**5 СИСТЕМНОЕ ТЕСТИРОВАНИЕ**

Системное тестирование - это тестирование программного обеспечения, выполняемое на полной, интегрированной системе, с целью проверки соответствия системы исходным требованиям [17]. Системное тестирование относится к методам тестирования чёрного ящика, и, тем самым, не требует знаний о внутреннем устройстве системы. Тестирование бывает:

* функциональное тестирование;
* тестирование пользовательского интерфейса;
* тестирование совместимости;
* тестирование безопасности;
* тестирование производительности;
* автоматическое тестирование;
* интеграционное тестирование.

Испытания системы имеют две основные цели:

* показать разработчику и клиенту, что программное обеспечение отвечает заявленным требованиям. С точки зрения клиента это означает, что для каждой функциональности, желаемой с его стороны и записанной в документе требований, проведен как минимум в один тест (как правило, конечно, больше). В случае общедоступного программного обеспечения то, что в программном обеспечении протестированы все заданные основные свойства. Соответствующий данной цели тест называют вариацией (проверкой достоверности). Успешное вариация указывает, что система работает как надо;
* найти ситуации, когда программное обеспечение ведет себя ошибочно, нежелательно или не соответствует спецификации. Следовательно, поиск ошибок в этом смысле предназначен для того, чтобы ликвидировать нежелательное поведение системы, как, например, крах системы, нежелательное взаимодействие с другими системами, неправильные расчеты, поврежденные данные. Выполняющее эту задачу тестирование именуется тестированием дефектов (defect testing). Здесь являющийся успешным тест показывает действие ошибки системы, или, другими словами, находит в системе ошибку (к дальнейшему исправлению которой, и приступают).

**5.1** **Функциональное тестирование**

Функциональное тестирование - это тестирование программного обеспечения в целях проверки реализуемости функциональных требований, то есть способности программного обеспечения в определённых условиях решать задачи, нужные пользователям [18]. Функциональные требования определяют, что именно делает программное обеспечение, какие задачи оно решает.

Функциональные требования включают в себя:

* функциональная пригодность;
* точность;
* способность к взаимодействию;
* соответствие стандартам и правилам;
* защищённость.

Тестирование и отладка программы являются наиважнейшими этапами разработки любых программных продуктов. Цель этого этапа - проверка правильности и точности реализации функций, выполнение которых возлагается на данный программный продукт. В случае выявления некоторых неточностей и ошибок необходимо проведение работ по их исправлению и доработке программного продукта до требуемого уровня.

На основе функций, которые должны быть протестированы в разработанной информационной обучающей системе, составлен чек-лист (checklist), приведенный в таблице 5.1.

Таблица 5.1 - Чек-лист

|  |  |  |  |
| --- | --- | --- | --- |
| Тестируемый модуль | | Тестируемая функция | Результат |
| Вход в систему | | Вход в систему | Выполнено успешно |
| Выход из системы | | Выход из системы | Выполнено успешно |
| Просмотр и поиск сотрудников | | Отображение информации и фильтрация | Выполнено успешно |
| Просмотр информации выбранного сотрудника | Отображение информации других таблиц на основе выбранного значения | | Выполнено успешно |
| Работа со статусами | Добавление, редактирование и удаление данных с выводом информации подчиненной таблицы | | Выполнено успешно |
| Работа с подразделениями | Добавление, редактирование и удаление данных с выводом информации подчиненной таблицы | | Выполнено успешно |
| Работа с должностями | Добавление, редактирование и удаление данных с выводом информации подчиненной таблицы | | Выполнено успешно |
| Работа с иностранными языками | Добавление, редактирование и удаление данных с выводом информации подчиненной таблицы | | Выполнено успешно |
| Работа с планирование повышения | Добавление, редактирование и удаление данных | | Выполнено успешно |

В качестве тестирования программного продукта был выбран тест-кейс (**Test Case**).

**Test Case** - это артефакт, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части.

Тестирование производилось на ОС «Windows 10».

**5.2** **Оценка безопасности**

Обеспечение безопасности информационных систем представляет собой ряд мероприятий, направленных на предотвращение несанкционированных воздействий на защищаемую информацию, а также её утечки.

Windows Forms содержит модель безопасности, основанную на коде (уровни безопасности задаются для кода, независимо от пользователя, запустившего код). Это дополнение к схемам безопасности, которые могут быть уже установлены на компьютере. К ним могут относиться данные в браузере (например, безопасность на основе зоны, доступная в Internet Explorer) или операционная система (например, безопасность на основе учетных данных Windows NT) [19].

До выпуска .NET Framework весь код, выполняемый на компьютере пользователя, имел те же права или разрешения на доступ к ресурсам, которые имел пользователь компьютера. Например, если пользователю был разрешен доступ к файловой системе, то код также получал доступ к файловой системе Если пользователю был разрешен доступ к базе данных, то и коду предоставлялся доступ к ней. Наличие таких прав или разрешений может быть допустимо для кода в исполняемых файлах, которые пользователь собственноручно установил на локальном компьютере, но оно неприемлемо в тех случаях, когда речь идет о потенциально вредоносном коде, поступившем из Интернета или локальной интрасети. Такой код не должен получать доступ к ресурсам компьютера пользователя без его разрешения.

В .NET Framework введена инфраструктура, называемая системой управления доступом для кода, которая позволяет отличать разрешения (или права) от прав, которые имеет пользователь. По умолчанию код, поступающий из Интернета или интрасети, может выполняться только в режиме частичного доверия. В режиме частичного доверия приложение подвергается ряду ограничений: помимо всего прочего, приложению отказано в доступе к локальным жестким дискам, и оно не может запускать неуправляемый код. .NET Framework управляет ресурсами, к которым коду разрешен доступ в зависимости от идентификатора этого кода: от того, откуда он поступил, имеет ли он сборки со строгими именами, подписаны ли они сертификатом и т. д.

Технология ClickOnce, используемая для развертывания Windows Forms приложений, помогает упростить разработку приложений, работающих в режиме частичного доверия, с полным доверием или с частичным доверием с повышенными разрешениями. Технология ClickOnce предоставляет такие функции, как повышение уровня разрешений и развертывание доверенных приложений, поэтому приложение может запрашивать полное доверие или повысить уровень разрешений от локального пользователя в ответственном способе.

**5.3 Тестирование производительности**

Тестирование производительности **-** это тестирование, которое проводится для определения скорости работы системы или её части при заданной нагрузке. Тестирование производительности стремится учесть производительность уже на стадии проектирования и моделирования и системы, до начала основной стадии разработки.

Тестирование производительности служит таким типичным целям:

* для демонстрации того, что система удовлетворяет критериям производительности;
* для определения производительности, какой из двух или нескольких систем лучше;
* для определения, какой элемент нагрузки или часть системы приводит к снижению производительности.

Приложение запускается менее чем за 10 секунд, используя для своих нужд 31 МБ памяти. На пике загрузки использует 7% ЦП.

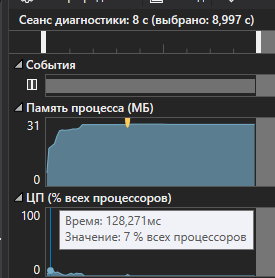


Рисунок 5.1 – Тестирование

Поиск сотрудника не добавляет количество используемой памяти. При добавление новой записи в таблицу используемая память увеличилась на 1 МБ.