

## 1. Тестирование текстового поля.

### Functional:

- Обязательность ввода
- Обработка только пробелов
- Использование пробелов в тексте (перед текстом, после него, и внутри; пробелы в начале и в конце строки должны отсекаются при сохранении)
- Минимально/максимально допустимое количество символов (осуществлять проверку на ввод большого текста без пробелов не нужно)
- Формат данных (исходя из его логического назначения и требований приложения)
- Формат числовых данных (если допускаются): негативные, дробные с точкой и запятой, с точкой и запятой 123.123.123,00 )
- Ввод тегов и скриптов (проверка должна осуществляться только в пользовательской части. Введенные теги должны отобразиться в том же виде, в котором они были введены)
- Использование специальных символов (введенные символы должны отобразиться в том же виде, в котором они были введены, если только ввод спецсимволов не запрещен требованиями приложения)
- Возможность редактирования введенных значений
- Корректное распределение текста по строкам (переход на новую строку автоматически)
- Уникальные данные (например, уникальность логина)
- Автоматическая постановка курсора в первое поле для ввода при открытии формы

### GUI:

- Название поля (спеллинг, соответствие с открытым модулем или страницей)
- Выравнивание названий полей (выравнивание по левому краю или правому краю (в зависимости от требований приложения, отступы, идентичность расстояний между названием и полем)
- Корректное расположение текста внутри текста, длинный текст не выходит за границы поля при вводе
- Унификация дизайна (цвет, шрифт, размер (высота/ширина), выравнивание полей)
- Расположение вводимого текста внутри поля (унификация, выравнивание по нижнему краю, если иное не определено конкретными требованиями приложения)

## 2. Тестирование кнопки.

### Functional:

- Отсутствие вызова одного и того же действия повторно при нажатии на кнопку несколько раз
- Недоступные кнопки не скрыты, а заблокированы
- Нажатие на пространство между близко расположенными кнопками не должно приводить к действию

### GUI:

- Название кнопки (спеллинг, соответствие с действием)
- Эффект 'нажатия' (вид кнопки должен изменяться при нажатии, если это не противоречит возможностям браузера)
- Название хинтов (соответствие с названием кнопки, спеллинг)
- Унификация дизайна (цвет, шрифт, размер (высота/ширина), цвет подсветки, выравнивание). Проверка осуществляется как для кнопки, как элемента, так и для названия кнопки

### 3. Тестирование radio buttons.

#### Functional:

- Функциональность (включение/выключение)
- Не может быть меньше 2 радиокнопок
- По умолчанию одна радиокнопка должна быть включена
- Не может быть включено более 1 радиокнопки
- При переходе на следующую страницу и возвращении назад выбранная радио кнопка не должна сбрасываться

#### GUI:

- Унификация дизайна для всего приложения
- Выравнивание расположения радиобаттона с соответствующим названием
- Выравнивание расположений радиобаттонов (по краю)

## 4. Тестирование check boxes.

### Functional:

- Функциональность (включение/выключение)
- Обязательность выбора хотя бы одного чекбокса
- Наличие дополнительного чекбокса, выставляющего/снимающего все чекбоксы при наличии больше 10 чекбоксов
- При переходе на следующую страницу и возвращении назад выбранный чекбокс не должен сбрасываться

### GUI:

- Унификация дизайна для всего приложения
- Выравнивание расположения чек бокса с соответствующим названием
- Корректность отображения задизэйбленного чек бокса

## 5. Тестирование полей со списком.

### Functional:

- Сортировка по алфавиту или по смыслу
- В случае, если значения выходят за границы списка, и нет возможности увеличения размера списка, то необходимо отображение хинтов (всплывающих подсказок)
- Выбор пункта списка по нажатию соответствующей первой буквы на клавиатуре
- Возможность выбора нескольких значений для поля со списком
- Возможность введения значений вручную (если это позволяет приложение)

### GUI:

- Спеллинг значений
- Подсветка при выборе каждого из значений, при выборе нескольких значений одновременно
- Унификация дизайна (цвет, шрифт, размер (высота/ширина), цвет подсветки, выравнивание). Проверка осуществляется как для поля, как элемента, так и для значений, и их названий

## 6. Тестирование меню.

### **Functional:**

- Осуществление соответствующего перехода при выборе пункта меню
- Визуальное различие в момент работы на определенной вкладке (подсветка, подчеркивание)

### **GUI:**

- Подсветка таба при наведении курсора
- Изменение курсора при наведении
- Эффект “нажатия”, если это не противоречит возможностям браузера
- Если работа в данный момент в выбранной вкладке, то в меню она отличается визуально (подсвечивается, подчеркивается)
- Совпадение названий в случае, если меню дублируется в нескольких местах

## 7. Тестирование окон.

### **Functional:**

- Возможность изменения окна браузера

### **GUI:**

- Появление скrolла при уменьшении (изменении) размера окна браузера
- Сохранение расположения элементов при уменьшении (изменении) окна браузера, при изменении масштаба
- Соответствие названия окна в зависимости от назначения страницы (например, название окна должно быть Profile, если пользователь находится на странице профиля)
- Спеллинг, синтаксис названий
- Унификация названий

## 8. Тестирование скроллинга.

### **Functional:**

- Отсутствие скrolла в случае, если текст помещается на странице без прокрутки.
- Соответствующее изменения текста при использовании скrolла.
- Возможность изменения положения скrolла при помощи мыши, кнопок Page up/down, Home/End.

### **GUI:**

- Унификация видов и типов скроллов на всех страницах (если есть кастомный скролл, он должен быть применен на всех идентичных формах)



## 9. Тестирование ссылок.

### Functional:

- Функционирование ссылки (должен осуществиться переход на соответствующую страницу)
- При наведении указателя мыши отображается подсказка (желательно)
- Форматы ссылок и префиксов

### GUI:

- Унификация стилей (в соответствие с дизайном сайта)
- Расположение ссылок (в соответствие с дизайном сайта). Например, расположение всех ссылок слева или справа от элементов
- Названия (унификация, идентичность названий ссылок одинакового назначения, спеллинг, соответствие с открытым модулем или страницей, вместимость названия ссылки в отведенном блоке)
- Изменение вида курсора при наведении на ссылку
- Изменение вида ссылки при наведении курсора (подчеркивание)

## 10. Тестирование таблиц.

### Functional:

- При появлении нескольких страниц есть кнопки Вперед, Назад, На первую, На последнюю страницу (пагинация)
- Проверка сортировок (+ проверка сортировки по дефолту)
- Проверка фильтрации (если есть возможность)
- Апдейт значений таблицы после добавления/изменения/удаления данных
- Единичное/множественное выделение нескольких значений

### GUI:

- Унификация дизайна для всего приложения (цвет, шрифт, размер (высота/ширина), выравнивание)
- Название (соответствие с текущим модулем, спеллинг)
- Выравнивание иконок сортировки в названии колонок
- Выравнивание названий колонок, значений внутри таблицы
- Корректное отображение длинных названий (соответствующие переходы на новые строки, сокращение названий (появление ..., либо сокращение по слову)
- Корректное отображение данных после использования сортировки (размеры колонок и столбцов фиксированы, текст не разбивает структуру таблицы)

## 11. Тестирование поп-апов.

### Functional:

- Должен быть модальным
- Корректное выделение бэкграунда страницы (если фон страницы изменяется при появлении поп апа, то при изменении масштаба страницы фон должен заполнять всю страницу - размер измененного фона соответствует размеру страницы)
- Фиксированное положение поп-апа (динамическое изменение положения) в случае использования скролла

### GUI:

- Спеллинг, синтаксис текста, расположенного на поп-апе
- Отображение поп-апа по центру страницы, окна, формы
- Выравнивание текста, представленного на поп-апе
- Корректное расположение текста на поп-апе: текст должен быть в рамках поп-апа, длинное название должно располагаться на новой строке, если иное не определено конкретными требованиями приложения)

## 12. Тестирование календарей.

### **Functional:**

- При возможности ввода даты вручную необходимо проверить разные форматы
- Проверки логичности ввода (даты в будущем, и т.д)
- Проверка високосного года

### **GUI:**

- Унификация дизайна для всего приложения (цвет, шрифт, размер (высота/ширина), выравнивание)
- Отображение календаря рядом с полем
- Корректное выравнивание всех элементов и ссылок в календаре

## 13. Тестирование полей для загрузки файлов.

### Functional:

- Обязательность выбора файла
- Форматы
- Ограничения на размер
- При отсутствии изображений должен быть соответствующий thumbnail, либо картинка совсем не должна отображаться
- Контроль за размером (высота/ширина), должен быть ресайз
- Загрузка исполняемых файлов (EXE, PHP, JSP etc.).
- Переименованный EXE

### GUI:

- Унификация дизайна для всего приложения (цвет, шрифт, размер (высота/ширина), выравнивание)
- Выравнивание названий загруженных файлов, самих thumbnails файлов

## 14. Тестирование сообщений.

### Functional:

- Пользователь должен быть информирован о действиях, происходящих в системе посредством сообщений об успешном завершении операции
- На необратимые действия, такие как удаление, должны быть подтверждающие сообщения

### GUI:

- Сообщение о том, что нет соответствующих айтемов (в таблицах, при поиске, при переходе на страницы)
- Спеллинг, синтаксис сообщений
- Соответствие сообщений по смыслу в зависимости от выполняемого действия
- Соответствие названия требуемого действия в сообщении об ошибке действию, которое пользователь должен выполнить. Например, если необходимо выбрать значение из списка, в сообщении об ошибке должно быть указано 'Please select', а не 'Please enter'
- Унификация стилей (цвет, размер) для всего приложения
- Соответствие цветов типу сообщений (красный для сообщений об ошибках, зеленый для сообщений об успешном завершении операции), если данные цвета не противоречат специфичным требованиям приложения
- Соответствие названий полей в сообщениях об ошибках и сообщениях об успешном завершении операции названиям полей, форм, таблиц, кнопок, и т.д.
- Соответствие порядка вывода сообщений об ошибках в соответствии с порядком расположения полей, в которых были найдены ошибки
- Поле, в котором содержится ошибка должно (желательно) выделяться цветом

## 15. Общие проверки.

- Пользователь должен быть информирован о действиях, происходящих в системе посредством сообщений об успешном завершении операции
- На необратимые действия, такие как удаление, должны быть подтверждающие сообщения
- 404 Error (переход по некорректному урлу должен вести на страницу с 404 ошибкой, а не просто на страницу Page cannot be found. страница 404 ошибки должна быть реализована в общем дизайне тестируемого приложения)
- Tab order (сверху вниз слева направо). Поля, доступные для прочтения и задизейбленные должны пропускаться
- Logo должен быть ссылкой на главную страницу
- Фокус на кнопке для исполнения действий (ввод данных -> нажатие Enter -> действие осуществилось)
- Проверка Breadcrumbs ("Хлебные крошки" - элемент навигации, являющийся признаком удобства пользования сайтом в целом и перемещением по его структуре)
- Уменьшение/увеличение масштаба страницы (элементы должны соответственно перераспределиться с сохранением пропорций)
- Отображение flash-элементов при отключенном/неустановленном в браузере flash-плеере (пользователю должно быть предложено скачать и установить последнюю версию flash-плеера; на месте flash-объекта должно отображаться альтернативное изображение)
- Проверка работоспособности: отправки email, уведомлений (как админу, так и пользователю), если только отсутствие писем не является спецификой проекта