34003548)

[2.3.1 Концептуальное проектирование 43](#_Toc34003549)

[2.3.2 Логическое проектирование 44](#_Toc34003550)

[2.3.3 Физическое проектирование 46](#_Toc34003551)

[2.4 Выбор подхода, языка и средств разработки 47](#_Toc34003552)

[2.5 Разработка алгоритмов работы системы 47](#_Toc34003553)

[2.5.1 Алгоритм создания объектов базы данных 47](#_Toc34003554)

[2.5.2 Алгоритм сохранения задания 48](#_Toc34003555)

[2.6 Разработка форм интерфейса 48](#_Toc34003556)

[2.6.1 Разработка форм интерфейса ресурс-менеджера 48](#_Toc34003557)

[2.6.2 Разработка форм интерфейса кандидата 48](#_Toc34003558)

[2.7 Разработка компонентов системы 48](#_Toc34003559)

[2.7.1 Разработка внутренних интерфейсов взаимодействия 48](#_Toc34003560)

[2.7.2 Работа с БД 48](#_Toc34003561)

[2.7.3 Разработка диаграммы классов системы 48](#_Toc34003562)

[2.7.4 Настройка почтового сервера 49](#_Toc34003563)

[2.7.5 Разработка функций IDE для форм ввода 49](#_Toc34003564)

[3 Тестирование и перспективы развития системы 50](#_Toc34003565)

[3.1 Автоматическое тестирование 50](#_Toc34003566)

[3.2 Тестирование безопасности 50](#_Toc34003567)

[3.3 Тестирование интерфейса пользователя 50](#_Toc34003568)

[3.4 Перспективы развития 50](#_Toc34003569)

[3.4.1 Развитие других видов тестирования 50](#_Toc34003570)

[3.4.2 Интеграция с Moodle 50](#_Toc34003571)

[ЗАКЛЮЧЕНИЕ 51](#_Toc34003572)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 52](#_Toc34003573)

# ОПРЕДЕЛЕНИЯ

SQL (Structured Query Language) - декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.

IDE (Integrated Development Environment) — система программных средств, используемая программистами для разработки программного обеспечения.

# ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

СКЗБД – система контроля языка запросов к базе данных

# ВВЕДЕНИЕ

В современных ИТ-компаниях проектные менеджеры достаточно большое количество времени тратят на собеседования, в том числе технические. Подготовка задания, его тестирование, проработка возможных ответов – это все занимает время менеджера, которое могло быть потрачено на развитие проектов компании.

Кроме того, техническая часть собеседования часто проводится на листочке, что дает только поверхностное понимание знаний кандидата, так как в рабочем процессе будет возможность использовать интернет ресурсы, собственные наработки – это все может значительно повысить производительность человека и некорректно лишать кандидата этих возможностей.

Данные проблемы могут быть решены использованием системы контроля знаний, которая позволит менеджерам создавать задания в онлайн форме, сохранять их, а затем, при необходимости, переиспользовать. Кандидату такая система позволит показать свои знания, используя все возможные источники информации, которые могут быть использованы в реальной работе.

# 1 Анализ аналогов и выбор основных функциональных особенностей

## 1.1 Выбор объекта анализа

Существует большое количество систем, которые можно рассматривать, как системы контроля знаний. Тестирование может проводиться в различных форматах: от простых тестов до написания сложных скриптов вручную.

Среди русскоязычных ресурсов можно выделить ресурсы «geekbrains» и «proghub». Данные ресурсы позволяют пройти тестирование на знания как языка запросов, так и теоретических основ реляционных баз данных.

Так как оба ресурса являются аналогичными по функциональности, рассмотрим только один из них. На рисунке 1 представлен пример тестового задания с ресурса «Geekbrains».

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 1 – Пример тестового задания с «GeekBrains»

Данный портал позволяет проходить заранее подготовленный тест, который не может быть изменен при необходимости. Задания делятся по темам и могут быть как тестовыми, так и с развернутым ответом. Задания с развернутым ответом являются самыми простейшими и могут быть решены в одну строчку кода. По причине того, что данный ресурс не дает возможности изменять тесты и решать сложные задания с развернутым ответом, то он не подходит для анализа.

Среди англоязычных ресурсов можно выделить популярный портал «codewars». Данный портал используется для практики программирования и предоставляет возможности как для обучения программированию на различных языках, так и для проведения собеседований. Версия для проведения собеседований является платной и может быть использована только в коммерческих целях. По информации из открытых источников – функциональность версии для собеседований аналогична функциональности для обучения программирования. Следовательно, вся необходимая функциональность может быть протестирована в бесплатной версии.

Обобщенные критерии сравнения обоих ресурсов с целью анализа функциональности для проведения собеседований представлены в таблице 1.

Таблица 1 – Сравнение «GeekBrains» и «CodeWars»

|  |  |  |
| --- | --- | --- |
| Критерий | GeekBrains | CodeWars |
| Задания в формате теста | Да | Нет |
| Простые задания с написанием запроса в 1-4 строчки | Да | Да |
| Сложные задания с написанием запроса в 5 и более строк | Да | Да |
| Задания с добавлением картинки | Нет | Нет |
| Возможность исправления задания | Нет | Да |
| Возможность добавления заданий | Нет | Да |
| Возможность отправки заданий по почте | Нет | Нет |
| Возможность ограничения времени решения конкретного задания/теста в целом | Да | Да |
| Возможность получения результатов решения (для менеджера) | Нет | Да |
| Возможность выбора СУБД | Нет | Да |
| Проверка синтаксиса в заданиях с развернутым ответом | Нет | Да |
| Возможность сбора статистики по тесту | Нет | Да |

По данным из таблицы 1 можно сделать вывод, что портал «CodeeWars» имеет большую часть необходимых для нас функций, поэтому для анализа будем использовать именно его.

## 1.2 Анализ и тестирование функциональности для кандидата

Типовое окно для тестируемого представлено на рисунке 2.

Изображение выглядит как снимок экрана, монитор

Автоматически созданное описание

Рисунок 2 – Типовое окно тестируемого на портале «codewars»

Анализ «codewars» для кандидатов будет производиться по следующим пунктам:

* блок задания (1);
* блок ввода решения (2);
* блок результатов, который вызывается при запуске решения (3);
* блок дополнительных функций (4).

### 1.2.1 Блок задания

Блок задания представлен на рисунке 3.

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 3 – Блок задания

Функционально данный блок можно разделить на 3 основные части: информация о задании, задание и теги задания. Рассмотрим каждую из них.

Информация о задании содержит:

* сколько раз задание добавлено в закладки,
* сколько раз задание добавлено в коллекции,
* пользовательский рейтинг,
* число успешных запусков,
* автор.

В ходе прочтения данной информации можно получить примерное представление о задании и о том, как его восприняло сообщество.

Самая большая часть в данном блоке отведена заданию. На примере, представленном на рисунке 3, задание представлено в текстовом виде и содержит информацию о рассматриваемом объекте и постановку вопроса. Задание может содержать особые выделенные структуры, которые хочет выделить автор (ключевые слова, примеры, куски кода). Пример с выделением таких структур представлен на рисунке 4.

Нижняя часть данного блока выделена для тегов задания. Теги могут быть использованы для сортировки задания (когда, например, пользователь хочет уделить особое внимание алгоритмам) и поиска.

Для проведения собеседований важными блоками будут автор и само задание. Информацию об удачных запусках, пользовательскому рейтингу задания, тегах, возможность добавлять в закладки и коллекции задание кандидату не нужно – это является избыточной для него информацией. Кроме того, для заданий на SQL было бы хорошо, если бы была возможность вставлять изображения в задания (например, схему БД).

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 4 – Пример задания с выделенными структурами

### 1.2.2 Блок ввода решения

Пример блока ввода решения представлен на рисунке 5

Изображение выглядит как снимок экрана, монитор

Автоматически созданное описание

Рисунок 5 – Блок ввода решения

Данный блок содержит информацию о выбранном языке программирования и его версии (в данном случае – SQL, PostgreSQL 9.6). Это дает пользователю возможность выбрать предпочтительную версию/расширения языка для работы. В случае SQL пользователь может выбрать с какой СУБД работать. Данный выбор представлен на рисунке 6.

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 6 – Выбор версии языка программирования

Далее в данном блоке присутствует возможность выбора темы интерфейса (темная или светлая), выбора типа редактора кода (обычный, vim, emacs) и включение полноэкранного режима.

Основную часть данного блока занимает окно ввода кода. Данное окно представлено в виде простейшей IDE, которая подсвечивает служебные слова, нумерует строки, но не имеет более сложны функций типа помощь при вводе и подсветки неправильного синтаксиса.

Для собеседований необходимыми являются функции выбора версии языка (для каких-то заданий изменение версии языка может быть запрещено) и окно ввода кода с простейшими функциями IDE для удобства пользователей.

### 1.2.3 Блок результатов

Тестирование производится с помощью автотестов, примеры которых представлены пользователю на странице. В блок результатов выводится информация о результатах данных автотестов. На рисунке 7 представлены результаты выполнения запроса.

Изображение выглядит как снимок экрана, текст

Автоматически созданное описание

Рисунок 7 – Результаты выполнения запроса

Из данного блока можно получить информацию о времени тестов, количество успешно пройденных, количество проваленных тестов, количество ошибок, результат. Далее представлены ошибки, которые возникали при выполнении скрипта. Данная информация позволяет пользователю сделать выводы о работе написанного скрипта и понять, какие изменения нужно вносить, если при выполнении возникли ошибки.

Для проведения собеседований кандидату имеет смысл показывать только информацию, если скрипт в принципе не выполняется (синтаксические ошибки). Если же скрипт выдает неверный результат и не проходит автотесты, то кандидату об этом знать не нужно. Такая практика является общепринятой и используется, к примеру, на официальных экзаменах Microsoft по MS SQL Server.

### 1.2.4 Блок дополнительных функций

Дополнительные функции представлены в виде панели под блоком ввода кода. Блок с данными функциями представлен на рисунке 8.

Изображение выглядит как снимок экрана, монитор, электроника, экран

Автоматически созданное описание

Рисунок 8 – Блок дополнительных функций

В данном блоке присутствует информация о простейших тестах, которыми можно проверить правильность свое решения и функциональные кнопки. С использованием функциональных кнопок можно:

* пропустить задание,
* показать решение,
* обсудить задание,
* перезапустить задание,
* провести простейшие тесты.

Для функционала собеседований необходимыми являются кнопки навигации по заданиям и запуска простейших тестов, то есть тестов синтаксиса и каких-либо других проверок на стороне сервера.

## 1.3 Анализ и тестирование функциональности для менеджера

Функциональность менеджера на данном портале представлена страницей создания задания. Данная страница представлена на рисунке 9. Выделим основные функциональные блоки на данной странице:

Изображение выглядит как монитор, снимок экрана, экран

Автоматически созданное описание

Рисунок 9 – Страница создания задания

* блок управления (1),
* блок информации о задании (2),
* блок задания (3),
* блок решения (4),
* блок тестов (5).

Анализ функциональности страницы создания задания будем проводиться по функциональным блокам, описанным выше.

### 1.3.1 Блок управления

Блок управления представлен на рисунке 10.



Рисунок 10 – Блок управления

Данный блок содержит 4 функциональных кнопки. Кнопка «Сохранить» - позволяет сохранить задание без его публикации. Кнопка «Сброс» очищает все заполненные поля и можно начать создание задания заново. Кнопка «Публикация» - осуществляет публикацию задания в открытый доступ. Кнопка «Удалить» - удаляет задание.

Для функционала собеседований необходимы аналогичные кнопки, так как данные 4 кнопки составляют необходимый функционал для управления заданием.

### 1.3.2 Блок информации о задании

Пример данного блока представлен на рисунке 11.

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 11 – Блок информации о задании

В данном блоке можно ввести название задание, которое будет отображаться пользователям. Далее представлена возможность выбора дисциплины (типа задания): основы, алгоритмы, исправление багов, рефакторинг кода, пазлы. Затем автору задания предоставляется возможность выбора минимального ранга, с которого доступно задание пользователям (ранг начисляется за верно решенные задания – так данная платформа ранжирует уровень знаний пользователей). Предпоследнее поле отвечает за теги, которые описывались в пункте 2.1. Завершает данный блок поле разрешения соавторства, которое позволяет разрешить или запретить другим пользователям предлагать изменения в задание.

С точки зрения менеджера ценность имеет только название задания и разрешение соавторства. Остальные же поля не имеют никакой ценности в условиях проведения технического собеседования.

### 1.3.3 Блок задания

В блоке задания имеются 3 вкладки: «описание», «предпоказ» и «помощь». Данные вкладки отвечают за редактирование задания, показ задания в виде, в котором оно будет демонстрироваться пользователю и инструкция к работе с этим полем соответственно.

Вкладка «описание» отвечает за полный контроль написания задания, то есть автор может писать в данной вкладке текс, как-либо форматировать его, чтобы получить задание в текстовом виде, которое будет представлено конечному пользователю. На рисунке 12 представлен простейший пример работы с данной вкладкой.

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 12 – Пример работы с вкладкой «описание»

На вкладке «предпоказ» (рисунок 13) можно получить информацию о том, как будет выглядеть задание, написанное в предыдущей вкладке для конечного пользователя. На данной вкладке представлен текст задания со всеми модификациями (например, выделение текста жирным).

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 13 – Вкладка «предпоказ»

Вкладка «помощь» (рисунок 14) дает автору возможность получить инструкцию по заполнению данного поля и получить полную и исчерпывающую информацию о возможностях, которые могут быть использованы для формирования текста задания.

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 14 – Вкладка «помощь» в блоке задания

Для проведения собеседований все вкладки важны и будут использоваться. Кроме того, необходимо предоставить функцию для добавления картинок в задание. Данная функция имеет ценность именно для заданий по написанию запросов к базе данных, так как с помощью картинку проще всего представить схему объектов базы данных.

### 1.3.4 Блок решения

Блок решения представляет из себя выбор языка и версии языка программирования и форма с 4 вкладками для создания решения задания. Данный блок представлен на рисунке 15. Выбор языка и версии языка аналогичен по функциональности выбору для пользователя, представленному на рисунке 6 и был описан ранее. Форма ввода решения состоит из вкладок: «полное решение», «начальное решение», «предзагруженное решение» и «помощь». Данная форма ввода решения так же, как блок решения у пользователя имеет функции простейшей IDE.

Изображение выглядит как снимок экрана, монитор, черный

Автоматически созданное описание

Рисунок 15 – Блок решения

Вкладка «полное решение» (рисунок 16) позволяет ввести эталонное решение задания, написанного прежде. Это решение будет показываться пользователям, если они решат пропустить задание, закончат его решение или нажмут на кнопку показа решения.

Изображение выглядит как монитор, снимок экрана, черный

Автоматически созданное описание

Рисунок 16 – Вкладка «полное решение»

На вкладке «начальное решение» (рисунок 17) автору необходимо написать то, что будет показываться пользователю в блоке написания решения. Это может быть кусок кода, который будет служить подсказкой для решения задания, или комментарий, упрощающий решение задания. При желании и необходимости можно оставить данный блок пустым, тогда у пользователя будет просто пустая первая строка в блоке с решением.

Изображение выглядит как электроника

Автоматически созданное описание

Рисунок 17 – Вкладка «начальное решение»

Вкладка «предзагруженное решение» позволяет автору увидеть, как пользователь будет видеть его решение (аналогично с вкладкой «предпоказ» в блоке задания).

Вкладка «помощь» содержит инструкции по работе с данным блоком, для чего он нужен и информацию по правильному написанию эталонных скриптов. Данная вкладка представлена на рисунке 18.

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 18 – Вкладка «помощь» в блоке решения

Для собеседований являются необходимыми вкладки «полное решение», «начальное решение» и «помощь». Также автору необходима возможность добавлять скрипты для создания объектов базы данных файловом виде или переиспользовать уже созданные объекты.

### 1.3.5 Блок тестов

Блок тестов используется для написания функционального тестирования скрипта. В данном блоке автор пишет тесты, которые будут запускаться при нажатии пользователем кнопки для запуска его скрипта. Данный блок состоит из кнопок запуска тестов, помощи, которые запускают тесты и вставляют примеры тестов соответственно, и 3 вкладок: «тесты», «примеры тестов» и «помощь». Данный блок представлен на рисунке 19.

Кнопка запуска тестов запускает все написанные автором тестом, которые указаны во вкладке «тесты». Кнопка помощи покажет примеры кода, которые могут быть использованы для написания тестов. Функция особенно необходима для авторов, которые пишут тесты впервые.

Изображение выглядит как снимок экрана, монитор

Автоматически созданное описание

Рисунок 19 – Блок тестов

Вкладка «тесты» (рисунок 20) содержит информацию о тестах, которые должны быть проведены для получения исчерпывающей информации о правильности работы пользовательского скрипта. Скрипт, написанный автором, должен проходить все тесты без ошибок для того, чтобы опубликовать задание. Считается хорошим тоном, если количество тестов приближается к нескольким сотням, содержат «граничные условия» и обработку неверных условий.

В вкладке «примеры тестов» (рисунок 21) находится информация о тестах, которые будут показаны пользователю в виде примера того, как будет проверяться его решение. Данная вкладка аналогична вкладке «тесты», но содержит лишь малую часть запускаемых тестов.

Вкладка «помощь» содержит инструкцию о работе с остальными вкладками в данном блоке, которая объяснить пользователю цель данного блока, что и как нужно писать в данном блоке и как это должно работать.

Изображение выглядит как снимок экрана, монитор, экран, внутренний

Автоматически созданное описание

Рисунок 20 – Вкладка «тесты»

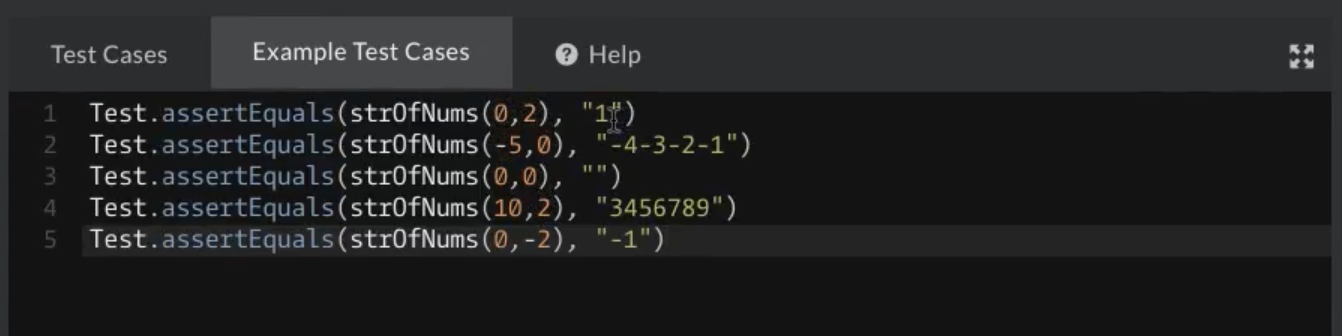


Рисунок 21 – Вкладка «пример тестов»

Для собеседований для контроля знаний языка написания запросов данный блок не является необходимым в связи с тем, что результатом работы SQL запроса обычно является таблица, поэтому проверить соответствие таблицы, полученной пользователем к таблице, полученной в результате выполнения запроса автора.

## 1.4 Выбор необходимых функций СКЗБД

Анализ систем контроля знаний языка запросов к базам данных, выполненный в данной главе, заключался в анализе систем, представленных на российском и зарубежном рынке. Наиболее подходящей системой оказалась система «codewars», которая в соревновательном формате позволяет изучать различные языки программирования, в том числе и интересующий нас SQL с различными версиями.

Анализ системы был проведен по всем функциональным блокам, присутствующим на портале как с точки зрения пользователя, так и с точки зрения автора заданий. В результате работы можно выделить основные пункты, необходимые для успешного использования системы в ходе технических собеседований. Данные функции, совместно со степенью их важности (от 1 – блокирующий да 5 - незначительный), для системы вынесены в таблицу 2.

Таблица 2 – Функции системы контроля знаний

|  |  |  |  |
| --- | --- | --- | --- |
| Функция | Приоритет | CodeWars | Проектируемая система |
| Для кандидата: | | | |
| Имя автора задания | 2 | Да | Да |
| Задание в текстовом формате | 1 | Да | Да |
| Нумерация строк в блоке решения | 3 | Да | Да |
| Выделение ключевых слов в блоке решения | 3 | Да | Да |
| Возможность менять темы интерфейса (темная/светлая) | 5 | Да | Да |
| Проверка синтаксиса написанного запроса | 2 | Да | Да |
| Навигация по заданиям | 1 | Да | Да |
| Ограничение по времени на выполнение задания | 1 | Да | Да |
| Для менеджера: | | | |
| Блок управления заданием (сохранить, удалить, опубликовать) | 1 | Да | Да |
| Разрешение соавторства | 2 | Да | Да |
| Название задания | 1 | Да | Да |
| Теги задания | 3 | Да | Да |
| Задание в текстовой форме | 1 | Да | Да |
| Задание в тестовой форме | 4 | Нет | Да |
| Возможность загружать картинки в условие задания | 1 | Нет | Да |
| Возможность выделения особых блоков в тесте задания (фрагментов кода) | 2 | Да | Да |
| Возможность предпоказа составленного задания | 3 | Да | Да |
| Инструкция по созданию задания | 1 | Да | Да |
| Блок для ввода эталонного решения | 1 | Да | Да |
| Возможность загружать скрипт создания объектов базы данных | 1 | Нет | Да |
| Возможность изменять созданные объекты из интерфейса системы | 4 | Нет | Да |
| Показ логина и пароля от пользователя БД для прямого подключения к ней | 1 | Нет | Да |
| Возможность переиспользовать объекты базы данных | 1 | Нет | Да |
| Возможность ввода начального решения для пользователя | 4 | Да | Да |
| Вкладка помощи при написании эталонного запроса | 1 | Да | Да |
| Возможность формирования работы из нескольких заданий | 2 | Нет | Да |
| Возможность отправки на почту кандидату заданий | 1 | Нет | Да |
| Сбор статистики по выполненным заданиям | 1 | Да | Да |
| Отчет по выполненному заданию кандидата | 1 | Нет | Да |

# 2 Проектирование и разработка системы

## 2.1 Разработка функциональной спецификации

Структурно-функциональная модель системы на основе методики IDEF0 позволит досконально разобраться в функциональных характеристиках разрабатываемой системы и рассмотреть все бизнес-процессы системы. Рассмотрим СКЗБД с нескольких точек зрения: с точки зрения ресурс-менеджера компании, который имеет потребность в оценке знаний кандидатов, и с точки зрения кандидата, который проходит тестирование в системе.

## 2.2 Функциональная диаграмма IDEF0

Структурно-функциональная модель системы на основе методики IDEF0 позволит досконально разобраться в функциональных характеристиках разрабатываемой системы и рассмотреть все бизнес-процессы системы. Рассмотрим СКЗБД с нескольких точек зрения: с точки зрения ресурс-менеджера компании, который имеет потребность в оценке знаний кандидатов, и с точки зрения кандидата, который проходит тестирование в системе.

### 2.2.1 Функциональная модель с точки зрения ресурс-менеджера

В соответствии с требованиями, предъявляемыми к разрабатываемой системе, можно выделать несколько входных данных:

* информация о кандидате – информация о навыках кандидата полученная из резюме или общения с сотрудником отдела кадров.
* Пул существующих вопросов – список вопросов, которые были созданы прежде и могут быть использованы для комплекта заданий.
* Пул существующих объектов БД – информация об объектах, которые существуют в базе данных и могут быть использованы для новых заданий.
* Практический опыт – опыт ресурс-менеджера, полученный в ходе повседневной работы.

В качестве управляющих данных можно выделить информацию о навыках, необходимых для вакансии, так как данная информация играет решающую роль в выборе кандидата.

Механизмами являются две основные системы:

* Тестирующая система – разрабатываемая система.
* СУБД – системы управления базами данных, которые будут использоваться для проверки правильности выполнения задания. Определяются в соответствии с техническим заданием.

Выходными данными являются результаты тестирования, которые будут использованы для принятия решения о соответствии кандидата на должность.

На основе представленных данных может быть разработана начальная контекстная диаграмма (рисунок 1).

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 1 – Начальная контекстная диаграмма А-0

При детализации диаграммы А-0 можно выделить несколько основных функциональных блоков:

1. Просмотр существующих заданий – получение списка уже существующих заданий и информации о них.
2. Создание нового задания – создание объектов, написание задания и эталонного решения данного задания.
3. Создание комплекта заданий – сборка нескольких заданий в комплект для отправки кандидату.
4. Отправка комплекта заданий кандидату
5. Проведение тестирования
6. Получение результатов – проверка полученного решения кандидата и сравнение с эталонном решением.

Диаграмма А0 представлена на рисунке 2.

Изображение выглядит как текст, карта

Автоматически созданное описание

Рисунок 2 – Диаграмма А0. Тестирование знаний языка SQL

Функциональный блок «Просмотр существующих заданий» может быть детализирован в следующие функциональные блоки:

1. Просмотр ранее созданных заданий
2. Просмотр заданий других менеджеров с открытым доступом
3. Принятие решения об использовании уже созданных заданий

Данная детализация представлена на рисунке 3.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3 – Диаграмма А1. Просмотр существующих заданий

Самым сложным функциональным блоком системы является блок о создании новых заданий. Данный блок состоит из следующих шагов:

1. Выбор схемы БД или создание новой – может быть выбрана схема с объектами, которая уже использовалась для создания заданий или может быть создана совершенно новая схема без объектов. В результате данного шага будут получены логин с паролем для прямого доступа к базе данных для точечной модификации объектов и данных.
2. Создание объектов – добавление объектов в базу данных для последующего составления заданий на их основе. Может быть сделано посредствам загрузки sql файла в интерфейсе тестирующей системы или с помощью прямого подключения к базе данных через программное обеспечение соответствующих СУБД. Данный функциональный блок будет дополнительно детализирован далее.
3. Написание условия задания – написание задания для кандидата в текстовом формате с возможностью добавления изображений (например, схемы СУБД).
4. Написание эталонного запроса – написание эталонного решения на написанное прежде задание. Данное решение будет использоваться для проверки правильности решения кандидата путем запуска решений и сверки результатов выполнения.
5. Задание времени на выполнение задания – установка максимального времени решения задания.
6. Проставление меток на задание – добавление меток о проверяемых знаниях кандидата и сложности к заданию.
7. Сохранение задания.

Диаграмма А2 представлена на рисунке 4.

Изображение выглядит как текст, карта

Автоматически созданное описание

Рисунок 4 – Диаграмма А2. Создание нового задания.

Как было упомянуто ранее, блок «Создание объектов БД» требует дополнительной детализации. Данный функциональный блок состоит из:

1. Получение логина и пароля доя подключения к БД.
2. Анализ существующих объектов.
3. Добавление объектов с помощью sql файла – загрузка и выполнение sql файла через разрабатываемую систему.
4. Точечное исправление объектов и данных в БД – исправление объектов БД с помощью прямого подключения к базе данных через программное обеспечение соответствующих СУБД.

Данная детализация представлена на рисунке 5.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5 – Диаграмма А22. Сохранение объектов в БД

Блок «Создание комплекта заданий» (рисунок 2) может быть детализирован до следующих блоков:

1. Выбор заданий для комплекта.
2. Расположение их по порядку.
3. Сохранение комплекта.

Описанная детализация представлена на рисунке 6.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6 – Диаграмма А3. Создание комплекта заданий

Последним блоком, нуждающимся в детализации, является блок «Получение результатов» (рисунок 2). Блоки детализации представлены далее:

1. Сверка решений кандидата с эталонными запросами.
2. Сбор статистики.
3. Показ результатов кандидатов.

Детализация представлена на рисунке 7.

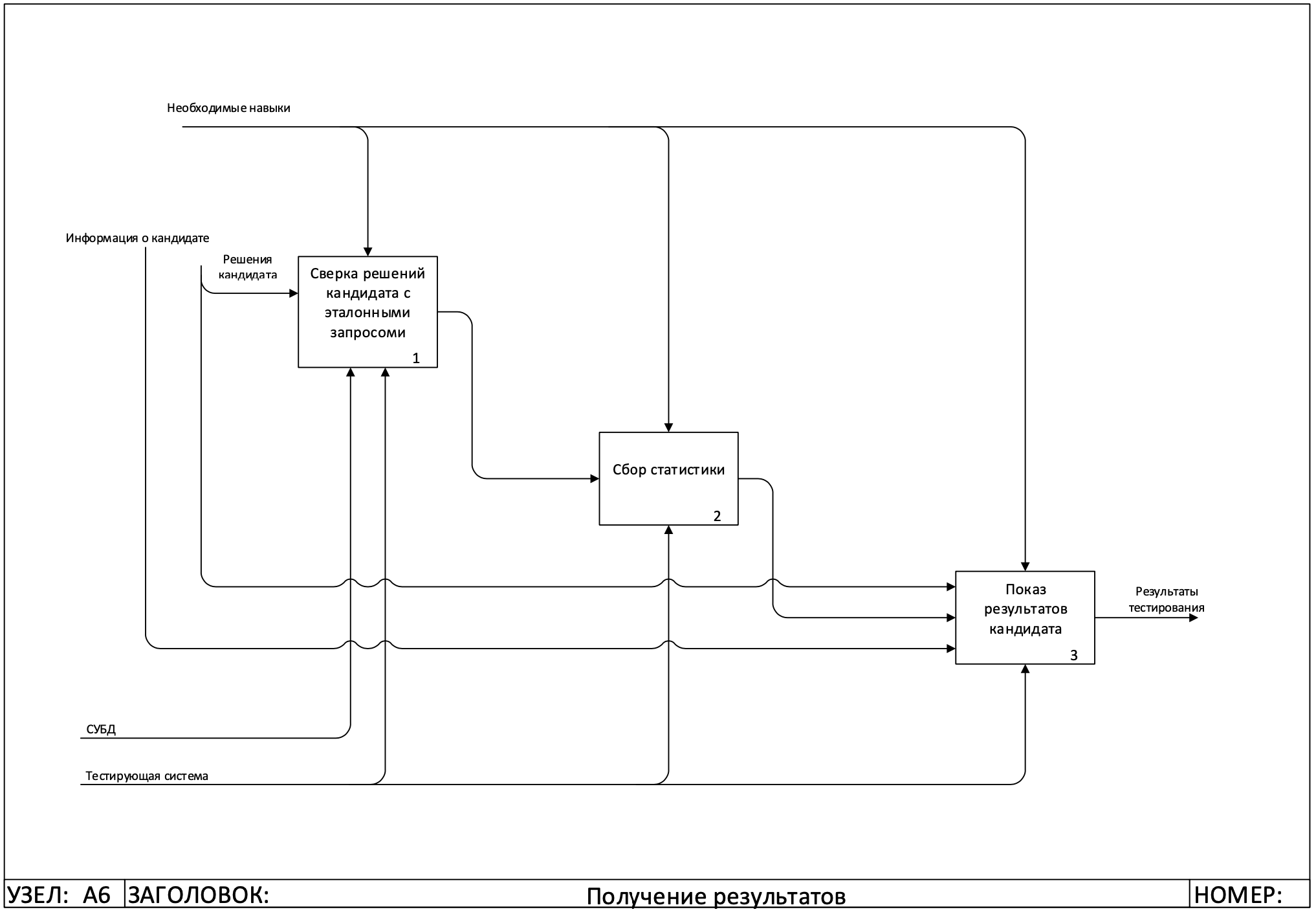


Рисунок 7 – Диаграмма А6. Получение результатов

### 2.2.2 Функциональная модель с точки зрения кандидата

С точки зрения кандидата разрабатываемая система намного проще, чем с точки зрения ресурс-менеджера, так как соискателю необходимо лишь пройти уже составленный тест, который был получен по электронной почте.

Кандидат имеет на вход письмо от работодателя с ссылкой на индивидуальный тест в тестирующей системе. Механизмом проведения тестирования является непосредственно разрабатываемая система. В качестве управляющих данных кандидат имеет собственные знания в рассматриваемой предметной области. Выходными данными считаются результаты тестирования, которые будут получены в устной или письменной форме от ресурс-менеджера на собеседовании или по электронной почте соответственно. Полученная начальная контекстная диаграмма представлена на рисунке 8.

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 8 – Начальная контекстная диаграмма А-0 с точки зрения кандидата

Представленная начальная контекстная диаграмма может быть детализирована до следующих функциональных блоков:

1. Переход по ссылке из письма.
2. Прочтение инструкций к тестированию.
3. Прохождение тестирования.
4. Сохранение ответов.
5. Получение результатов после рассмотрения ответов менеджером.

Описанная детализация представлена на рисунке 9. Ни один их описанных выше функциональных блоков не может быть более детализирован.

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 9 – Диаграмма А0. Тестирование знаний языка SQL с точки зрения кандидата

## 2.3 Проектирование структуры базы данных

СКЗБД должна содержать объекты, отвечающие различным бизнес-потребностям и использующиеся для различных целей: часть объектов необходимы системе для ее функционирования (системные объекты) и объекты для тестирования (тестовые объекты). Из соображений безопасности имеет смысл разместить различные типы объектов в различных базах данных, в ином случае существует вероятность использования sql инъекций, которые могут принести непоправимый вред системе. Кроме того, тестовые объекты могут находиться в различных СУБД, однако системные объекты должны быть доступны разрабатываемой системе постоянно и находиться в одном месте. Размещение тестовых и системных объектов в различных базах способно решить и эту проблему.

Алгоритм работы с тестовыми объектами будет описан в последующих главах, в данной главе рассмотрим системную базу данных.

### 2.3.1 Концептуальное проектирование

Концептуальное проектирование состоит в построении информационной модели самого высокого уровня. Так как бизнес-процессы разрабатываемой системы были описаны с помощью нотации IDEF0, то могут быть получены основные сущности БД:

* Пользователи (Users) – в таблице будут храниться пользователи (ресурс-менеджеры) с логинами и паролями, информация о времени последней авторизации.
* Департамент (OrgUnit) – в таблице будут храниться департаменты компании, в которых может находиться пользователь.
* Схемы (Schemes) – названия схем тестовых объектов с указанием СУБД, создателя схемы и описанием, заполненным создателем схемы.
* Задания (Tasks) – текст задания, дополнительная информация о нем, эталонное решение, информация об используемой схеме.
* Метки (Tags) – таблица с метками которые могут быть использованы в заданиях.
* Метки задания (TaskTags) – связь заданий с метками.
* Комплекты заданий (Packs) – описание комплекта заданий, информация о времени создания комплекта и его создателе, количестве заданий.
* Таблица для связи комплектов и заданий (TasksOfPacks) – таблица будет использоваться для хранения информации о заданиях, которые находятся в комплекте заданий, их порядковом номере в комплекте.
* Отправки комплектов заданий (PacksSendings) – информация об отправках комплектов заданий на электронные почты кандидатов
* История изменений (ChangeHistory) – таблица для логирования любой информации об изменениях.
* Результаты решений (TestResults) – результаты тестирования кандидатов: запрос кандидата и результаты его проверки.

Все описанные сущности вместе со связями могут быть представлены в виде инфологической схемы, представленной на рисунке 10.

Изображение выглядит как карта

Автоматически созданное описание

Рисунок 10 – Инфологическая модель системной базы данных

### 2.3.2 Логическое проектирование

Этап логического проектирования для данной базы данных состоит в подробном описании полей таблиц, ограничителей целостности. Даталогическая модель базы данных представлена на рисунке 11.

Большая часть полей соответствует техническому заданию и не нуждается в дополнительных комментариях. Далее будут описаны лишь поля, цель использования которых невозможно понять из ТЗ и бизнес-процессов системы:

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 11 – Инфологическая схема системной базы данных

* CreationDate (DATETIME) – дата создания задания. Будет использоваться для указания даты и времени создания задания в системе, для понимания актуальности задания.
* CreatedBy (User) – пользователь, создавший задание. Может быть использовано для личного контакта при необходимости.
* ChangeHistoryInfo (JSON) – любые изменения в системе, которые должны логироваться. Хранится в формате JSON в связи с тем, что изменения могут быть какие угодно и их структура может быть независимой друг от друга.
* SchemeName (NVARCHAR(50)) – имя схемы базы данных для тестовых объектов для прямого обращения к данной схеме.
* SchemeAlias (NVARCHAR(50)) – пользовательское имя схемы базы данных, указывается при создании.
* PackTime (INT) – общее время на выполнение комплекта заданий. Определяется как сумма времени на выполнение всех заданий.

Основными ограничителями целостности являются первичные и вторичные ключи, которые представлены на даталогической схеме (рисунок 11). В качестве первичных ключей в системе используются авто инкрементируемые суррогатные ключи, так как они позволяют со 100% вероятностью обеспечить целостность данных. Поля, которые являются бизнес ключами будут иметь ограничитель целостности UNIQUE, что позволит контролировать их уникальность без риска нарушения целостности данных.

Работа с базой данных из разрабатываемой системы будет осуществляться в пределах транзакций, что позволит исключить вероятность лишних блокировок и позволит поддерживать данные всегда в актуальном состоянии.

### 2.3.3 Физическое проектирование

Основная задача физического этапа проектирования в разрабатываемой базе данных – выбор СУБД.

На основе технического задания система должна работать с системами Oracle 11g или выше и MS SQL Server 2012 ли выше. Сравним основные преимущества обеих СУБД.

Преимущества MS SQL Server:

* Простота использования
* Утилиты (SQL Server Profiler, SQL Server Management Studio, BI tools и Database Tuning Advisor)
* Онлайн поддержка
* Простота и скорость восстановления (в сравнении с другими СУБД)
* Цена

Преимущества Oracle:

* Возможность обновлений без перезагрузок
* Кроссплатформенность
* Скорость работы с большими объемами данных (в сравнении с другими СУБД)
* Средства виртуализации

В целом, данные СУБД похожи друг на друга и имеют лишь небольшие различия, не имеющие значения для не высоконагруженных систем. В связи с этим выбор следует делать из соображений квалификации проектной команды. По данному критерию выбор будет сделан в пользу MS SQL Server.

Так как система не является высоконагруженной, то использование дополнительных индексов и каких-либо группировок данных не требуется. Снятие резервных будет производиться средствами системы управления базами данных и операционной системы в автоматическом режиме.

## 2.4 Выбор подхода, языка и средств разработки

За многолетний опыт разработки программного обеспечения людьми были разработано достаточно большое количество различных подходов к разработке, каждый из которых имеет свои преимущества. В настоящее время наибольшую популярность приобрели гибкие методологии. Данные методологии позволяют создавать продукт более эффективно, нежели традиционные. Это происходит за счет того, что наибольшую ценность составляет бизнес ценность для заказчика программного продукта, а не сам программный продукт. По данной причине для разработки данной системы будет выбран именно Agile модель и методология Scrum, суть которой заключается в работе специальными временными промежутками (1-2 недели) – спринтами, в результате которых производится показ результатов работы заказчику.

Выбор языка разработки серверной части программного обеспечения осуществляется по нескольким критериям:

* наличие библиотек для интеграции с различными СУБД,
* обширное сообщество,
* возможности масштабирования,
* возможности многопоточности.

Выбор осуществлялся среди 3 самых популярных языков для веб разработки: Java, .NET, Python. Данные языки имеют примерно одинаковые характеристики по выделенным критериям, в связи с чем было принято решение использовать Java, так как разработчик знает этот язык лучше, чем другие из представленных.

Язык для фронтенд разработки будет выбран по популярности языка среди сообщества, так как разработчик системы не имеет обширных знаний ни в одном из языков фронтенд разработки, а самый популятный язык имеет наибольшее количество обучающих видео и форумов. По данному критерию был выбран язык JavaScrict, библиотека React.js.

## 2.5 Разработка алгоритмов работы системы

### 2.5.1 Алгоритм создания объектов базы данных

### 2.5.2 Алгоритм сохранения задания

## 2.6 Разработка форм интерфейса

### 2.6.1 Разработка форм интерфейса ресурс-менеджера

### 2.6.2 Разработка форм интерфейса кандидата

## 2.7 Разработка компонентов системы

### 2.7.1 Разработка внутренних интерфейсов взаимодействия

### 2.7.2 Работа с БД

### 2.7.3 Разработка диаграммы классов системы

### 2.7.4 Настройка почтового сервера

### 2.7.5 Разработка функций IDE для форм ввода

# Тестирование и перспективы развития системы

## 3.1 Автоматическое тестирование

## 3.2 Тестирование безопасности

## 3.3 Тестирование интерфейса пользователя

## 3.4 Перспективы развития

### 3.4.1 Развитие других видов тестирования

### 3.4.2 Интеграция с Moodle

# ЗАКЛЮЧЕНИЕ

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Основы SQL [Электронный ресурс] https://proghub.ru/t/sql-basic (дата обращения 23.11.2019).
2. GeekBrains – образовательный портал [Электронный ресурс] https://geekbrains.ru/ (дата обращения 23.11.2019).
3. Codewars [Электронный ресурс] https://www.codewars.com/ (дата обращения 23.11.2019).
4. Codewars.com wiki [Электронный ресурс] https://www.codewars.com/ (дата обращения 23.11.2019).
5. Teach coding with games: a review of Codewars and CodeCombat [Электронный ресурс] https://opensource.com/education/15/7/codewars-codecombat-review (дата обращения 23.11.2019).
6. Why Codewars is the best way to learn a new programming language! [Электронный ресурс] https://opensource.com/education/15/7/codewars-codecombat-review (дата обращения 23.11.2019).
7. Code Wars: Creating a Problem / Kata [Электронный ресурс] https://www.youtube.com/watch?v=Ytl4WPJ\_zgA (дата обращения 23.11.2019).
8. How to Create a Kata in Codewars [Электронный ресурс] https://www.youtube.com/watch?v=TZTnP3bZZSU (дата обращения 23.11.2019).