

УДК 004.031.042

**ПРОЕКТИРОВАНИЕ ФУНКЦИОНАЛЬНОЙ ЧАСТИ
ВЕБ-ПРИЛОЖЕНИЯ «АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА
ВЕДЕНИЯ ОТЧЕТНОСТИ ДЛЯ ОАО «НАФТАН»»**

А.Н. РОДЧЕНКОВ

(Представлено: канд. техн. наук, доц. А.Ф. ОСЬКИН)

Представлен объект разработки – веб-приложение «Автоматизированная информационная система ведения отчетности для ОАО «Нафтан»». Рассмотрены вопросы проектирования функциональной части веб-приложения «Автоматизированная информационная система ведения отчетности для ОАО «Нафтан»».

Создание программного обеспечения для надежного хранения данных о функционировании предприятия и формирование внутренних отчетов является целью данного исследования. Основная цель разработки приложения – сделать хранение данных о функционировании предприятия и формирование отчетов как можно больше простым и универсальным.

Первым этапом проектирования программного обеспечения является *определение функциональных и других возможностей, которым программное обеспечение должно соответствовать*. Предполагается, что с данной информационной системой будут работать сотрудники установки, и они должны иметь возможность решать с ее помощью следующие задачи:

- ввод и редактирование данных о технологических позициях, приборах и креплениях в удобной форме;
- просмотр данных о технологических позициях, приборах и креплениях в отсортированном виде;
- просмотр и подготовка к выводу списков технологических позиций и закрепленных за ними приборов;
- формирование отчетов согласно стандартам технического регламента;
- экспорт информации и сформированных отчетов из базы данных в Excel и наоборот;
- организация надежного хранения данных;
- организация поиска по ID или части ID.

Таким образом, при разработке продукта были выдвинуты следующие требования к функциональным характеристикам:

- предоставлять пользователю информацию, хранящуюся в информационной системе;
- предоставлять базовый функционал сортировки информации при работе с информационной системой;
- предоставлять пользователю возможность вносить изменения в информацию, хранящуюся в информационной системе;
- предоставлять пользователю возможность экспорта информации из базы данных в Excel и наоборот;
- предоставлять авторизованному пользователю возможность формирования отчетов согласно внутреннему техническому регламенту.

Так как разрабатываемое программное средство является автоматизированной информационной системой, то требования, предъявляемые к любой автоматизированной информационной системе, будут распространены и на разрабатываемый программный продукт. Среди них выделяют:

1) наличие визуальной среды для просмотра и обработки информации. Для обработки информации, которая поступает в информационную систему, необходимо разработать удобную оболочку, через которую, посылая запросы к модулям логики работы системы, пользователь осуществляет необходимую ему обработку данных. Для такой визуальной среды важными характеристиками являются дизайн пользовательского интерфейса, интуитивность и удобство навигации;

2) возможность выполнять с информацией, поступающей в информационную систему, различные стандартные операции. К таким операциям относят операции добавления, удаления, а также изменения (редактирования) данных;

3) логическое разделение автоматизированной информационной системы на программные компоненты. В данном случае информационную систему удобнее всего разделить на четыре основных вышечисленных компонента: база данных, подсистема генерации отчетов, подсистема импорта данных из Excel, редактор базы данных;

4) ориентированность на пользователя – для работы с разрабатываемым программным средством не требуется предварительной подготовки персонала;

5) стабильность работы – приложение должно обеспечивать возможность корректного завершения в случае ошибки.

Следующим этапом проектирования является *разработка функциональной структуры программного обеспечения*. В данном случае информационную систему удобнее всего разделить на четыре основных компонента: база данных, подсистема генерации отчетов, подсистема импорта данных из Excel, редактор базы данных.

Среди функций разрабатываемых подсистем можно выделить следующее:

- 1 Для подсистемы база данных:
 - удобное и надежное хранение необходимой информации;
 - обеспечение целостности хранимой информации.
- 2 Для подсистемы импорта данных из Excel:
 - непосредственно импорт данных из Excel в базу данных;
 - возможность дополнения или обновления имеющийся базы с помощью импорта данных из Excel;
 - возможность выбора заполняемой таблицы;
 - возможность выбора файла для импорта данных из Excel.
- 3 Для подсистемы генерации отчетов:
 - генерация отчетов различных типов;
 - возможность просмотра генерируемых отчетов;
 - изменение параметров генерации отчетов;
 - возможность сохранения отчетов в различных форматах.
- 4 Для подсистемы редактор базы данных:
 - добавление записей в базу данных согласно формату;
 - удаление записей из базы данных;
 - изменение записей в базе данных по ключу;
 - различные варианты группировок при работе с базой данных для более удобной обработки информации;
 - возможность ускоренного поиска по части слова в выбранной ячейке;
 - возможность скрыть меню, расширение рабочей области;
 - ограничение количества выводимых полей на одной странице.

Следующим этапом проектирования программного обеспечения является *создание диаграммы вариантов использования* [1].

При создании диаграмм вариантов использования нужно выделить два вида сущностей – актеры и прецеденты. Актер – это множество логически связанных ролей, исполняемых при взаимодействии с прецедентами. Прецедент – это описание множества последовательных событий, включая варианты, выполняемые системой, которые приводят к наблюдаемому актером результату.

На диаграмме вариантов использования указан один актер – «Сотрудник», который совершает все действия, выделенные при проектировании. Диаграмма вариантов использования представлена на рисунке.

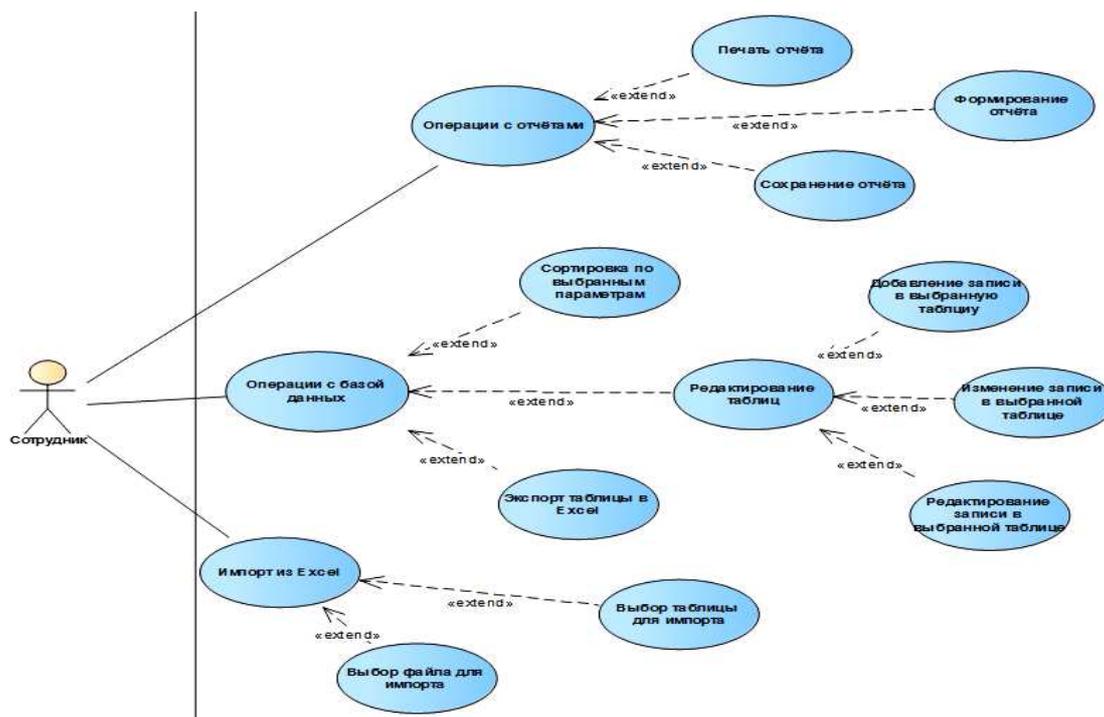


Диаграмма вариантов использования

Последний этап проектирования функциональной части программного обеспечения – *выбор инструментов*.

Выбор средств решения поставленной задачи исходил из того, какое программное и аппаратное обеспечение используется в данный момент на предприятии.

Выбор СУБД – сложная задача, должен основываться, в первую очередь, на потребностях с точки зрения информационной системы и пользователей. Определяющими здесь являются вид программного продукта и категория пользователей. Другими показателями, влияющими на выбор СУБД, служат [2]:

- удобство и простота использования;
- качество средств разработки, защиты и контроля базы данных;
- уровень коммуникационных средств в случае применения ее в сетях;
- фирма-разработчик;
- стоимость.

Система MS SQL Server 2014 позволяет обращаться к данным из любого приложения, разработанного с применением технологий Microsoft .NET и Visual Studio. MS SQL Server обеспечивает высочайший уровень безопасности, надежности и масштабируемости для критически важных приложений. Чтобы использовать новые возможности, постоянно возникающие в быстро меняющемся деловом мире, предприятиям нужно быть способными быстро создавать и развертывать решения, управляемые данными. MS SQL Server 2014 позволяет сократить затраты времени и средств, требуемые на управление и развертывание таких приложений. Также следует учесть, что фирма-разработчик данной СУБД является также разработчиком самой распространенной ОС. В финансовом плане важным фактором является то, что существуют бесплатные сборки данной СУБД (Express).

Для реализации приложения была выбрана среда разработки Microsoft Visual Studio 2013, в качестве языка программирования – C#.

Достоинства платформы .NET [3]:

1 Вся платформа .NET основана на единой объектно-ориентированной модели. Все сервисы, интерфейсы и объекты, которые платформа предоставляет разработчику, объединены в единую иерархию классов. Другими словами, все, что может потребоваться при создании приложений под платформу .NET будет всегда у вас под рукой. Причем все это сгруппировано удобно и интуитивно понятно.

2 Приложение, написанное на любом .NET-совместимом языке, является межплатформенным (в идеале). Почему в идеале? Дело в том, что приложение, написанное, скажем, на том же C#, не зависит от платформы, на которой будет выполняться, но зато зависит от наличия платформы .NET.

3 В состав платформы .NET входит «сборщик мусора», который освобождает ресурсы. Таким образом, приложения защищены от утечки памяти и от необходимости освобождать ресурсы. Это делает программирование более легким и более безопасным.

4 Приложения .NET используют безопасные типы, что повышает их надежность и совместимость.

5 NET приложения могут быть сертифицированы на безопасность. Это является особенностью промежуточного кода, в который преобразуются все .NET приложения.

6 Абсолютно все ошибки обрабатываются механизмом исключительных ситуаций. Это позволяет избежать разногласий, которые иногда возникают при программировании под Win32.

7 Повторное использование кода стало еще удобнее. Это связано с тем, что промежуточный язык MSIL не зависит от языка программирования. Например, вы можете написать программу на C#, а патч к ней писать уже, скажем, на J#.

Заключение

В результате проведенной работы спроектировано веб-приложение, позволяющее хранить, изменять, добавлять информацию в базу данных, осуществлять быстрый поиск и различную группировку информации с последующим формированием различных отчетов.

ЛИТЕРАТУРА

1. Лекция 3: Элементы графической нотации диаграммы вариантов использования [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/studies/courses/32/32/lecture/1004?page=2>. – Дата доступа: 27.09.2017.
2. Выбор СУБД [Электронный ресурс]. – Режим доступа <http://websideworld.com/main/data-storage/6/2>. – Дата доступа: 27.09.2017.
3. Нгуен, П. Достоинства и недостатки .Net / Павел Нгуен [Электронный ресурс]. – Режим доступа: http://blog.nguen.net/post33-good_bad_plus_minus_dot_net.html. – Дата доступа: 27.09.2017.