

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
Петрозаводский государственный университет (ПетрГУ)
Институт математики и информационных технологий
Кафедра информатики и математического обеспечения

Отчет по курсу «Верификация ПО»

Выполнила:
студентка 6 курса группы 22608
А.С. Бодрякова

Преподаватель:
к.ф-м.н., доцент К. А. Кулаков

Петрозаводск
2018

Содержание

1. Объект тестирования.....	4
1.1 Функциональные требования объекта тестирования.....	4
2. Стратегия тестирования.....	4
2.1 Архитектура объекта тестирования.....	4
2.2 Стратегия блочного тестирования.....	6
2.2.1 Описание методов блочного тестирования.....	6
2.3 Стратегия интеграционного тестирования.....	7
2.4 Стратегия аттестационного тестирования.....	8
2.5 Стратегия специального тестирования.....	8
2.6 Стратегия нагрузочного тестирования.....	8
2.7 Критерий прохождения тестов.....	9
2.8 Критерий приостановки тестов.....	9
2.9 Критерий возобновления тестов.....	9
3. Детальный план тестов.....	9
3.1 Блочные тесты.....	9
3.2 Интеграционные тесты.....	14
3.3 Аттестационные тесты.....	15
3.4 Специальные тесты.....	18
3.5 Нагрузочные тесты.....	20
3.6 Пример реализации теста.....	20
4. Журнал тестирования.....	21
4.1 Журнал блочного тестирования.....	21
4.2 Журнал интеграционного тестирования.....	22
4.3 Журнал аттестационного тестирования.....	22
4.4 Журнал специальных тестов.....	23
4.4 Журнал нагрузочных тестов.....	23
5. Журнал найденных ошибок.....	23

5.1 Отчет об ошибке №1.....	23
5.2 Отчет об ошибке №2.....	23
5.3 Отчет об ошибке №3.....	24
5.4 Отчет об ошибке №4.....	24
5.5 Отчет об ошибке №5.....	24
5.6 Отчет об ошибке №6.....	24
5.7 Отчет об ошибке №7.....	25
6. Результаты.....	25

1. Объект тестирования

Объектом тестирования является веб-сервис для создания и хранения заметок, планирования событий, прикрепление файлов к заметкам и создания напоминаний. Данный сервис реализован на языке php.

1.1 Функциональные требования объекта тестирования

1. Работа с календарем
 - 1.1 Просмотр календаря на неограниченное число месяцев вперед и назад
2. Работа с заметками
 - 2.1 Создание заметки на любой день календаря
 - 2.2 Удаление заметки
 - 2.3 Редактирование заметки

Работа с текстом заметки

 - 2.4 Написание текста заметки
 - 2.5 Редактирование текста заметки

Работа с прикрепленными файлами заметки

 - 2.6 Прикрепление файла к заметке
 - 2.7 Добавление комментария к файлу
 - 2.8 Удаление файла
3. Работа с пользователями
 - 3.1 Регистрация пользователя
 - 3.2 Авторизация уже существующего пользователя

2. Стратегия тестирования

2.1 Архитектура объекта тестирования



Компоненты системы в вышеописанной схеме:

- Клиент – человек, взаимодействующий с сервером посредством браузера и HTTP-запросов.
- Web-сервер, хранящий в себе исходные коды и обеспечивающий их выполнение.
- База данных – объект, используемый для хранения информации, обособлен от Web-сервера.

Клиент представляет собой:

- Навигация: отображение главного меню.
- Начальная страница: страница приветствия с возможностью входа или регистрации.
- Структура страницы: модуль отображает основной контент посередине, по бокам - блоки, отображающие вспомогательный контент.

Web-сервер состоит из модулей и блоков:

- Core.php - ядро системы, класс, управляющий загрузкой системы.
- Database.php - класс для работы с базой данных.
- Shape.php - класс для работы с визуализацией.
- ModuleCalendar.php - класс для работы с календарём.
- ModuleNotes.php - класс для работы с заметками.
- ModuleProfile.php - класс для работы с профилями пользователей.
- ModuleAuthorization.php - класс для работы с авторизацией.
- ModuleMainPage.php - класс для работы с главной страницей
- ModuleMessages.php - класс для работы с сообщениями.
- BlockMenu.php - блок меню.
- BlockDoing.php - блок близлежащих событий.
- BlockSearch.php - блок поиска пользователей по логину.
- BlockMessages.php - блок пришедших сообщений.
- BlockCategories.php - блок категорий.
- BlockFavourites.php - блок избранных пользователей.

Взаимодействие компонентов описывается следующим образом:

1. Клиент посредством браузера формирует запрос к серверу.
2. Web-сервер получает запрос и обрабатывает его, обращаясь к базе данных по мере необходимости.
3. Web-сервер оформляет ответ в виде, воспринимаемом браузером, отправляет клиенту ответ.
4. Браузер клиента обрабатывает ответ и выдаёт его клиенту.

2.2 Стратегия блочного тестирования

Сервис в первую очередь будет подвергнут блочному тестированию. В блочном тестировании участвуют следующие методы:

- `bool public static function actionShow ();`
- `bool public static function actionEdit_Note ();`
- `bool public static function actionSave_Note ();`
- `bool public static function actionDelete_Note ();`
- `bool public static function actionAuthorization ();`
- `bool public static function actionRegistration ();`
- `bool public static function actionCheck_Authorization ();`
- `bool public static function actionCheck_Registration ();`

2.2.1 Описание методов блочного тестирования

1. Модуль: "ModuleCalendar.php" — класс для работы с календарем.
 - метод: "bool public static function actionShow ()" — метод, записывающий html-код календаря во внутреннее хранилище `Core::content['module']` в зависимости от параметров GET-строки. Возвращает `true` в случае успеха или `false` в случае ошибки.
2. Модуль: "ModuleNotes.php" — класс для работы с заметками.
 - метод: "bool public static function actionEdit_Note ()" — метод формирования страницы редактирования или создания заметки. Записывает html-код страницы во внутреннее хранилище `Core::content['module']`. Если `ID = -1`, то создаётся новая заметка. Если `ID != -1` и `>= 0`, то редактируется уже существующая заметка. ID заметки берётся из GET-строки, проверяются права на редактирование заметки.
 - метод: "bool public static function actionSave_Note ()" — метод сохранения заметки по её ID. Если `ID = -1`, то создаётся новая заметка. Если `ID != -1` и `>= 0`, то редактируется уже существующая заметка. ID заметки берётся из GET-строки, проверяются права на редактирование заметки. Возвращает `true` в случае успеха или `false` в случае ошибки. Перенаправляет на страницу просмотра созданной или изменённой заметки.
 - метод: "bool public static function actionDelete_Note ()" — метод удаления заметки по её ID. ID заметки берётся из GET-строки, проверяются права на удаление заметки. Возвращает `true` в случае успеха или `false` в случае ошибки.

3. Модуль: "ModuleAuthorization.php" — класс для работы с авторизацией.

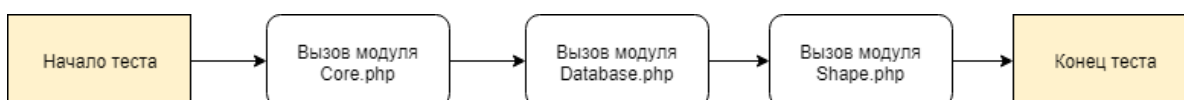
- метод: "bool public static function actionAuthorization ()" — метод формирования страницы авторизации пользователя. Записывает html-код страницы авторизации во внутреннее хранилище Core::content['module']. В зависимости от параметра GET-строки mode формирует страницу первой авторизации (mode=first) или повторной при ошибке (mode=error). Возвращает true в случае успеха или false в случае ошибки.
- метод: "bool public static function actionRegistration ()" — метод формирования страницы регистрации пользователя. Записывает html-код страницы регистрации во внутреннее хранилище Core::content['module']. Возвращает true в случае успеха или false в случае ошибки.
- метод: "bool public static function actionCheck_Authorization ()" — метод проверки введённых пользователем данных при авторизации. Данные посылаются методом POST. В зависимости от результата проверки перенаправляет на:
 - просмотр месяца и формирует массив COOKIES;
 - повторную авторизацию.
 Возвращает true в случае успеха или false в случае ошибки.
- метод: "bool public static function actionCheck_Registration ()" — метод проверки введённых пользователем данных при регистрации. Данные посылаются методом POST. В зависимости от результата проверки:
 - регистрирует нового пользователя, перенаправляет на просмотр месяца и формирует массив COOKIES;
 - перенаправляет на повторную регистрацию.
 Возвращает true в случае успеха или false в случае ошибки.

2.3 Стратегия интеграционного тестирования

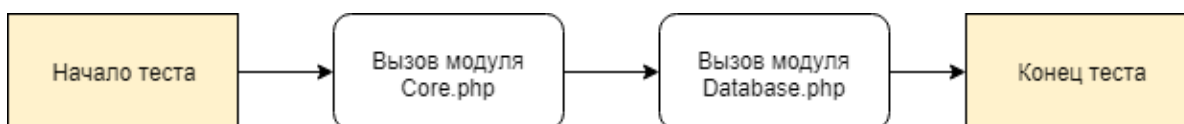
Второй этап — интеграционное тестирование. Тестирование будет проведено для следующих взаимодействий между модулей:

- Core.php — ядро системы, класс, управляющий загрузкой системы.
- Database.php — класс для работы с базой данных.
- Shape.php — класс для работы с визуализацией.

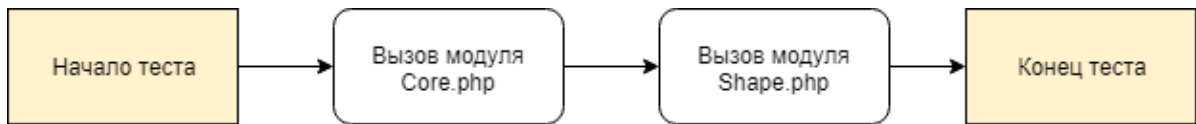
1. Взаимодействие между Core.php, Database.php и Shape.php



2. Взаимодействие между Core.php и Database.php



3. Взаимодействие между Core.php и Shape.php



2.4 Стратегия аттестационного тестирования

Третий этап — аттестационное тестирование. Проверяются функциональные требования, а именно:

1. Просмотр календаря на неограниченное число месяцев вперед и назад.
2. Добавление заметки.
3. Редактирование заметки.
4. Удаление заметки.
5. Добавление текста к заметке.
6. Редактирование текста к выбранной заметке.
7. Добавление файла к заметке.
8. Удаление файла у заметки.
9. Авторизация пользователя.
10. Регистрация пользователя

2.5 Стратегия специального тестирования

Четвертый этап тестирования — специальное тестирование. Проверяет, как поведет себя структура веб-страниц при разном количестве контента.

Проверяются следующие страницы:

- Страница регистрации;
- Страница авторизации;
- Страница календаря;
- Страница заметок;
- Страница создания / редактирования заметки.

2.6 Стратегия нагрузочного тестирования

Последний этап тестирования — нагрузочное тестирование. Проверяет работу системы при большом объеме данных.

2.7 Критерий прохождения тестов

Тест считается пройденным если результат равен ожидаемому, который описан в разделе стратегии тестирования.

Тест считается не пройденным если результат не равен ожидаемому.

2.8 Критерий приостановки тестов

Тестирование должно быть приостановлено, если при прохождении теста было потрачено больше 3 секунд. Такой тест считается не удавшимся.

2.9 Критерий возобновления тестов

Тестирование возобновляется после исправления ошибки, выявленных при предыдущем тестировании.

3. Детальный план тестов

3.1 Блочные тесты

№	V1
Цель теста	Проверка правильного вывода календаря
Тип теста	Общий
Объект теста	Метод "bool public static function actionShow ()"
Входные данные	GET-строка (year, month) имеют валидные значения (year = положительное число&month = положительное число).
Ожидаемый результат	Вывод правильного календаря в соответствии с входными данными.

№	V2
Цель теста	Проверка вывода создания / редактирования заметки, правильность вывода текущих характеристик заметки
Тип теста	Общий
Объект теста	Метод "bool public static function actionEdit_Note ()"
Входные данные	GET-строка (ID = положительное число)
Ожидаемый результат	Открытие страницы создания / редактирования заметки.

№	В3
Цель теста	Проверка правильного сохранения заметки
Тип теста	Общий
Объект теста	Метод "bool public static function actionSave_Note ()"
Входные данные	GET-строка (ID = положительное число), POST - массив данных отправленной формы
Ожидаемый результат	Сохранённая заметка при валидных данных формы и сообщение об ошибке при невалидных данных формы.

№	В4
Цель теста	Проверка правильного удаления заметки.
Тип теста	Общий
Объект теста	Метод "bool public static function actionDelete_Note ()"
Входные данные	GET-строка (ID = положительное число)
Ожидаемый результат	Заметка удаляется из БД.

№	В5
Цель теста	Проверка правильного формирования страницы авторизации пользователя в соответствии с параметром mode
Тип теста	Общий
Объект теста	Метод "bool public static function actionAuthorization ()"
Входные данные	Login = строка password = строка
Ожидаемый результат	Вход пользователя. Отображение календаря в личном кабинете.

№	B6
Цель теста	Проверка правильного формирования страницы регистрации пользователя
Тип теста	Общий
Объект теста	Метод "bool public static function actionRegistration ()"
Входные данные	Login = строка password = строка, повторный password = строка (тот же самый пароль)
Ожидаемый результат	Пользователь зарегистрирован, переход на страницу «Календарь».

№	B7
Цель теста	Проверка правильной обработки посланных пользователем данных при авторизации
Тип теста	Общий
Объект теста	Метод "bool public static function actionCheck_ Authorization ()"
Входные данные	POST - массив данных отправленной формы
Ожидаемый результат	При валидных данных: перенаправление на просмотр календаря При невалидных данных: перенаправление на повторную авторизацию

№	B8
Цель теста	Проверка правильной обработки посланных пользователем данных при регистрации
Тип теста	Общий
Объект теста	Метод "bool public static function actionCheck_Registration ()"
Входные данные	POST - массив данных отправленной формы
Ожидаемый результат	При валидных данных: создание нового пользователя, перенаправление на просмотр календаря При невалидных данных: перенаправление на повторную регистрацию

№	B9
Цель теста	Проверка вывода ошибки неправильного ввода года или месяца
Тип теста	Негативный
Объект теста	Метод "bool public static function actionShow ()"
Входные данные	GET-строка (year, month) имеют невалидные значения (year = строка&month = строка)
Ожидаемый результат	Вывод ошибки, сообщающий, что выбран неправильный год или месяц

№	B10
Цель теста	Проверка вывода ошибки неправильного ввода года или месяца
Тип теста	Негативный
Объект теста	Метод "bool public static function actionShow ()"
Входные данные	GET-строка (year, month) имеют невалидные значения (year = положительное число, выходящее из интервала значений для переменной&month = отрицательное число)
Ожидаемый результат	Вывод ошибки, сообщающий, что выбран неправильный год или месяц

№	B11
Цель теста	Проверка правильного формирования страницы авторизации пользователя в соответствии с параметром mode
Тип теста	Негативный
Объект теста	Метод "bool public static function actionAuthorization ()"
Входные данные	GET-строка: login = пустая строка password = пустая строка
Ожидаемый результат	Пользователь остается на странице авторизации. Поля «Логин» и «Пароль» выводят дополнительную информацию о заполнении полей.

№	B12
Цель теста	Проверка правильного формирования страницы авторизации незарегистрированного пользователя
Тип теста	Негативный
Объект теста	Метод "bool public static function actionAuthorization ()"
Входные данные	Login = строка password = строка
Ожидаемый результат	Пользователь остается на странице авторизации. Выводится ошибка о несуществующем пользователе.

№	B13
Цель теста	Проверка правильного формирования страницы регистрации пользователя
Тип теста	Негативный
Объект теста	Метод "bool public static function actionRegistration ()"
Входные данные	Login = строка password = строка, повторный password = строка, не совпадающая с предыдущей
Ожидаемый результат	Пользователь не зарегистрирован. Сброс паролей, выводится ошибка «Пароли не совпадают».

№	B14
Цель теста	Проверка вывода создания / редактирования заметки, правильность вывода текущих характеристик заметки
Тип теста	Негативный
Объект теста	Метод "bool public static function actionEdit_Note ()"
Входные данные	GET-строка (ID = отрицательное число)
Ожидаемый результат	Сообщение об ошибке «Данной страницы не существует»

3.2 Интеграционные тесты

№	И1
Цель теста	Проверить работу вызова модулей Shape.php и Database.php из Core.php
Тип теста	Общий
Объект теста	Core.php, Database.php, Shape.php
Входные данные	-
Ожидаемый результат	Вызов функций произошел.

№	И2
Цель теста	Проверить работу модуля Core.php с полученными данными от модуля Database.php
Тип теста	Общий
Объект теста	Database.php, Core.php
Входные данные	<code>\$content = array ();</code> Содержимое базы данных.
Ожидаемый результат	Корректный вызов функции, ожидаемый результат работы этой функции. Система инициализирована данными.

№	И3
Цель теста	Проверить работу модуля Core.php с полученными данными от модуля Shape.php
Тип теста	Общий
Объект теста	Shape.php, Core.php
Входные данные	<code>\$content = array ();</code> Данные с шаблона.
Ожидаемый результат	Корректный вызов функции, ожидаемый результат работы этой функции. Core.php получает массив данных вызванного шаблона.

3.3 Аттестационные тесты

№	A1 (ФТ 1.1)
Цель теста	Сервис должен предоставить возможность просмотра календаря на неограниченное число месяцев вперед и назад
Тип теста	Общий
Объект теста	Страница календаря
Входные данные	Авторизоваться и перейти на вкладку "Календарь". Щелкнуть на соответствующие кнопки для перехода по месяцам вперед или назад
Ожидаемый результат	На экране будет отображаться тот месяц, на который пользователь перешел, нажав нужное количество раз на кнопку

№	A2 (ФТ 2.1)
Цель теста	Сервис должен предоставить возможность создание заметки на любой день календаря
Тип теста	Общий
Объект теста	Страница заметок
Входные данные	Авторизоваться и перейти на вкладку "Заметки"-"Мои заметки". Нажать на кнопку "Добавить".
Ожидаемый результат	На экране появится окно. По интуитивно понятному интерфейсу пользователь заполняет поля новой заметки

№	A3 (ФТ 2.2)
Цель теста	Сервис должен предоставить возможность удаление заметки
Тип теста	Общий
Объект теста	Страница заметок
Входные данные	Авторизоваться и перейти на вкладку "Заметки"-"Мои заметки". Выбрать нужную заметку, нажать на нее, а затем на появившуюся кнопку "Удалить".
Ожидаемый результат	Заметка должна удалиться из списка заметок

№	A4 (ФТ 2.3)
Цель теста	Сервис должен предоставить возможность редактирования заметки
Тип теста	Общий
Объект теста	Страница заметок
Входные данные	Авторизоваться и перейти на вкладку "Заметки"- "Мои заметки". Выбрать нужную заметку, нажать на нее,а затем на появившуюся кнопку "Редактировать".
Ожидаемый результат	Заметка должна быть активна для редактирования

№	A4 (ФТ 2.4)
Цель теста	Сервис должен предоставить возможность написания текста заметки
Тип теста	Общий
Объект теста	Страница заметок
Входные данные	Авторизоваться и перейти на вкладку "Заметки"- "Мои заметки". Нажать на кнопку "Добавить".
Ожидаемый результат	Строка с заметкой будет активна для редактирования

№	A5 (ФТ 2.5)
Цель теста	Сервис должен предоставить возможность редактирование текста заметки
Тип теста	Общий
Объект теста	Страница заметок
Входные данные	Авторизоваться и перейти на вкладку "Заметки"- "Мои заметки". Выбрать нужную заметку, нажать на нее,а затем на появившуюся кнопку "Редактировать".
Ожидаемый результат	Строка с заметкой будет активна для редактирования

№	A6 (ФТ 2.6)
Цель теста	Сервис должен предоставить возможность прикрепление файла к заметке
Тип теста	Общий
Объект теста	Страница заметок
Входные данные	Авторизоваться и перейти на вкладку "Заметки"- "Мои заметки". Выбрать нужную заметку, нажать на нее,а затем на появившуюся кнопку "Добавить файл".
Ожидаемый результат	Сервис предоставит возможность прикрепить файл-выбрать файл со своего компьютера

№	A7 (ФТ 2.7)
Цель теста	Сервис должен предоставить возможность добавить комментарий к файлу
Тип теста	Общий
Объект теста	Страница заметок
Входные данные	Авторизоваться и перейти на вкладку "Заметки"- "Мои заметки". Выбрать нужную заметку, нажать на нее,а затем на появившуюся кнопку "Добавить файл". Добавить в поле «Комментарий» ваш комментарий.
Ожидаемый результат	Сервис предоставит возможность добавить комментарий в поле для комментария

№	A8 (ФТ 2.8)
Цель теста	Сервис должен предоставить возможность удалить файл, прикрепленный к заметке
Тип теста	Общий
Объект теста	Страница заметок
Входные данные	Авторизоваться и перейти на вкладку "Заметки"- "Мои заметки". Выбрать нужную заметку, нажать на нее, затем удалить файл, прикрепленный к заметке. Нажать «Удалить файл»
Ожидаемый результат	Сервис удалит файл, при этом заметка останется

№	A9 (ФТ 3.1)
Цель теста	Сервис должен предоставить возможность регистрации пользователя
Тип теста	Общий
Объект теста	Страница регистрации
Входные данные	Перейти на главную страницу сервиса. Выбрать в меню "Log in"
Ожидаемый результат	Сервис предоставит окно входа, там же можно выбрать "Регистрация" и поля для входа расширятся. После регистрации пользователь будет перенаправлен в личный кабинет

№	A10 (ФТ 3.2)
Цель теста	Сервис должен предоставить возможность входа в систему с помощью логина / пароля
Тип теста	Общий
Объект теста	Страница авторизации
Входные данные	Перейти на главную страницу сервиса. Выбрать в меню "Log in"
Ожидаемый результат	Сервис предоставит окно входа, при вводе уже зарегистрированного логина / пароля появляется страница календаря.

3.4 Специальные тесты

№	C1
Цель теста	Проверить, как поведет себя структура веб-страниц при разном количестве контента
Тип теста	Общий
Объект теста	Страница регистрации
Входные данные	Страница регистрации
Ожидаемый результат	Вёрстка не ломается при добавлении в элементы большего количества текста. Текст не выпадает из объектов. Переполнение контентными блоками не приводит к нарушению сетки.

№	C2
Цель теста	Проверить, как поведет себя структура веб-страниц при разном количестве контента
Тип теста	Общий
Объект теста	Страница авторизации
Входные данные	Страница авторизации
Ожидаемый результат	Вёрстка не ломается при добавлении в элементы большего количества текста. Текст не выпадает из объектов. Переполнение контентными блоками не приводит к нарушению сетки.

№	C3
Цель теста	Проверить, как поведет себя структура веб-страниц при разном количестве контента
Тип теста	Общий
Объект теста	Страница календаря
Входные данные	Страница календаря
Ожидаемый результат	Вёрстка не ломается при добавлении в элементы большего количества текста. Текст не выпадает из объектов. Переполнение контентными блоками не приводит к нарушению сетки.

№	C4
Цель теста	Проверить, как поведет себя структура веб-страниц при разном количестве контента
Тип теста	Общий
Объект теста	Страница заметок
Входные данные	Страница заметок
Ожидаемый результат	Вёрстка не ломается при добавлении в элементы большего количества текста. Текст не выпадает из объектов. Переполнение контентными блоками не приводит к нарушению сетки.

№	C5
Цель теста	Проверить, как поведет себя структура веб-страниц при разном количестве контента
Тип теста	Общий
Объект теста	Страница создания / редактирования заметки
Входные данные	Страница создания / редактирования заметки
Ожидаемый результат	Вёрстка не ломается при добавлении в элементы большего количества текста. Текст не выпадает из объектов. Переполнение контентными блоками не приводит к нарушению сетки.

3.5 Нагрузочные тесты

№	H1
Цель теста	Проверить работу модуля Database.php при большом объеме данных
Тип теста	Негативный
Объект теста	Database.php
Входные данные	База данных
Ожидаемый результат	База данных считана за 1 секунду

3.6 Пример реализации теста

Пример реализации блочного теста:

```
include ('php/classes/ModuleAuthorization.php')

public static function TestCheck_Registration() {
    $_POST ['login'] = sha1 ('bodr');
    $_POST ['password'] = sha1 ('password');

    actionCheck_Registration();

    if ($_COOKIE['login'] === $_POST ['login'] && $_COOKIE['password'] === $_POST ['password']) {
        header ('location: ?module=calendar&action=show');
    }
    else {
        header ('location: ?module=main_page&action=main');
    }
}
```

4. Журнал тестирования

4.1 Журнал блочного тестирования

№	Дата	Тестирующий	Объект	Кол-во тестов	Кол-во ошибок
Б1	04.12.18	Бодрякова	метод:"bool public static function actionShow ()"	3	0
Б2	04.12.18	Бодрякова	метод:"bool public static function actionEdit_Note ()"	1	0
Б3	04.12.18	Бодрякова	метод:"bool public static function actionSave_Note ()"	1	0
Б4	04.12.18	Бодрякова	метод:"bool public static function actionDelete_Note ()"	1	0
Б5	04.12.18	Бодрякова	метод:"bool public static function actionAuthorization ()"	3	0
Б6	04.12.18	Бодрякова	метод:"bool public static function actionRegistration ()"	3	0
Б7	04.12.18	Бодрякова	метод:"bool public static function actionCheck_Authorization ()"	3	0
Б8	04.12.18	Бодрякова	метод:"bool public static function actionCheck_Registration ()"	3	0
Б9	10.12.18	Бодрякова	метод:"bool public static function actionShow ()"	1	1 (Отчет об ошибке №1)
Б10	10.12.18	Бодрякова	метод:"bool public static function actionShow ()"	1	1 (Отчет об ошибке №2)
Б11	10.12.18	Бодрякова	метод "bool public static function actionAuthorization ()"	1	1 (Отчет об ошибке №3)
Б12	10.12.18	Бодрякова	метод "bool public static function actionAuthorization ()"	1	1 (Отчет об ошибке №4)
Б13	10.12.18	Бодрякова	метод "bool public static function actionRegistration"	1	1 (Отчет об ошибке №5)

)"		
Б14	10.12.18	Бодрякова	метод:"bool public static function actionEdit_Note ()"	1	1 (Отчет об ошибке №6)

4.2 Журнал интеграционного тестирования

№	Дата	Тестирующий	Объект	Кол-во тестов	Кол-во ошибок
И1	06.12.18	Бодрякова	Core.php	2	0
И2	06.12.18	Бодрякова	Database.php	2	0
И3	06.12.18	Бодрякова	Shape.php	2	0

4.3 Журнал аттестационного тестирования

№	Дата	Тестирующий	Объект	Кол-во тестов	Кол-во ошибок
A1	04.12.18	Бодрякова	Страница календаря	2	0
A2	04.12.18	Бодрякова	Страница заметок	2	0
A3	04.12.18	Бодрякова	Страница заметок	2	0
A4	04.12.18	Бодрякова	Страница заметок	2	0
A5	04.12.18	Бодрякова	Страница заметок	2	0
A6	04.12.18	Бодрякова	Страница заметок	2	0
A7	04.12.18	Бодрякова	Страница заметок	2	0
A8	04.12.18	Бодрякова	Страница заметок	2	0
A9	04.12.18	Бодрякова	Страница регистрации	2	0
A10	04.12.18	Бодрякова	Страница авторизации	2	0

4.4 Журнал специальных тестов

№	Дата	Тестирующая	Объект	Кол-во тестов	Кол-во ошибок
C1	10.12.18	Бодрякова	Страница регистрации	2	0
C2	10.12.18	Бодрякова	Страница авторизации	2	0
C3	10.12.18	Бодрякова	Страница календаря	2	0
C4	10.12.18	Бодрякова	Страница заметок	2	0
C5	10.12.18	Бодрякова	Страница создания / редактирования заметки	2	0

4.4 Журнал нагрузочных тестов

№	Дата	Тестирующая	Объект	Кол-во тестов	Кол-во ошибок
H1	10.12.18	Бодрякова	Database.php	1	1 (Отчет об ошибке №7)

5. Журнал найденных ошибок

5.1 Отчет об ошибке №1

Дата составления отчета: 10.12.2018

Номер теста: Б9

Входные данные: year = рк&month = ыв,

Ожидаемый результат: Вывод ошибки, сообщающий, что выбран неправильный год или месяц

Фактический результат: Отображение страницы «Не найдена страница»

5.2 Отчет об ошибке №2

Дата составления отчета: 10.12.2018

Номер теста: Б10

Входные данные: year = 4000&month = -1

Ожидаемый результат: Вывод ошибки, сообщающий, что выбран неправильный год или месяц

Фактический результат: Отображение страницы «Не найдена страница»

5.3 Отчет об ошибке №3

Дата составления отчета: 10.12.2018

Номер теста: Б11

Входные данные: login = &password =

Ожидаемый результат: Пользователь остается на странице авторизации. Поля «Логин» и «Пароль» выводят дополнительную информацию о заполнении полей.

Фактический результат: Переход на страницу календаря.

5.4 Отчет об ошибке №4

Дата составления отчета: 10.12.2018

Номер теста: Б12

Входные данные: login = hfhfhf&password = hfhfhf

Ожидаемый результат: Пользователь остается на странице авторизации. Выводится ошибка о несуществующем пользователе.

Фактический результат: Переход на страницу календаря.

5.5 Отчет об ошибке №5

Дата составления отчета: 10.12.2018

Номер теста: Б13

Входные данные: login = bodr&password = password, повторный password = ddhfhj

Ожидаемый результат: Пользователь не зарегистрирован. Ожидается сброс паролей, выводится ошибка «Пароли не совпадают».

Фактический результат: Отображение страницы «Не найдена страница»

5.6 Отчет об ошибке №6

Дата составления отчета: 10.12.2018

Номер теста: Б14

Входные данные: note_id = -1

Ожидаемый результат: Сообщение об ошибке «Данной страницы не существует»

Фактический результат: Отображение страницы «Не найдена страница»

5.7 Отчет об ошибке №7

Дата составления отчета: 10.12.2018

Номер теста: Н1

Входные данные: БД. Объем данных 5000 строк.

Ожидаемый результат: Вызов БД. Данные переданы модулю Core.php за 1 секунду.

Фактический результат: Данные считались за 3 секунды.

6. Результаты

В ходе блочного тестирования были найдены 6 ошибок, которые существенно влияют на работу системы. При проведении нагрузочного теста, система так же не справилась, но данная ошибка не является критической, однако соответствует критерию приостановки теста, поставленной в данной работе. В ходе интеграционного, аттестационного и специального тестирования не было выявлено ошибок.

Полученные сведения о тестировании могут сказать, что на данный момент система не достаточно работоспособна. Основные проблемы возникли на стадии регистрации и авторизации пользователей, что является важным фактором для использования системы. Так же есть проблемы с проверкой входящих данных и выводом правильных ошибок.