МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «КубГУ»)

Факультет компьютерных технологий и прикладной математики  
Кафедра информационных технологий

**ОТЧЁТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЕ №8 ПО ДИСЦИПЛИНЕ «ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»**

Работу выполнила студентка 4ИТ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. А. Арабова

Проверил доцент кафедры ИТ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Н.Полетайкин

Краснодар

2023

Тема: Тестирование ПС

Цель: Освоение методики сценарного и интеграционного тестирования ПО, в том числе разработанной программной системы в ручном режиме.

Задание

Выполнить системное пользовательское тестирование работоспособности ПС, разработанную при выполнении лабораторной работы №7, посредством воздействия на её интерфейсную часть. Для этого необходимо:

1. Согласно функциональным требованиям к ПС, предъявленным при выполнении лабораторной работы №3, разработать несколько сценариев тестирования ПС.

2. Реализовать сценарии и задокументировать полученные результаты тестирования.

3. По согласованию с преподавателем разработать сценарий тестирования стороннего программного продукта согласно данным, представленным в приложении Е. Оформить отчет по выполненному сценарному тестированию.

4. Выполнить интеграционное тестирование на компонентном уровне разработанных при выполнении лабораторной работы №7 автоматизированных функций программного продукта, используя подход Bottom Up Integration.

5. Выполнить нагрузочное тестирование программы и оценить эффективность разработанных при выполнении лабораторной работы №6 автоматизированных функций ПС, запросов к БД. В качестве критерия эффективности использовать время выполнения функции. Предварительно подготовить 5 массивов исходных данных на 10, 50, 100, 500 и 1000 записей.

Построить графики зависимости времени вычислений от объема исходных данных.

Индивидуальная тема: Сервис проверки корректности технической документации.

1. Системное (ручное) пользовательское тестирование ПС.

Системное тестирование – это тестирование всей системы в целом, как правило, через ее пользовательский интерфейс. Следует проверить соответствует ли разработанная ПС заявленным требованиям.

Система адаптивна под различные разрешения экрана, интерфейс простой и незапутанный.

Для корректного тестирования необходимо провести 2 теста: со стороны пользователя и со стороны администратора.

1.1 Тест №1 (Инженер)

Состоит из проверки функционала пользователя состоит из следующих действий:

1) Запуск программы

2) Выбор файлов с неправильным расширением

3) Проверить валидацию данных

4) Выбор файлов с верным расширением

5) Выбор несоответствующего шаблона

6) Загрузить файл с экспериментальной ВАХ

7) Проверить, что программа корректно загружает файл и шаблон

8) Нажать на кнопку «Обработать» и проверить корректность работы

Ожидается, что все элементы интерфейса корректно работают. Этот тест успешно пройден.

Скриншоты срабатывания некоторых пунктов проверки:

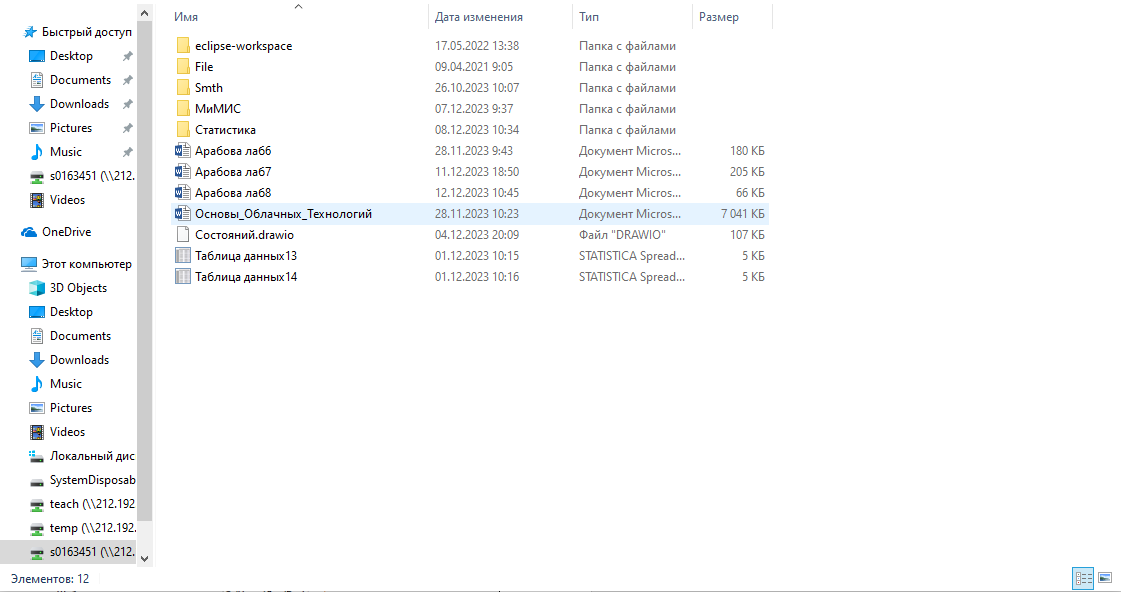


Рисунок – Окно выбора файла до нажатия «Выбрать файл»

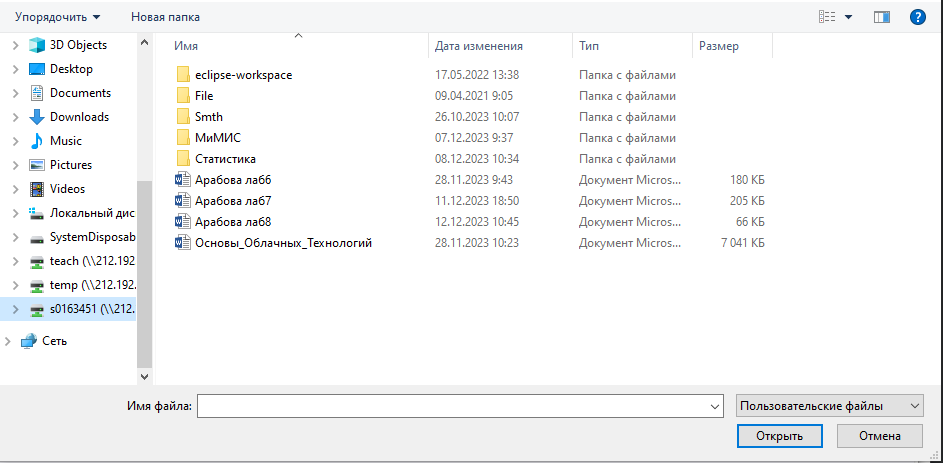


Рисунок – Окно выбора файла, отсеяны все не doc и docx

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рисунок 2 – Интерфейс экрана вывода ошибок

2. Нагрузочное тестирование

Выполним нагрузочное тестирование программы и оценим эффективность автоматизированных функций ПС, запросов к БД. В качестве критерия эффективности используем время выполнения функции. Для тестирования подготовим 5 массивов исходных данных на 10, 50, 100, 500 и 1000. Тестирование проводилось на таблице клиентов. Результаты тестирования представлены в таблице 1.

Технические характеристики используемой машины:

• Оперативная память: 16 Гб

• Процессор: 4 ядра с тактовой частотой 2.333 ГГц, 8 потоков

• Разрядность Windows: 64-разрядная операционная система

• Размер жёсткого диска: 512 Гб

|  |  |  |  |
| --- | --- | --- | --- |
| № | Кол-во записей | Операция | Результат (сек.) |
| 1 | 10 | Вставка | 0,2445 |
| 2 | 50 | Вставка | 0,2636 |
| 3 | 100 | Вставка | 0,3134 |
| 4 | 500 | Вставка | 0,4321 |
| 5 | 1000 | Вставка | 0,7611 |

Таблица 1 – Сравнительная таблица нагрузочного тестирования

При данном тестировании в базу данных загружается сначала 10 записей об объектах, далее 50, 100, 500 и 1000. Далее строится проект, тем самым проверяется скорость работы ПС с различными объёмами данных

После проведения теста мы получили следующие данные:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Количество исходных данных | 10 | 50 | 100 | 500 | 1000 |
| Время вычислений | 1870 мс | 1980 мс | 5093 мс | 5516 мс | 9452 мс |

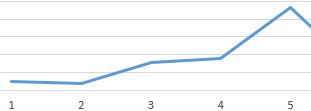


Рисунок 6 – График работы программы

3. Стрессовое тестирование

В этом виде тестирования необходимо проверить работу системы при чрезмерной нагрузке на неё. Стрессовое тестирование ПС будет проведено посредством отправки многократных запросов к БД PostgreSQL. Во время тестирование было обнаружено, что даже в стрессовых ситуациях приложение продолжает функционировать, БД быстро обрабатывает запросы. Могут возникать небольшие задержки в работе интерфейса, но после выполнения череды запросов задержки перестают себя проявлять.

4. Модульное тестирование

Для тестирования работы вставки был написан следующий тест:

def select\_file():  
 file1\_path = filedialog.askopenfilename(initialdir="/", title="Select file",  
 filetypes=(("text files", "\*.docx"), ("all files", "\*.\*")))  
 file1\_entry.delete(0, tk.END)  
 file1\_entry.insert(0, file1\_path)  
  
 file2\_path = filedialog.askopenfilename(initialdir="/", title="Select file",  
 filetypes=(("text files", "\*.docx"), ("all files", "\*.\*")))  
  
 file2\_entry.delete(0, tk.END)  
 file2\_entry.insert(0, file2\_path)

