Петрозаводский государственный университет
Институт математики и информационных технологий
Кафедра прикладной математики и кибернетики

Отчет по дисциплине «Верификация программного обеспечения»

Выполнила:
студентка 3 курса группы 22407

А. Р. Хуснутдинова _____
подпись

Петрозаводск — 2023

Содержание

1 Объект тестирования

1.1 Описание объекта тестирования

Объектом тестирования является телеграм-бот для генерации заявлений. Бот генерирует три вида заявлений: заявление на отдых в Спортивно-оздоровительном лагере Шотозеро, заявление на перевод в поликлинику №2 и заявление на каникулы после сдачи ГИА. Все заявление составляются по шаблонам, заявленным ПетрГУ и предназначены для использования студентами ПетрГУ.

Телеграм-бот выполняет следующие функции:

1. Получение запроса пользователей на создание нового заявления;
2. Запрос и получение необходимых недостающих данных для заполнения запрашиваемого заявления;
3. Удаление всех данных о пользователе по запросу пользователя;
4. Генерация запрашиваемого заявления;
5. Отправка пользователю запрашиваемого заявления.

2 Стратегия тестирования

2.1 Высокоуровневая архитектура

Чат-бот разбит на два модуля (Рис. 1).

Модуль бота bot отвечает за взаимодействия с пользователем при помощи API телеграма. Модуль запрашивает у пользователя данные, принимает и обрабатывает их, а затем отправляет в базу данных. Модуль бота осуществляет проверку вводимых данных, нахождение их в базе данных, вызывает функции из модуля генерации.

Модуль генерации generate осуществляет генерацию запрашиваемых документов, используя методы функции pylatex. Данные для генерации достаются из базы данных.

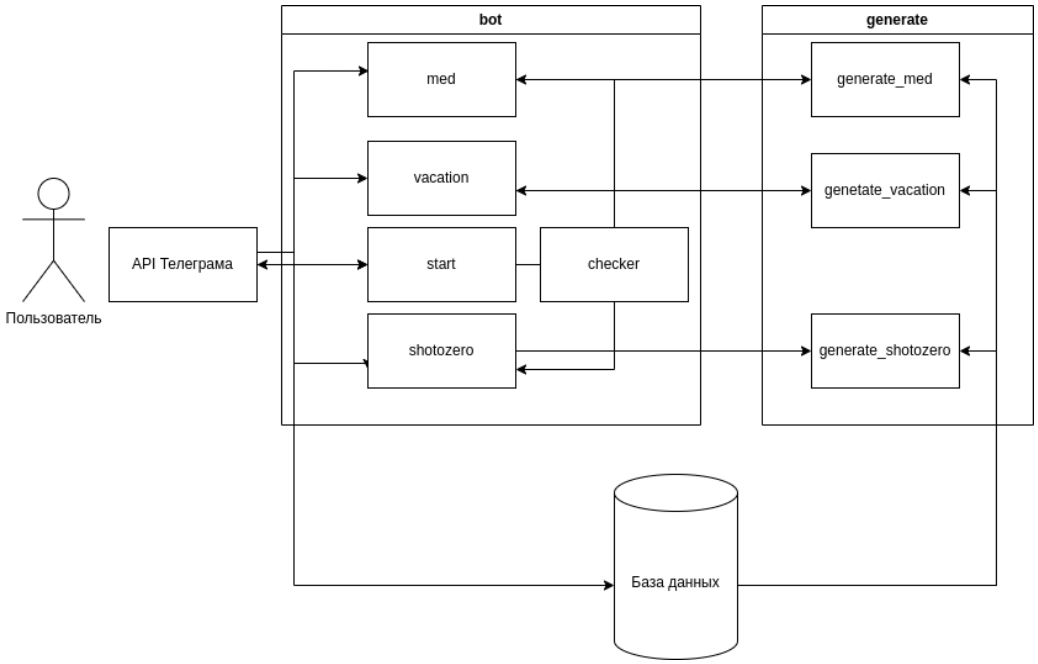

Рис. 1 – Клиент-серверная модель

2.1.1 Модуль bot

Модуль bot отвечает за ввод и вывод данных, сохранение их в баз данных, взаимодействие с пользователем.

- Модуль start:
  - Функция start(message)

    Функция получает на вход команду от пользователя и выводит приветственное сообщение в чат.

- Функция clear(message)

  Функция получает на вход команду от пользователя и очищает данные о пользователе в базе данных

- Функция bot_message(message)

  Функция получает на вход название генерируемого документа и запускает соответствующий модуль получения данных

- Модуль получения данных med:

  - Функция get_med_data(chat, username)

    Функция получает на вход идентификатор чата и имя пользователя, выводит сообщение о сборе данных

  - Функция get_info(chat, username)

    Функция получает на вход идентификатор чата и имя пользователя, достаёт значения из базы данных и поочерёдно вызывает функции для недостующих полей, отправляет сообщения о запросах пользователю, вызывает функцию генерации из модуля generate после завершения ввода.

  - Однотипные функции get_medOrg, get_firstname(message), get_secondname(message) get_surname(message), get_datebirth(message), get_gender(message), get_phone(messa get_polic(message), get_policOrg(message), get_policeDate(message), get_state(messag get_passportSeria(message), get_passportNumber(message), get_passportPlace(message get_passportDate(message)

    Функции получают на ввод сообщения от пользователя и добавляют сответствующие значения в базу данных. Для некоторых функций вызывваются функции из модуля checker, если значение не проходит проверку, об этом сообщается пользователю

- Модуль получения данных shotozero:

  - Функция get_shotozero_data(chat, username) Функция получает на вход идентификатор чата и имя пользователя, выводит сообщение о сборе данных

  - Функция get_info(chat, username)

    Функция получает на вход идентификатор чата и имя пользователя, достаёт значения из базы данных и поочерёдно вызывает функции для недостующих полей, отправляет сообщения о запросах пользователю, вызывает функцию генерации из модуля generate после завершения ввода.

- Однотипные функции shotozero_firstname(message) shotozero_secondname(message) shotozero_surname(message) shotozero_phone(message) shotozero_adress(message) shotozero_course(message) shotozero_groupnumber(message) shotozero_NEWstart(mess shotozero_year(message) shotozero_date(message)
  Функции получают на ввод сообщения от пользователя и добавляют сответствующие значения в базу данных. Для некоторых функций вызывваются функции из модуля checker, если значение не проходит проверку, об этом сообщается пользователю

- Модуль получения данных vacation:
  - Функция get_vac_data(chat, username) Функция получает на вход идентификатор чата и имя пользователя, выводит сообщение о сборе данных
  - Функция get_info(chat, username)
    Функция получает на вход идентификатор чата и имя пользователя, достаёт значения из базы данных и поочерёдно вызывает функции для недостующих полей, отправляет сообщения о запросах пользователю, вызывает функцию генерации из модуля generate после завершения ввода.
  - Однотипные функции get_firstname(message) get_secondname(message) get_surname(message) get_course(message) get_groupnumber(message) get_housenumber(m get_speccode(message) get_specname(message) get_date(message) get_year(message)
    Функции получают на ввод сообщения от пользователя и добавляют сответствующие значения в базу данных. Для некоторых функций вызывваются функции из модуля checker, если значение не проходит проверку, об этом сообщается пользователю

- Модуль проверки значений checker:
  - Функция check_firstname(firstname)
    Получает на вход строку, возвращает результат проверки на первую заглавную букву. Возвращает булевое значение.
  - Функция check_secondname(secondname)
    Получает на вход строку, возвращает результат проверки на первую заглавную букву. Возвращает булевое значение.
  - Функция check_surname(surname)

Получает на вход строку, возвращает результат проверки на первую заглавную букву. Возвращает булевое значение.

- Функция check_passportSeria(seria)
  Функция получает на вход строку, проверяет, являются ли значения цифрами и соответствие необходимой длине. Возвращает булевое значение.
- Функция check_passportNumber(number)
  Функция получает на вход строку, проверяет, являются ли значения цифрами и соответствие необходимой длине. Возвращает булевое значение.
- Функция check_date(date) Функция получает на вход строку и проверяет, подходит ли она на формат даты. Возвращает булевое значение.

Блок check отвечает за проверку полученных данных для каждого документа на корректность.

2.1.2 Модуль generate

Функции модуля generate отвечают за генерацию документа, выбранного пользователем.

- Модуль med_generate
  - Функция generate(user)
    Функция получает на вход имя пользователя, достаёт данные из базы данных и сохраняет на сервере сгенерированный документ.
- Модуль vac_generate
  - Функция generate(user)
    Функция получает на вход имя пользователя, достаёт данные из базы данных и сохраняет на сервере сгенерированный документ.
- Модль shotozero_generate
  - Функция generate(user)
    Функция получает на вход имя пользователя, достаёт данные из базы данных и сохраняет на сервере сгенерированный документ.

2.2 Структуры и базы данных

Для хранения данных используется база данных Mongo.

База данных содержит коллекцию объектов, каждый из которых соответствует уникальному пользователю. Каждый объект имеет следующие поля:

- name, никнейм пользователя в telegram;
- firstname, имя;
- secondname, отчество;
- surname, фамилия;
- medOrg, медицинская организация;
- datebitrh, дата рождения;
- gender, пол;
- phone, номер телефона;
- polic, номер полиса;
- policOrg, имя организации, выдавшей полис;
- policDate, дата выдачи полиса;
- state, гражданство;
- passportSeria, серия паспорта;
- passportNumber, номер паспорта;
- passportPlace, место выдачи паспорта;
- passportDate, дата выдачи паспорта;
- adress, адрес регистрации;
- course, номер курса;
- groupnumber, номер группы;
- shotozeroStart, дата начала отдыха в Шотозеро;

- shotozeroYear, год отдыха в Шотозеро;
- date, дата создания заявления;
- housenumber, название института;
- speccode, код направления обучения;
- specname, название направления обучения.

2.3 Стратегия блочного тестирования

Блочное тестирование будет проводиться для функций модулей bot и generate с использованием библиотеки pytest.

Функции модуля bot, являющиеся чекерами и возвращающие значения будут тестироваться методом белого ящика. Для остальных функций, являющихся базами данных, будут проводиться предварительное приведение базы данных в нужное состояние, результат тестирования будет оцениваться по изменению состояния баз данных.

Функции модуля generate, процедуры, будут тестироваться при помощи предварительного приведения базы данных в нужное состояние, и дальнейшей проверкой структуры созданных функцией объектов.

2.4 Стратегия интеграционного тестирования

Интеграционное тестирование будет проводиться для взаимодействий между модулями genetate и bot: тестирование взаимодействия модуля meg (функция get_info()) и модуля generate_med (функция generate()); модуля vacation (функция get_info()) и модуля generate_vacation (функция generate()); модуля shotozero (функция get_info()) и модуля generate_shotozero (функция generate()).

Модули вызываются последовательно друг из друга, предварительно вызываются остальные функции соответствующего модуля из модуля bot для заполнения данных, проверяемых функцией get_info. Будет производится тестирование нисходящих интеграций в соответсвии с архитектурой: функции модуля generate вызываются из функций модуля bot, вызываемых из основной функции запуска телеграм-бота.

Результатом интеграций является сгенерированный файл, для проверки корректности интеграций будем проверять соответсвие полученного файла с ожидаемым.

Тесты будут реализовываться с использованием библиотеки pytest.

2.5 Стратегия аттестационного тестирования

Аттестационные тесты будут проводиться в соответствии со сценариями использования системы вручную.

2.6 Нагрузочное тестирование

В данной группе тестов будет проводиться проверка работы системы на больших количествах параллельных взаимодействий с сервисом.

Тесты будут реализовываться с использованием пользователей-тестировщиков.

3 Детальный план тестов

3.1 Блочные тесты

ID	Б1
Описание теста	Проверка отчистки базы данных
Тип теста	общий
Объект тестирования	Функция clear(message)
Входные данные	Сообщение, содержащее команду «clear» от пользователя username
Косвенные входные данные	База данных содержит непустые данные для пользователя username
Ожидаемый результат	База данных содержит пустые поля для пользователя username

ID	Б2
Описание теста	Проверка отчистки базы данных, когда она ещё пуста
Тип теста	специальный
Объект тестирования	Функция clear(message)
Входные данные	Объект message, где message.chat.username = username, message.chat.text = «clear»
Косвенные входные данные	База данных не содержит данные для пользователя username
Ожидаемый результат	База данных содержит пустые поля для пользователя username

ID	Б3
Описание теста	Проверка добавления поля имени в базу данных
Тип теста	общий
Объект тестирования	Модуль med, get_firstname(message)
Входные данные	Объект message, где message.chat.username = username, message.chat.text = Иванов
Косвенные входные данные	
Ожидаемый результат	База данных содержит значение «Иванов» для поля firstname пользователя username

ID	Б4
Описание теста	Проверка добавления пустого поля имени в базу данных
Тип теста	негативный
Объект тестирования	Модуль med, get_firstname(message)
Входные данные	Объект message, где message.chat.username = username, message.chat.text = null
Косвенные входные данные	База данных содержит пустое значение для поля firstname пользователя username
Ожидаемый результат	База данных содержит пустое значение для поля firstname пользователя username

ID	Б5
Описание теста	Проверка добавления некорректного поля имени в базу данных
Тип теста	позитиынй
Объект тестирования	Модуль med, get_firstname(message)
Входные данные	Объект message, где message.chat.username = username, message.chat.text = ИвАнов
Косвенные входные данные	База данных содержит пустое значение для поля firstname пользователя username
Ожидаемый результат	База данных содержит значение «ИваАнов» для поля firstname пользователя username

ID	Б6
Описание теста	Проверка добавления поля даты в базу данных
Тип теста	общий
Объект тестирования	Модуль med, get_date(message)
Входные данные	Объект message, где message.chat.username = username, message.chat.text = 21.04.2023
Косвенные входные данные	
Ожидаемый результат	База данных содержит значение «21.04.2023» для поля date пользователя username

ID	Б7
Описание теста	Проверка добавления поля даты в базу данных с некорректным значением
Тип теста	негатвный
Объект тестирования	Модуль med, get_date(message)
Входные данные	Объект message, где message.chat.username = username, message.chat.text = 54.04.2023
Косвенные входные данные	
Ожидаемый результат	База данных содержит пустое значение для поля date пользователя username

ID	Б8
Описание теста	Проверка работы чекера имени
Тип теста	общий
Объект тестирования	Модуль checker, check_firstname(firstname)
Входные данные	firstname = «Михаил»
Косвенные входные данные	
Ожидаемый результат	True

ID	Б9
Описание теста	Проверка работы чекера имени
Тип теста	негативный
Объект тестирования	Модуль checker, check_firstname(firstname)
Входные данные	firstname = «Michail»
Косвенные входные данные	
Ожидаемый результат	False

ID	Б10
Описание теста	Проверка работы чекера имени
Тип теста	негативный
Объект тестирования	Модуль checker, check_firstname(firstname)
Входные данные	firstname = «АННА»
Косвенные входные данные	
Ожидаемый результат	False

ID	Б11
Описание теста	Проверка работы чекера фамилии
Тип теста	общий
Объект тестирования	Модуль checker, check_surname(surname)
Входные данные	surname = «Михаил»
Косвенные входные данные	
Ожидаемый результат	True

ID	Б12
Описание теста	Проверка работы чекера фамилии
Тип теста	негативный
Объект тестирования	Модуль checker, check_surname(surname)
Входные данные	surname = «Иванов123»
Косвенные входные данные	
Ожидаемый результат	False

ID	Б13
Описание теста	Проверка работы чекера фамилии
Тип теста	позитивный
Объект тестирования	Модуль checker, check_surname(surname)
Входные данные	surname = «Салтыков-Щедрин»
Косвенные входные данные	
Ожидаемый результат	True

ID	Б14
Описание теста	Проверка работы чекера отчества
Тип теста	общий
Объект тестирования	Модуль checker, check_secondname(secondname)
Входные данные	secondname = «Михаил»
Косвенные входные данные	
Ожидаемый результат	True

ID	Б15
Описание теста	Проверка работы чекера отчества
Тип теста	негативный
Объект тестирования	Модуль checker, check_secondname(secondname)
Входные данные	secondname = «Иванов123»
Косвенные входные данные	
Ожидаемый результат	False

ID	Б16
Описание теста	Проверка работы чекера отчества
Тип теста	негативный
Объект тестирования	Модуль checker, check_secondname(secondname)
Входные данные	secondname = «$_{4>>}$
Косвенные входные данные	
Ожидаемый результат	False

3.2 Аттестационные тесты

ID	A1
Описание теста	Проверка вывода сообщения-приветствия
Тип теста	общий
Объект тестирования	Модуль start
Действия	Запускается телеграм-бот, пользователь переходит по ссылке на чат
Ожидаемый результат	Появляется окно чата с кнопкой «start»

ID	A2
Описание теста	Проверка вывода сообщения-приветствия во время процесса ввода данных
Тип теста	общий
Объект тестирования	Модуль start
Действия	Пользователь вводит сообщение с командой «start» в ответ на запрос бота
Ожидаемый результат	Появляется приветсвенное сообщение, содержащим никнейм пользователя в Telegram

ID	A3
Описание теста	Проверка работы генерации заявления на перевод в поликлинику
Тип теста	общий
Объект тестирования	Модуль med_generate
Действия	Пользователь заходит в диалог с запущенным ботом, нажимает кнопку генерации заявления на перевод в поликлинику. Пользователь вводит запрашиваемые данные.
Ожидаемый результат	Бот отправляет файл, поля которого соответствуют введённым значениям

ID	A4
Описание теста	Проверка работы генерации заявления на каникулы после сдачи экзаменов
Тип теста	общий
Объект тестирования	Модуль vac_generate
Действия	Пользователь заходит в диалог с запущенным ботом, нажимает кнопку генерации заявления на каникулы после сдачи экзаменов. Пользователь вводит запрашиваемые данные.
Ожидаемый результат	Бот отправляет файл, поля которого соответствуют введённым значениям

ID	A5
Описание теста	Проверка работы генерации заявления на отдых в Шотозеро
Тип теста	общий
Объект тестирования	Модуль med_generate
Действия	Пользователь заходит в диалог с запущенным ботом, нажимает кнопку генерации заявления на отдых в Шотозеро. Пользователь вводит запрашиваемые данные.
Ожидаемый результат	Бот отправляет файл, поля которого соответствуют введённым значениям

3.3 Интеграционные тесты

ID	И1
Описание теста	Проверка корректности передачи данных между модулями med и med_generate
Тип теста	общий
Объект тестирования	Модуль med и модуль med_generate
Действия	В модуле med вызывается функция get_info, в которую передаётся стартовое сообщение пользователя. Далее последовательно вызываются остальные функции модуля med, которые на вход получают сообщения пользователя. После завершения выполнения функций вызывается функция generate из модуля med_generate. Функция достаёт из базы данных данные, которые были получены в первом модуле и генерирует нужный документ.
Ожидаемый результат	На сервере должен появиться файл, соответствующий введённым данным

ID	И2
Описание теста	Проверка корректности передачи данных между модулями vacation и vac_generate
Тип теста	общий
Объект тестирования	Модуль vacation и модуль vac_generate
Действия	В модуле vac вызывается функция get_info, в которую передаётся стартовое сообщение пользователя. Далее последовательно вызываются остальные функции модуля vac, которые на вход получают сообщения пользователя. После завершения выполнения функций вызывается функция generate из модуля vac_generate. Функция достаёт из базы данных данные, которые были получены в первом модуле и генерирует нужный документ.
Ожидаемый результат	На сервере должен появиться файл, соответствующий введённым данным

ID	ИЗ
Описание теста	Проверка корректности передачи данных между модулями shotozero и shotozero_generate
Тип теста	общий
Объект тестирования	Модуль shotozero и модуль shotozero_generate
Действия	В модуле med вызывается функция get_info, в которую передаётся стартовое сообщение пользователя. Далее последовательно вызываются остальные функции модуля shotozero, которые на вход получают сообщения пользователя. После завершения выполнения функций вызывается функция generate из модуля shotozero_generate. Функция достаёт из базы данных данные, которые были получены в первом модуле и генерирует нужный документ.
Ожидаемый результат	На сервере должен появиться файл, соответствующий введённым данным
Ожидаемый результат	На сервере должен появиться файл, соответствующий введённым данным

3.4 Нагрузочные тесты

ID	H1
Описание теста	Проверка функционирования бота при большом количестве параллельных подключений
Тип теста	общий
Объект тестирования	Все модули
Действия	10 пользователей одновременно начинают работу с ботом
Ожидаемый результат	Время ожидания ответа бота не превышает минуты, пользователи получают на выходе файл с введёнными данными

4 Примеры тестов

4.1 Блочные тесты

```python
def test_B3():
    collection.bios.delete_one({"name" : "username"})
    collection.bios.insert_one({"name" : "username"})
    chat = types.Chat(id=1, type="privat", username="username")

    msg = types.Message(message_id=123, from_user={"username": "username"},
        date=1635628400, chat=chat, content_type="text", options=[], json_string="{}")
    msg.text = "Иванов"
    get_firstname(msg)
    items = collection.bios.find({"name" : "username"})
    item = items[0]
    firstname = item.get("firstname")
    assert firstname == "Иванов"
```

Рис. 2 – Блочный тест 3

4.2 Интеграционные тесты

```python
def testI2():
    msg = types.Message(message_id=123, from_user={"username": "username"},
        date=1635628400, chat=chat, content_type="text", options=[], json_string="{}")
    msg.text = "Иван"
    get_firstname(msg)
    msg = types.Message(message_id=123, from_user={"username": "username"},
        date=1635628400, chat=chat, content_type="text", options=[], json_string="{}")
    msg.text = "Иванович"
    get_secondname(msg)
    msg = types.Message(message_id=123, from_user={"username": "username"},
        date=1635628400, chat=chat, content_type="text", options=[], json_string="{}")
    msg.text = "Иванов"
    get_surname(msg)
    msg = types.Message(message_id=123, from_user={"username": "username"},
        date=1635628400, chat=chat, content_type="text", options=[], json_string="{}")
    msg.text = "4"
    get_course(msg)
    msg = types.Message(message_id=123, from_user={"username": "username"},
        date=1635628400, chat=chat, content_type="text", options=[], json_string="{}")
    msg.text = "22407"
    get_groupnumber(msg)
    msg = types.Message(message_id=123, from_user={"username": "username"},
        date=1635628400, chat=chat, content_type="text", options=[], json_string="{}")
    msg.text = "123"
    get_housenumber(msg)
    msg = types.Message(message_id=123, from_user={"username": "username"},
        date=1635628400, chat=chat, content_type="text", options=[], json_string="{}")
    msg.text = "919191"
    get_specnumber(msg)
    msg = types.Message(message_id=123, from_user={"username": "username"},
        date=1635628400, chat=chat, content_type="text", options=[], json_string="{}")
    msg.text = "20.11.2023"
    get_date(msg)
    msg = types.Message(message_id=123, from_user={"username": "username"},
        date=1635628400, chat=chat, content_type="text", options=[], json_string="{}")
    msg.text = "2023"
    get_year(msg)
    msg = types.Message(message_id=123, from_user={"username": "username"},
        date=1635628400, chat=chat, content_type="text", options=[], json_string="{}")
    msg.text = ""
    get_info(msg)

    file = open("./" + message.chat.username + "doc.pdf", "rb")
    test = open("./" + message.chat.username + "docTEST.pdf", "rb")

    assert(file == test)
```

Рис. 3 – Интеграционный тест 2

5 Журнал тестирования

5.1 Журнал блочного тестирования

ID	Дата	Ожидаемый результат	Полученный результат	Номер ошибки
Б1	28.11.23	Пустые поля	Пустые поля	
Б2	28.11.23	Пустые поля	Пустые поля	
Б3	28.11.23	username = «Иванов»	username = «Иванов»	
Б4	28.11.23	Пустые поля	Пустые поля	
Б5	28.11.23	username = «ИвАнов»	username = «ИвАнов»	
Б6	28.11.23	date = «21.04.2023»	date = «21.04.2023»	
Б7	28.11.23	Поле date пусто	date = «21.04.2023»	О1
Б8	28.11.23	True	True	
Б9	28.11.23	False	True	О2
Б10	28.11.23	False	True	О3
Б11	28.11.23	True	True	
Б12	28.11.23	False	False	
Б13	28.11.23	True	True	
Б14	28.11.23	True	True	
Б15	28.11.23	False	False	
Б16	28.11.23	False	False	

5.2 Журнал аттестационного тестирования

ID	Дата	Ожидаемый результат	Полученный результат	Номер ошибки
А1	28.11.23	Окно чата, кнопка «Start»	-//-	
А2	28.11.23	Приветственное сообщение с ником пользователя	-//-	
А3	28.11.23	Файл с введёнными данными	-//-	
А4	28.11.23	Файл с введёнными данными	-//-	
А5	28.11.23	Файл с введёнными данными	-//-	

5.3 Журнал интеграционного тестирования

ID	Дата	Ожидаемый результат	Полученный результат	Номер ошибки
И1	28.11.23	Файл с введёнными данными	-//-	
И2	28.11.23	Файл с введёнными данными	-//-	
И3	28.11.23	Файл с введёнными данными	-//-	

5.4 Журнал нагрузочного тестирования

ID	Дата	Ожидаемый результат	Полученный результат	Номер ошибки
Н1	28.11.23	Бот выдаёт корректные данные не дольше чем за 10 секунд	-//-	

6 Журнал ошибок

1. Функция получения и записи даты get_date(date), тест Б7.

   Вводимые данные: Объект message, где message.chat.username = username, message.chat.tex = 54.04.2023

   Ожидаемый результат: значение поля date пользователя username будет пустым.

   Полученный результат: значение поля date пользователя username равно «54.04.2023».

   Ошибка: программа позволяе вносить в документ несуществующее значение даты.

   Причина ошибки: отсутсвие проверки на существование даты в функции check_date.

   Статус ошибки: исправлена.

2. Функция провреки формата имени check_firstname(firstname), тест Б9.

   Вводимые данные: Строка «Michael»

   Ожидаемый результат: False

   Полученный результат: True

   Ошибка: фукция определяет имя, написанное латиницей, как имя с правильным форматом.

   Причина ошибки: отсутсвия проверки введённых символов на принадлежность к кириллице.

   Статус ошибки: исправлена.

3. Функция проверки формата имени check_firstname(firstname), тест Б10.

   Вводимые данные: Строка «АННА»

   Ожидаемый результат: False

   Полученный результат: True

   Ошибка: фукция определяет имя, написанное в верхнем регистре, как имя с правильным форматом.

   Причина ошибки: Отсутсвие проверки символов кроме первого на соответсвие установленному формату.

   Статус ошибки: исправлена.

7 Результаты

Были проведены блочные, аттестационные и интеграционные тестирования системы, а также тестирование системы на нагрузку.

Проведённое тестирование показало, что код программы нуждается в отладке и доработке. Выявленные в ходе тестирования ошибки были устранены. Покрытие тестами составляет 64%, следовательно, программа нуждается в дополнительном покрытии тестами.