Задание подготовлено в рамках проекта АНО «Лаборатория модернизации образовательных ресурсов» «Кадровый и учебно-методический ресурс формирования общих компетенций обучающихся по программам СПО», который реализуется с использованием гранта Президента Российской Федерации на развитие гражданского общества, предоставленного Фондом президентских грантов.

**Разработчик**

Плюснина Елена Викторовна, ГБПОУ «Тольяттинский социально-экономический колледж»

**Назначение задания**

Целеполагание и планирование. Уровень II

МДК.05.03 Тестирование информационных систем

Тема: Отладка и тестирование информационных систем

**Комментарии**

Задание предлагается на этапе освоения нового материала, когда обучающиеся еще не знакомы с форматом теста для проведения ручного тестирования. Требования к формату вводятся по итогам обсуждения результатов выполнения задания.

Если преподаватель планирует облегчить работу обучающихся, он может предложить бланк с уже отчеркнутыми в соответствии с верным количеством шагов строками, а также интегрировать в источник 2 пример, который показан после инструмента проверки.

Вы работаете в маленькой ИТ-фирме, где выполняете самые разные функции. Сейчас перед вами поставили задачу протестировать программный продукт, который разработали ваши коллеги.

Изучите ту часть функциональных требований, прописанных в техническом задании, на предмет соответствия которым следует протестировать продукт (источник 1) и описание предметной области (источник 2). Ознакомьтесь с порядком функционального тестирования пользовательского интерфейса (источник 3).

**Напишите Test Case системы.**

Заполните бланк. (*Отчеркивайте каждый шаг в новую строку*).

*Бланк*

| № | Шаги тестирования | Ожидаемый результат |
| --- | --- | --- |
|  |  |  |

***Источник 1***

**Техническое задание (фрагмент)**

Основными требованиями к системе являются:

* возможность ввода и идентификации персональных данных (номера амбулаторной карты и пароля);
* корректное отображение пунктов меню, возможности перехода по пунктам;
* запись на прием (и ее занесение в соответствующую базу данных или хранилище);
* проверка результата и получение напечатанного талона;
* запрет на выполнение следующей записи при наличии трех действительных на данное число.

***Источник 2***

**Описание предметной области**

Информационная система записи на прием к врачу через инфомат. Система должна иметь высокий коэффициент готовности, скорость и надежность, а также, поскольку будет доступна большому числу пользователей, иметь интуитивно понятный интерфейс.

Данное приложение имеет клиент-серверную структуру. На клиентской части через браузер пользователь выбирает специалиста, фамилию и, если возможно, время приема. Вся обработка информации идет на стороне сервера. В качестве ограничений от спамерских атак предусмотрена необходимость ввода номера амбулаторной карты и пароля, выдаваемого конкретному пациенту, и возможность записи не более чем к трем специалистам в один день.

На рисунке 1 представлен интерфейс входа в систему. Пользователь может произвести запись, только введя номер амбулаторной карты и пароля. Очевидно, что в период простоя на инфомате может отображаться заставка, которая при начале работы убирается с экрана и появляется стартовая страница. При тестировании необходимо убедиться, что отображаемая страница соответствует странице, описанной в спецификации.

Следует заметить, что экран инфомата является сенсорным, поэтому поля ввода переключаются прямым нажатием на соответствующее поле.



Рисунок 1 - Интерфейс входа в систему ному состоянию

На рисунке 2 представлена страница записи к специалисту.



Рисунок 2 - Страница записи к специалистам

***Источник 3***

**Функциональное** **тестирование пользовательского интерфейса**

Ручное *тестирование пользовательского интерфейса* проводится тестировщиком-оператором, который руководствуется в своей работе описанием тестовых примеров в виде набора сценариев. Каждый сценарий включает в себя перечисление последовательности действий, которые должен выполнить оператор, и описание важных для анализа результатов тестирования ответных реакций системы, отражаемых в пользовательском интерфейсе.

Типичная форма записи сценария для проведения ручного тестирования - таблица, в которой в одной колонке описаны действия (шаги сценария), в другой - ожидаемая реакция системы, а третья предназначена для записи того, совпала ли ожидаемая реакция системы с реальной и перечисления несовпадений (таблица 1).

Таблица 1 - Пример сценария для ручного *тестирования пользовательского интерфейса*

|  |  |  |  |
| --- | --- | --- | --- |
| № | Действие | Реакция системы | Результат |
| 1 | Щелкните на пиктограмме System и выберите пункт меню 'System Management Applet'. | Появится окно ввода логина и пароля | Верно |
| 2 | Введите в появившееся окно ввода имя пользователя 'guest1' и пароль 'guest'. Затем нажмите кнопку 'Login'. | Появится окно 'System Management Applet'. В верхнем правом углу должно быть выведено имя вошедшего пользователя guest1Все опции в окне должны быть отключены (выведены серым цветом) | НеверноОкно имеет название 'System *Management Application*' |
| 3 | Завершите сеанс работы с апплетом щелчком по пиктограмме 'Logout'. | Окно 'System Management Applet' должно быть закрыто | Верно |

**Тест-кейсы** для проверки пользовательского интерфейса, как правило, представляют собой сценарии, описывающие действия пользователя при работе с системой. Сценарии могут быть записаны либо на естественном языке, либо на формальном языке какой-либо системы автоматизации пользовательского интерфейса. Выполнение тестов при этом производится либо оператором в ручном режиме, либо системой, которая эмулирует поведение оператора.

**Требования к пользовательскому интерфейсу могут быть разбиты на две группы:**

* требования к внешнему виду пользовательского интерфейса и формам взаимодействия с пользователем;
* требования по доступу к внутренней функциональности системы при помощи пользовательского интерфейса.

Другими словами, первая группа требований описывает взаимодействие подсистемы интерфейса с пользователем, а вторая - с внутренней логикой системы.

К первой группе можно отнести следующие типы требований.

**Требования к содержанию и оформлению выводимых сообщений**

Требования к содержанию и оформлению выводимых сообщений определяют текст сообщений, выводимых системой, его шрифтовое и цветовое оформление. Также часто в таких требованиях определяется, в каких случаях выводится то или иное сообщение.

Так, например, для тестирования требования

*Сообщение "Невозможно открыть файл" должно выводиться в статусную строку прижатым к левому краю, красным цветом, полужирным шрифтом в случае недоступности открываемого файла по чтению*.

необходимо проверить, что при возникновении указанной ситуации сообщение действительно выводится согласно требованиям.

Однако в случае тестирования требования вида

*Сообщения об ошибках должны выводиться в статусную строку прижатыми к левому краю красным цветом полужирным шрифтом*.

необходимо проверять форматы всех возможных сообщений об ошибках программы во всех возможных ошибочных ситуациях. Таким образом, очевидно, что при тестировании пользовательского интерфейса не всегда можно однозначно определить количество тестовых примеров, которые понадобятся для тестирования требования. Эта проблема вызвана тем, что требования к пользовательскому интерфейсу зачастую кажутся слишком очевидными для их точной формулировки. Эта неконкретность требований и вызывает большое количество тестов для каждого требования.

**Требования к форматам ввода**

Данная группа требований определяет, в каком виде информация поступает от пользователя в систему. При этом кроме собственно требований, определяющих корректный формат, к этой группе относятся требования, определяющие реакцию системы на некорректный ввод. Для проверки таких требований необходимо проверять как корректный ввод, так и некорректный. Желательно при этом разбивать различные варианты ввода на классы эквивалентности (как минимум на два - корректные и некорректные).

Ко второй группе относятся следующие типы требований.

**Требования к реакции системы на ввод пользователя**

Данный тип требований определяет связь внутренней логики системы и интерфейсных элементов. Например,

*При нажатии кнопки "Сброс" значение таймера синхронизации передачи должно сбрасываться в 0*.

Для проверки такого требования в тестовом примере должно быть сымитировано нажатие на кнопку "Сброс", после чего должна проводиться проверка значения таймера. Однако некоторые требования определяют в качестве реакции системы не то, как меняется ее внутреннее состояние, а реакцию пользовательского интерфейса. Например, в требовании

*При нажатии кнопки "Отложенный сброс" должно выводиться окно "Ввод значения времени для отложенного сброса"*.

в качестве реакции на использование одного интерфейсного элемента определяется появление другого интерфейсного элемента. Такие требования проверяются при помощи имитации ввода пользователя и анализа появляющихся интерфейсных элементов.

**Шаги тест-кейса (steps)** описывают последовательность действий, которые необходимо реализовать в процессе выполнения тест-кейса.

**Общие рекомендации по написанию шагов таковы:**

1. Начинайте с понятного и очевидного места, не пишите лишних начальных шагов (запуск приложения, очевидные операции с интерфейсом и т.п.).
2. Даже если в тест-кейсе всего один шаг, нумеруйте его (иначе возрастает вероятность в будущем случайно «приклеить» описание этого шага к новому тексту).
3. Если вы пишете на русском языке, то используйте безличную форму (например, «открыть», «ввести», «добавить» вместо «откройте», «введите», «добавьте»), в английском языке не надо использовать частицу «to» (т.е. «запустить приложение» будет «start application», не «to start application»).
4. Соотносите степень детализации шагов и их параметров с целью тест-кейса, его сложностью, уровнем и т.д. В зависимости от этих и многих других факторов степень детализации может варьироваться от общих идей до предельно чётко прописанных значений и указаний.
5. Ссылайтесь на предыдущие шаги и их диапазоны для сокращения объёма текста (например, «повторить шаги 3–5 со значением…»).
6. Пишите шаги последовательно, без условных конструкций вида «если… то…».

**Ожидаемые результаты (expected results)** по каждому шагу тест-кейса описывают реакцию приложения на действия, описанные в поле «шаги тест-кейса». Номер шага соответствует номеру результата.

По написанию ожидаемых результатов можно порекомендовать следующее:

* Описывайте поведение системы так, чтобы исключить субъективное толкование (например, «приложение работает верно» - плохо, «появляется окно с надписью…» - хорошо).
* Пишите ожидаемый результат по всем шагам без исключения, если у вас есть хоть малейшие сомнения в том, что результат некоего шага будет совершенно тривиальным и очевидным (если вы всё же пропускаете ожидаемый результат для какого-то тривиального действия, лучше оставить в списке ожидаемых результатов пустую строку - это облегчает восприятие).
* Пишите кратко, но не в ущерб информативности.
* Избегайте условных конструкций вида «если… то…».

*Использованы материалы источников:*

[*https://studref.com/388374/informatika/primer\_testirovaniya\_programmnogo\_obespecheniya\_prilozheniya*](https://studref.com/388374/informatika/primer_testirovaniya_programmnogo_obespecheniya_prilozheniya)

[*https://sergeygavaga.gitbooks.io/kurs-lektsii-testirovanie-programnogo-obespecheni/content/lektsiya-4-ch3.html*](https://sergeygavaga.gitbooks.io/kurs-lektsii-testirovanie-programnogo-obespecheni/content/lektsiya-4-ch3.html)

[*https://intuit.ru/studies/courses/1040/209/lecture/5418*](https://intuit.ru/studies/courses/1040/209/lecture/5418)

Инструмент проверки

| № | Шаги тестирования | Ожидаемый результат |
| --- | --- | --- |
| 1 | Работоспособность виртуальной клавиатуры, т. е. при нажатии на единицу в поле реально отображается единица (и так далее) | Возможность ввода номера амбулаторной карты и пароля |
| 2 | Переход между полями ввода | Возможность перехода между полями ввода |
| 3 | В поле пароль ввести неверные данные | Проверка корректности срабатывания системы и выдачи сообщения об ошибке в случае ввода пароля, не соответствующего карте |
| 4 | Вход в систему в качестве пользователя | Проверяет, что система идентифицировала пользователя верно, т. е. сверху появились номер его карты, его фамилия, имя, отчество и дата рождения  |
| 5 | Работоспособность кнопок «к специалисту» и «на обследование» | Переход на соответствующую страницу |
| 6 | На странице записи к специалистам: возможность перехода на страницу выбранного специалиста | Переход на страницу выбранного специалиста |
| 7 | Работают кнопки прокрутки | Переход выше/ ниже |
| 8 | Кнопка возврата на предыдущий шаг работоспособна | Переход на предыдущий шаг |
| 9 | Самая верхняя левая кнопка работает на выход из режима записи на прием | Выход из режима записи на прием |

*Подсчет баллов*

|  |  |
| --- | --- |
| За каждый шаг, спланированный верно и в верной последовательности | 1 балл |
| *Максимально* | *9 баллов* |
| За каждый результат, соответствующий верно спланированному шагу | 1 балл |
| *Максимально* | *9 баллов* |
| Отсутствуют избыточные шаги (балл ставится при полном наборе верных шагов)  | 1 балл |
| ***Максимальный балл*** | ***19 баллов*** |

*Пример для интеграции в источник 2 для облегчения понимания обучающимися*

Одним из вариантов пользовательского сценария может быть, например, такой: пациент Иванов Иван Иванович хочет записаться к определенному врачу-специалисту на определенное время и дату. Он вводит номер карты и пароль, переходит на страницу специалистов, проверяет, что система его идентифицировала верно, т. е. сверху появились номер его карты, его фамилия, имя, отчество и дата рождения. Далее он переходит на страницу врачей, выбирает требуемого, переходит на страницу расписания работы врача, выбирает дату и, если есть время, записывается, если нет, то выходит из системы самозаписи.