

# -План тестирования

## 1 Введение

Данный план тестирования разработан для проверки работоспособности всех модулей разработанного проекта, а так же для выявления ошибок и неисправностей в работе приложения при выполнении определенных сценариев действий пользователя. На основании выводов, представленных в данном документе, разработчик может определить, какие изменения необходимы в новой версии ПО и какие корректировки требуется осуществить для улучшения работоспособности и производительности приложения. В данном документе описаны подходы и аспекты тестирования данного программного обеспечения, по результатам которого возможно устранение каких-либо логических ошибок и избежание некорректного исполнения задач.

## 2 Объект тестирования

### 2.1 Описание проекта и его компонентов

Приложение "Easy To Study" нацелено на облегчение учебного процесса студента. Пользователю предоставляется возможность удобного контроля таких показателей, как:

- 1) дедлайны.
- 2) отметки.
- 3) пропуски занятий.
- 4) заметки любого характера.

Также есть возможность изменения внесенной информации, просмотр статистики по пропускам и сданным лабораторным работам, а также хранение всей информации в текстовых файлах.

### 2.2 Атрибуты качества (по ISO 9126)

Функциональность	Пригодность для применения
	Корректность
Надежность	Завершенность
	Восстанавливаемость
Практичность	Понятность
	Простота использования
Эффективность	Использование ресурсов
Сопровождаемость	Стабильность
Мобильность	Мультиплатформенность
	Сосуществование

## 3 Риски

В качестве начальной инициализации списка предметов, с которым в дальнейшем пользователь будет осуществлять работу, используется текстовый файл. При первом использовании приложения пользователь должен создать дерево соответствующих каталогов и текстовых документов (рис. 1). В корневом каталоге Files требуется создать два текстовых файла subjects.txt и note.txt соответственно. В файле subject.txt необходимо организовать структуры типа <кол-во лаб.раб., предмет>, пример – рис. 2. В каталогах absences и marks требуется создать файлы, названия которых соответствуют названиям предметов, перечисленных в файле subjects.txt. Остальные файлы оставить пустыми.

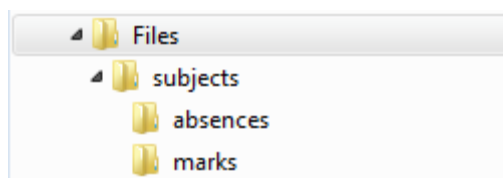


Рис.1 – дерево каталогов.

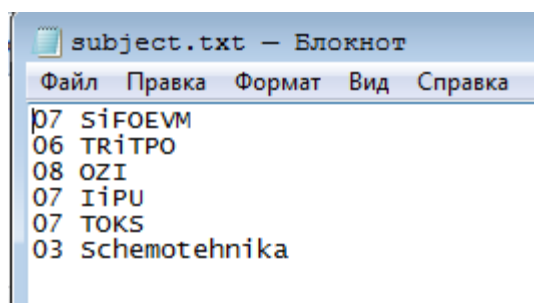


Рис.2 – организация файла subjects.txt

Данная инициализация непосредственно влияет на качество продукта, т.к. несоблюдение инструкций описанных выше ведет к некорректному исполнению функций разработанного приложения.

## 4 Аспекты тестирования

### 4.1 Пригодность для применения

Необходимо удостовериться, может ли в целом приложения выполнять главную задачу, поставленную в документе требований к ПО – облегчение учебного процесса.

### 4.2 Корректность

Корректность исполнения всех заявленных и реализованных функций гарантируют целостную успешность приложения.

### 4.3 Завершенность

Конечно же, в приложении должны быть реализованы все функции, заявленные в требованиях к проекту.

### 4.4 Восстанавливаемость

В случае каких-либо непредвиденных ошибок системы, либо обычного выхода из приложения, все данные занесенные пользователем должны каким-то образом храниться и перезаписываться в случае изменения их пользователем. Если же, файлы каким-то образом были утеряны, пользователь должен иметь возможность самостоятельного создания их собственными усилиями.

#### 4.5 Понятность и простота использования

Имея цель, достижение которой возможно средствами данного ПО, пользователь должен интуитивно понимать, в каком направлении ему следует двигаться и какие действия совершить для удовлетворения своих потребностей.

#### 4.6 Использование ресурсов

Все файлы, используемые при работе с приложением – файлы, в которые ведется запись всей, добавленной пользователем информации должны считываться корректно. Также необходимо удостовериться, что программа не использует достаточно большого кол-ва аппаратных ресурсов.

#### 4.7 Мультиплатформенность

Требуется установить – имеется ли возможность запуска приложения на платформе Windows/UNIX, чего необходимо было достичь в соответствии с документом требований.

#### 4.8 Стабильность

Не должно возникать никаких непредвиденных ситуаций и логических ошибок при использовании, разработанного ПО.

#### 4.9 Сосуществование

В процессе тестирования, требуется удостовериться, что приложение не вступает в конфликты с иным ПО и ни каким образом не оказывает влияние на систему.

### 5 Подходы к тестированию

Каждый из пунктов аспектов тестирования (п.п. 4.1 – 4.9) будет протестирован с помощью пользовательских тестов, сценарии которых описаны в пунктах ниже.

## 6 Представления результатов

### 6.1 Тестовый сценарий пригодности для применения

#### 6.1.1 Идентификатор

Пригодность для применения

#### 6.1.2 Назначение

В ходе исполнения данного сценария требуется удостовериться, может ли в целом приложение выполнять главную задачу, поставленную в документе требований к ПО – облегчение учебного процесса.

### 6.1.3 Сценарий

- 1) В запущенном приложении нажать кнопку "orginize".
- 2) Перейти к выполнению функции добавления информации путем нажатия кнопки "add...".
- 3) Выбрать один из предложенных видов информации.
- 4) Заполнить предложенные поля.
- 5) Нажать кнопку "apply".
- 6) Повторить п.п. 2-5 для всех видов информации.
- 7) Перейти к редактированию информации (нажать кнопку "edit info").
- 8) Заполнить предложенные поля.
- 9) Нажать кнопку "apply".
- 10) Повторить п.п. 7-9 для всех видов информации.
- 11) Проверить наличие всех добавлений и изменений на таймлайне.
- 12) Нажать кнопку "statistics".
- 13) В случае соответствия используемых выше функций требованиям, описанным в соответствующем документе считать тестовый сценарий выполненным успешно.

### 6.1.4 Ожидаемый результат

Приложение выполняет главную задачу, поставленную в документе требований – облегчение учебного процесса.

## 6.2 Тестовый сценарий корректности

### 6.2.1 Идентификатор

Корректность

### 6.2.2 Назначение

В ходе исполнения данного сценария требуется удостовериться, что все заявленные и реализованные функции выполняют именно те функции, которые подразумеваются.

### 6.2.3 Сценарий

- 1) В запущенном приложении нажать кнопку "orginize".
- 2) Перейти к выполнению функции добавления информации путем нажатия кнопки "add.../edit info".
- 3) Выбрать один из предложенных видов информации.
- 4) Заполнить предложенные поля.
- 5) Нажать кнопку "apply".
- 6) Проверить результат работы функции на таймлайне или в окне статистики (нажать кнопку "statistics").

7) В случае выполнения функции таким образом, каким подразумевал ее тестировщик, имея конкретную цель, считать сценарий тестирования выполненным успешно.

#### 6.2.4 Ожидаемый результат

Все заявленные и реализованные функции гарантируют целостную успешность приложения.

### 6.3 Тестовый сценарий завершенности

#### 6.3.1 Идентификатор

Завершенность

#### 6.3.2 Назначение

В ходе исполнения данного сценария требуется удостовериться, что в приложении реализованы все функции, заявленные в требованиях к проекту.

#### 6.3.3 Сценарий

- 1) Открыть файл требований к проекту.
- 2) В соответствии с каждым пунктом файла требований пройти один из пунктов тестового сценария "корректность".
- 3) Проверить соответствие выполненной функции проверяемому требованию.
- 4) Повторить п.п. 2-3.
- 5) При соответствии всех результатов выполненных функций требования считать тестовый сценарий выполненным успешно.

#### 6.3.4 Ожидаемый результат

В приложении реализованы все функции, заявленные в требованиях к проекту.

### 6.4 Тестовый сценарий восстанавливаемости

#### 6.4.1 Идентификатор

Восстанавливаемость

#### 6.4.2 Назначение

В ходе исполнения данного сценария требуется удостовериться, что в случае каких-либо непредвиденных ошибок системы, либо обычного выхода из приложения, все данные занесенные пользователем каким-то образом сохраняются и перезаписываются в случае изменения их пользователем. Также требуется установить, что при потере файлов инициализации, пользователь может создать их собственными усилиями.

#### 6.4.3 Сценарий

- 1) Удалить каталог "files".
- 2) Перейти к тестовому сценарию "использование ресурсов" (п. 8).
- 3) В соответствии с данным пунктом, восстановить все

файлы.

4) Запустить программу и пройти сценарий ”завершенность”.

#### 6.4.4 Ожидаемый результат

В случае непредвиденных ошибок системы, все данные занесенные пользователем сохраняются и перезаписываются корректно. При потере файлов инициализации пользователь смог полностью восстановить их собственными усилиями.

### 6.5 Тестовый сценарий понятности и простоты использования

#### 6.5.1 Идентификатор

Понятность и простота использования

#### 6.5.2 Назначение

В ходе исполнения данного сценария требуется удостовериться, имея цель, достижение которой возможно средствами данного ПО, пользователь должен интуитивно понимать, в каком направлении ему следует двигаться и какие действия совершить для удовлетворения своих потребностей.

#### 6.5.3 Сценарий

1) Поставить одну из задач, соответствующую требованиям к приложению.

2) Без посторонней помощи проделать шаги, которые по мнению тестировщика помогут ему в достижении поставленной цели.

3) Убедиться в том, что поставленная цель выполнена (просмотреть таймлайн или перейти к статистике – кнопка “statistics”).

#### 6.5.4 Ожидаемый результат

Имея цель, достижение которой возможно средствами данного ПО, пользователь на интуитивном уровне понял, в каком направлении ему следует двигаться и какие действия совершить для удовлетворения своих потребностей.

### 6.6 Тестовый сценарий использования ресурсов

#### 6.6.1 Идентификатор

Использование ресурсов

#### 6.6.2 Назначение

В ходе исполнения данного сценария требуется удостовериться, что все файлы, используемые при работе с приложением – файлы, в которые ведется запись всей, добавленной пользователем информации должны считываться корректно. Также необходимо удостовериться, что программа не использует достаточно большого кол-ва аппаратных ресурсов.

#### 6.6.3 Сценарий

- 1) Выполнить тестовый сценарий "корректность".
- 2) Открыть в проводнике каталог "files" с данными которыми оперирует программа.
- 3) Открыть файл "subjects.txt".
- 4) В первой колонке – кол-во лаб. работ. Во второй – название предмета. Открыть программу, нажать кнопку "statistics".
- 5) В правой таблице нажать на элемент "выпадающий список"
- 6) Убедиться в соответствии названий предметов и кол-во лаб. работ в файле и в программе.
- 7) Аналогично проверить остальную информацию записанную в файл (см. вспомогательный пункт 8).

8\*) Файл "note.txt" : первая колонка – число заметки  
вторая колонка – месяц заметки  
третья колонка – заметка.

Файл папки "absences": первая колонка – число пропуска  
вторая колонка – месяц пропуска

Файл папки "marks": первая колонка – число дедлайна  
вторая колонка – месяц дедлайна  
третья колонка – число сдачи  
четвертая колонка – месяц сдачи

#### 6.6.4 Ожидаемый результат

Все файлы, используемые при работе с приложением – файлы, в которые ведется запись всей, добавленной пользователем информации считываются корректно. Программа не использует достаточно большого кол-ва аппаратных ресурсов.

### 6.7 Тестовый сценарий мультиплатформенности

#### 6.7.1 Идентификатор

Мультиплатформенность

#### 6.7.2 Назначение

В ходе исполнения данного сценария требуется удостовериться – имеется ли возможность запуска приложения на платформе Windows/UNIX, что необходимо достичь в соответствии с документом требований.

#### 6.7.3 Сценарий

- 1) Записать исходные программы файлы на накопитель или просто запустить скомпилированное приложение на одной из систем (Windows/UNIX).
- 2) Убедиться в работоспособности приложения (пройти сценарий тестирования "корректность").
- 3) Если в п.1 приложение запускалось под Windows, то



запустить приложение под UNIX (иначе под Windows).

4) Повторить п.2.

#### 6.7.4 Ожидаемый результат

Приложение корректно запускается на системах Windows и UNIX.

### 6.8 Тестовый сценарий стабильности

#### 6.8.1 Идентификатор

Стабильность

#### 6.8.2 Назначение

В ходе исполнения данного сценария требуется удостовериться, что не возникает никаких непредвиденных ситуаций и логических ошибок при использовании, разработанного ПО.

#### 6.8.3 Сценарий

- 1) В запущенном приложении нажать кнопку "orginize".
- 2) Перейти к выполнению функции добавления информации путем нажатия кнопки "add...".
- 3) Выбрать один из предложенных видов информации.
- 4) Заполнить предложенные поля.
- 5) Нажать кнопку "apply".
- 6) Повторить п.п. 2-5 для всех видов информации.
- 7) Перейти к редактированию информации (нажать кнопку "edit info").
- 8) Заполнить предложенные поля.
- 9) Нажать кнопку "apply".
- 10) Повторить п.п. 7-9 для всех видов информации.
- 11) Проверить наличие всех добавлений и изменений на таймлайне.
- 12) Нажать кнопку "statistics".
- 13) В случае соответствия информации в таблицах статистики информации добавленной или измененной тестировщиком считать тестовый сценарий выполненным успешным.
- 14) Нажать кнопку "back".
- 15) В случае соответствия информации на таймлайне информации добавленной или измененной тестировщиком считать тестовый сценарий выполненным успешным.

#### 6.8.4 Ожидаемый результат

Непредвиденных ситуаций и логических ошибок в процессе пользования программой не возникло.

### 6.9 Тестовый сценарий сосуществования



### 6.9.1 Идентификатор

Существования

### 6.9.2 Назначение

В ходе исполнения данного сценария требуется удостовериться, что приложение не вступает в конфликты с иным ПО и ни каким образом не оказывает отрицательное влияние на систему.

### 6.9.3 Сценарий

1) Выполнить тестовый сценарий "корректность".

2) В случае не возникновения непредвиденных ошибок системы и конфликтов с другими приложениями считать тестовый сценарий выполненным успешно.

### 6.9.4 Ожидаемый результат

Приложение не вступило в конфликты с иным ПО, и ни каким образом не оказывает отрицательного влияния на систему.

## 7 Заключение

При выполнении тестовых сценариев (п. 6), тестируемый продукт должен полностью подтвердить каждый из аспектов тестирования описанных в пункте 4. При полном соблюдении тестового плана и в случае успешного прохождения каждого из тестов, пользователю гарантируется надежность и полная работоспособность разработанного программного обеспечения.