МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Кафедра информационных технологий**

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ № 3**

**по дисциплине  
 «ТЕХНОЛОГИЯ ПРОЕКТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»**

Выполнила студентка группы 35/2                                       В. С. Паничева

Отчет принял

доц. каф. ИТ                                                                                        А. Н. Полетайкин

Краснодар

2024 г.

Тема: Техническое задание на создание программного продукта.

Цель: Освоение методики предварительного анализа разрабатываемой программы; освоение задач формулирования функциональных и нефункциональных требований к программной реализации отдельных задач и к программе в целом; выработка навыков разработки технического задания.

Задание

1. Установить назначение и общую цель создания программы.
2. Определить структуру программы и состав функциональных задач.
3. Разработать функциональные требования к программе.
4. Разработать модель требований в нотации UML.
5. Разработать требования к информационному обеспечению (к базе данных).
6. Разработать требования к инструментальному программному обеспечению (к системе управления базой данных (СУБД), к средству разработки программ (IDE), средствам автоматизированного проектирования ПО).
7. Установить нефункциональные требования к программе. Дать не менее 5 наиболее очевидных для данной системы требований.

Тема проекта: Цифровой помощник учителя математики

**1 Назначение подсистемы и цели создания**

Назначением программы является автоматизация процесса «Проведение урока математики» для центра дополнительного образования «Школа Пифагорум».

Целью создания программы является оптимизация процесса обучения математики, в последствии которой произойдет: улучшение эффективности работы учителя, повышение интереса учеников к данному предмету и обеспечение более наглядного способа изучения уравнений через графическое отображение.

**2 Функциональные задачи**

Определены следующие функциональные задачи:

1. Задача выбора типа уравнения для решения.

2. Задача выбора существующего математического примера из базы данных по определенной теме.

3. Задача генерации нового уравнения по соответствующей теме.

4. Задача ввода уравнения, придуманного пользователем.

5. Задача отображения графика уравнения для визуального представления.

6. Задача предоставления правильного ответа на данное уравнение для проверки решения.

**3 Функциональные требования к системе**

Программа для помощи преподавателям математики должна иметь большой набор возможностей. Поэтому к ней предъявляется множество требований.

Функциональные требования:

1. Возможность выбора типа задачи из предоставленных сервисом. Пользователь может выбрать тип задачи для решения. Это могут быть линейные уравнения, квадратные уравнения, системы уравнений.
2. Возможность ввода математических функций. После выбора пользователем типа задачи, он должен иметь возможность ввести математическую функцию в формате "3\*x - 2". Пользователь должен также смочь вводить параметры для других математических функций, таких как радиус и центр окружности, координаты вершин геометрических фигур и т. д. В зависимости от задачи ему будут доступны соответствующие поля для ввода информации. После ввода всех необходимых данных пользователь должен иметь возможность нажать кнопку "Построить график", чтобы увидеть результаты на экране в виде графика, соответствующего введенной функции.
3. Генерация математического примера. После нажатия кнопки «Сгенерировать новую задачу» пользователем, программа должна сгенерировать новый пример по выбранной теме. Система генерации математических примеров должна использовать фасетную технологию. Для генерации разнообразных вариантов однотипных задач должны использоваться заготовленные шаблоны, в которых будут меняться только определенные параметры.
4. Отображение ответа. Сервис должен предоставлять правильный ответ на выбранную задачу по кнопке «Показать ответ».
5. Построение графика. После ввода пользователем функции, выбора примера или генерации новой задачи, приложение должно предоставить пользователю возможность построить график соответствующего математического выражения. Для визуализации графика должен быть использован элемент HTML5 Canvas, который предоставляет широкие возможности для рисования и отображения графических объектов веб-приложения. Программа должна рассчитывать точки графика, в зависимости от значений переменных и функции. После этого, с помощью функций Canvas, должен отображаться график на экране.
6. Поддержка различных типов графиков. Программа должна визуализировать графики для различных типов кривых:
   * Функции , такие как линейные, тригонометрические (синус, косинус, тангенс), экспоненциальные (экспонента, логарифм), степенные (квадрат, куб), гиперболические функции (гиперболический синус, гиперболический косинус).
   * Построение кривых, заданных параметрически в декартовой системе координат:
   * Окружность: по центру и точке на ней; по центру и радиусу.
   * Эллипс - по двум фокусам и точке на кривой.

**4 Диаграмма вариантов использования**

Для наглядности взаимодействия пользователя с системой на рисунке 3 представлена диаграмма вариантов использования.

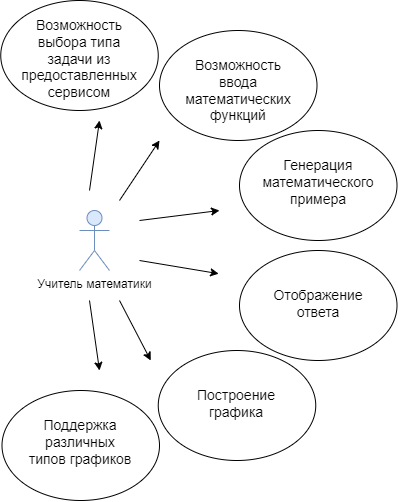


Рисунок 2 - Диаграмма вариантов использования

**5 Требования к СУБД и IDE**

Требования к СУБД:

1. Поддержка структурирования и хранения базы данных с задачами по математике.

2. Обеспечение быстрого доступа к данным и операциям с ними.

3. Совместимость с языком программирования, используемым для разработки приложения.

4. СУБД должна поддерживать принципы CRUD (Create, Read, Update, Delete) для каждой таблицы.

Требования к IDE:

1. Поддержка разработки веб-приложений для удобного использования программы через интернет.

2. Интеграция с выбранной СУБД для удобной работы с базой данных.

3. Поддержка языка программирования, который будет использоваться для создания программы.

4. Возможность отладки и тестирования приложения внутри среды разработки.

**6 Нефункциональные требования к системе**

Нефункциональные требования к системе включают:

1. Интуитивный интерфейс. Пользовательский интерфейс должен быть понятным и легко использоваться, даже для пользователей без специальных знаний в области математики или программирования.
2. Точность. Автоматические ответы, предоставляемые на задачи, должны максимально соответствовать результатам аналитического решения. Графики должны строиться с высокой степенью точности.
3. Логичность. Генерация новых математических примеров должна быть систематичной и логичной, обеспечивая пользователю убедительные упражнения для решения.
4. Возможность настройки графиков. Пользователь должен иметь возможность добавлять несколько функций на одном графике, цвет каждого графика должен отличаться, чтобы можно было отличить их друг от друга, даже если они пересекаются или расположены близко друг к другу.
5. Малая стоимость. Данное ПО должно быть дешёвым в разработке, чтобы оно могло конкурировать на рынке, но это не должно сказаться на качестве разрабатываемого ПО.

В целом, приложение должно обладать простым интерфейсом, быть удобным в использовании, а также быть точным и логичным.

**Вывод:** В работе были выделены основные функциональные и нефункциональные требования к цифровому помощнику учителя математики, представлена модель требований, созданная в нотации UML, разработана структура программы и состав задач. Также были определены требования к информационному и инструментальному программному обеспечению, учитывая потребности пользователей и цели программы.