Мантуленко А.В., Шохин А.О
Проектирование и разработка информационной системы на основе современных технологий // Вестник Самарского университета. Экономика и управление. 2016. № 2. С. 142–148.

УДК 330.4

А.В. Мантуленко, А.О. Шохин*

ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ НА ОСНОВЕ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ

В статье рассматривается применение современных программных технологий, и описываются особенности и влияние от их использования при разработке информационных систем, соответствующих текущим требованиям надежности, безопасности и производительности.

Ключевые слова: информационная система, база данных MySQL, современные технологии, Microsoft Azure, криптографический алгоритм RSA, хеширование SHA-2.

В соответствии с темой приоритетной задачей в данной работе является изучение комплекса современных технологий, которые могут быть эффективно применены для создания информационных систем, и последующее их использование для проектирования и разработки программы, удовлетворяющей требованиям надежности, производительности и безопасности.

Конечной целью исследовательской деятельности являлось создание аналитической информационной системы для организаций, ведущих предпринимательскую деятельность в сфере электронной коммерции, используя сайт интернет-магазина. Выбор такого направления разработки обоснован актуальностью и востребованностью данной модели ведения бизнеса, что подтверждается исследованиями специализирующихся на рынке информационных технологий консалтинговых и информационных агентств, согласно которым число интернет-магазинов стабильно растет на протяжении последних нескольких лет [2].

В соответствии с современными тенденциями в области проектирования распределенных информационных систем программа, которой посвящена данная работа, разработана и реализована на основе трехуровневой архитектуры клиент-сервер (рисунок 1).

Первый уровень архитектуры – слой клиента. Это самый верхний уровень приложения, непосредственно взаимодействующий с пользователем. Здесь сосредоточены исключительно элементы управления системой – интерфейс, главная функция которого – представление результатов, понятных пользователю.

Ввиду архитектурных особенностей выбранного подхода пользовательская часть системы представляет собой так называемый «тонкий» клиент (англ. thin client), что подразумевает отсутствие прямых подключений к базе данных и делегирование основных функций обработки информации на следующие слои архитектуры. На этом уровне обычно допускается только простейшие элементы, включая механиз-

Мантуленко Алексей Вячеславович (mantulenko83@mail.ru), кафедра математики и бизнес-информатики, Самарский университет, 443086, Российская Федерация, г. Самара, Московское шоссе, 34.

Шохин Александр Олегович (int.47@yandex.ru), студент IV курса группы 23402.50 – направления бизнес-информатики факультета экономики и управления, Самарский университет, 443096, Российская Федерация, г. Самара, Московское шоссе, 34.

мы авторизации пользователей и шифрование данных, а также несложные манипуляции с данными, предварительно загруженными из источника [3].

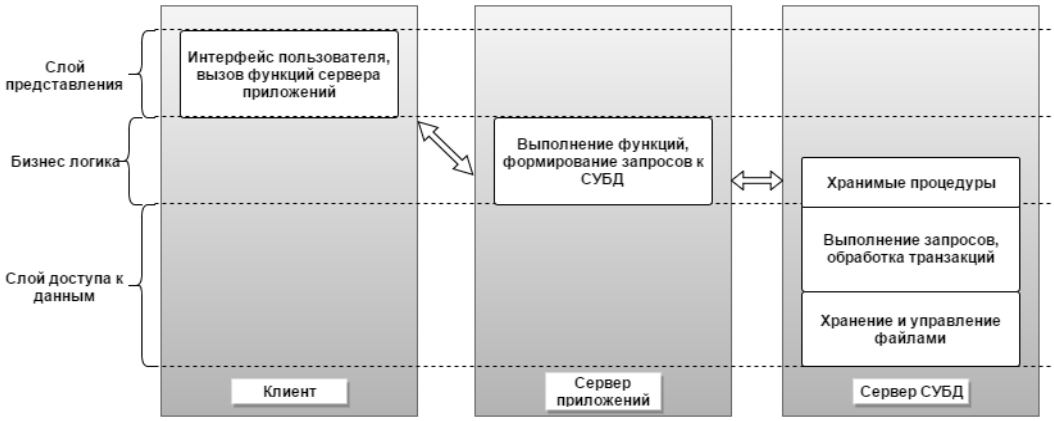

Рис. 1. Пример трехзвенной архитектуры клиент-сервер

На втором уровне архитектуры – слой логики приложения. Будучи связующим звеном между клиентом и сервером данных, этот компонент выполняет координирующую роль в системе. На этом слое располагаются алгоритмы, выполняющие логические решения, производящие вычисления и расчеты и обрабатывающие данные между окружающими слоями. Использование подобной структуры системы позволяет получить в качестве преимуществ возможности горизонтального масштабирования и отсутствие необходимости вносить изменения в программный код клиента. Выделение слоя логики приложения на отдельный уровень архитектуры направлено на увеличение производительности, безопасности и отказоустойчивости всей системы.

Последний, третий слой – это сервер данных, обеспечивающий хранение информации и ее извлечение по требованию логического слоя для последующей обработки и передаче конечного результата клиенту пользователя. Основные подключения к слою данных осуществляются только с уровня логики. Уровень данных представлен системой управления базами данных, а также фрагментами логики, содержащимися непосредственно в базе данных, такими как хранимые процедуры [4].

Одним из важнейших компонентов в архитектуре приложения является поддержка плагинов для информационной системы. Данная возможность реализована на основе технологий Reflection API, входящих в состав платформы .NET Framework и относящихся к пространству имен System.Reflection.

Технологии рефлексии позволяют программе производить мониторинг собственного состояния и модифицировать свою структуру и поведение прямо в процессе исполнения исходного кода. [1,5]

Используя специальный реализованный интерфейс, разработчики могут создать плагин в виде DLL библиотеки расширения приложения, который может быть запущен из основной программы и функционировать с ней как единое целое.

Все успешно загруженные и готовые к запуску плагины отображаются в соответствующем разделе приложения с указанием данных полученных из интерфейса плагина: его название, описание и текущую версию.

Все, что требуется от конечного пользователя – разместить нужные ему плагины в папке «Plugins», расположенной в корневой директории программы, и они будут готовы к использованию.

Это дает существенные возможности для расширения стандартного функционала приложения, позволяет конфигурировать информационную систему под нужды конкретного предприятия или пользователя и способствует развитию приложения как гибкой программной платформы.

Помимо вышеописанных технологий в разработанной информационной системе применяются механизмы развертывания приложения и его последующего систематического обновления.

В качестве инструмента развертывания используется сервис Microsoft ClickOnce (рисунок 2). Использование технологии ClickOnce дает программе следующие преимущества:

- автоматическое обновление – приложение самостоятельно проверяет наличие новой версии программы и в случае подтверждения загружает не все файлы, а только изменившиеся, и мгновенно устанавливает обновление;
- приложение, развернутое по технологии ClickOnce самодостаточно и автономно, его наличие и функционирование никак не влияет на остальные установленные на компьютере программы;
- в отличие от традиционных способов установки программ приложение ClickOnce не требует наличие прав администратора для установки;
- при установке программы также будут установлены необходимые для работы приложения компоненты, например, такие как .NET Framework.

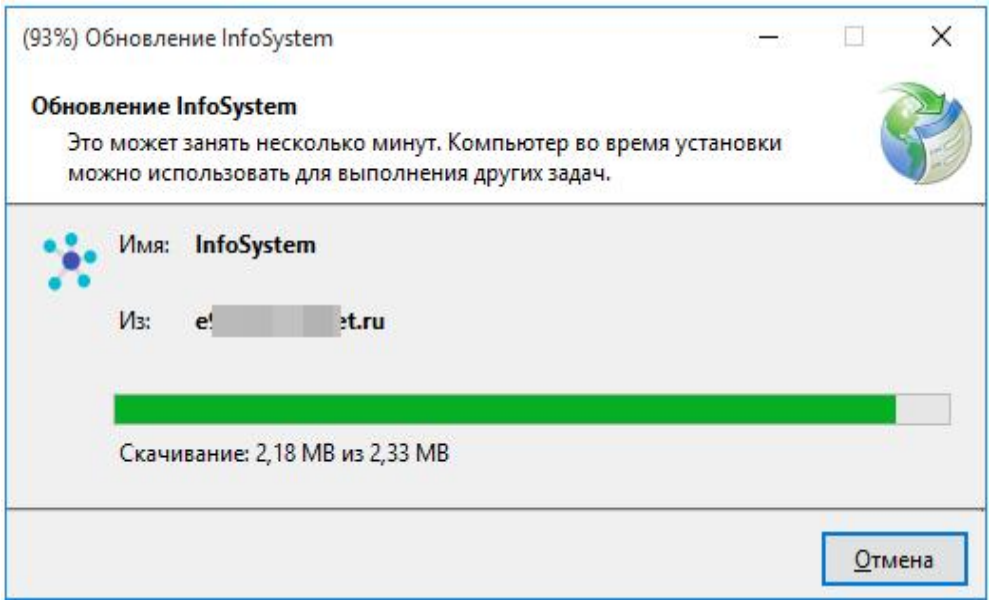

Рис. 2. Обновление программы, развернутой по технологии ClickOnce

Одной из технологий, использующихся для обеспечения максимально возможного быстродействия и высокого уровня производительности, является асинхронное программирование.

В разработанной программе использование методов асинхронного программирования направлено на поддержание работоспособности системы во время выполнения ресурсоемких задач, требующих длительной обработки данных, например, парсинг XML файла или экспортирование отчета. В отличие от обычного асин-

хронный вызов функции ожидает завершение выполнения операции, позволяя программе выполнять другую работу, которая не зависит от функции, ожидающей результат.

Немаловажным для потребителя является гарантия безопасности хранения данных. Для защиты таких данных как пароли, сведения о подключении к базе данных и прочее в созданной информационной системе задействуются ассиметричный алгоритм шифрования RSA (рисунок 3), входящих в одну из библиотек платформы .NET Framework. Ввиду того, что криптостойкими считаются ключи не менее 2048 бит, то именно такая длина ключа и используется в реализованной программе, так как большее значение неизбежно скажется на производительности отрицательным образом. [2, 5]

```
static public byte[] RSAEncrypt(byte[] DataToEncrypt, string publicXml, bool DoOAEPPadding)
{
    try
    {
        byte[] encryptedData;

        using (RSACryptoServiceProvider RSA = new RSACryptoServiceProvider(4096))
        {
            RSA.FromXmlString(publicXml);

            encryptedData = RSA.Encrypt(DataToEncrypt, DoOAEPPadding);
        }
        return encryptedData;
    }

    catch (CryptographicException e)
    {
        Console.WriteLine(e.Message);

        return null;
    }

}
```

Рис. 3. Пример реализации алгоритма RSA с помощью поставщика служб криптографии для платформы .NET Framework

База данных MySQL, являющаяся основным компонентом слоя данных на третьем уровне архитектуры разработанной информационной системы, была развернута на платформе облачных технологий Microsoft Azure (рисунок 4).

Microsoft Azure отличается повышенными показателями эффективности, производительности, отказоустойчивости и безопасности благодаря использованию гибридных, географически-распределенных и автономных систем обеспечивающих бесперебойное, полнофункциональное состояние развернутых в облаке баз данных [6].

С помощью среды MySQL Workbench в базе данных были созданы две таблицы: основная – для хранения информации о заказах, и таблица, содержащая аутентификационные данные пользователей информационной системы.

Следует отметить, что в таблице аутентификации не хранятся пароли в их изначальной форме ввиду нецелесообразности данного подхода, так как если к учетным данным пользователей получит доступ злоумышленник, то он без труда сможет их использовать.

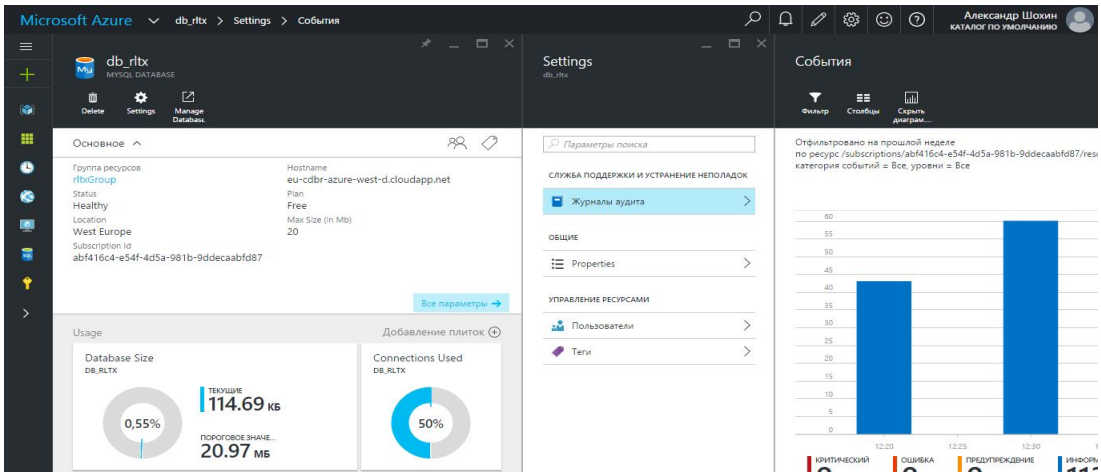

Рис. 4. Панель управления и мониторинга БД в Microsoft Azure

В данном случае в БД записывается не сам пароль, а его хеш-значение, полученное после обработки пароля алгоритмом криптографического хеширования SHA-2, действие которого реализуется одноименной функцией MySQL.

В качестве хеш-функции был выбран именно этот алгоритм поскольку он считается более безопасным, чем его предыдущая версия SHA-1 или традиционный MD5.

Несмотря на то, что процедура хеширования считается необратимой, опрометчиво допускать отсутствие угроз безопасности, так как существуют различные методы атак на хеши, позиционирующиеся как способные вскрыть даже сложнообратимые хеш-функции [7].

Для повышения степени защищенности паролей при разработке информационной системы был применен метод «соленых» хешей. Суть данного метода заключается в том, что перед процедурой хеширования пароля проводится его конкатенация со случайным модификатором, в криптографии называемым «солью», чтобы в результате получилось измененное хеш-значение, образованное на основе удлиненной строки пароля. Это усложняет атаку злоумышленника на хеши паролей и призвано защитить от таких механизмов вскрытия как радужные таблицы.

В качестве размера хеша выбрано значение 256 бит, что является оптимальным вариантом для разработанной информационной системы.

Чтобы минимизировать затраты на хранение хешей в базе данных, хеш-значения содержатся в таблице в виде бинарной строки (рисунок 5), в которую конвертируются при сохранении и преобразовываются обратно при сравнении во время аутентификации. В перспективе такой подход позволяет сэкономить до 45% памяти.

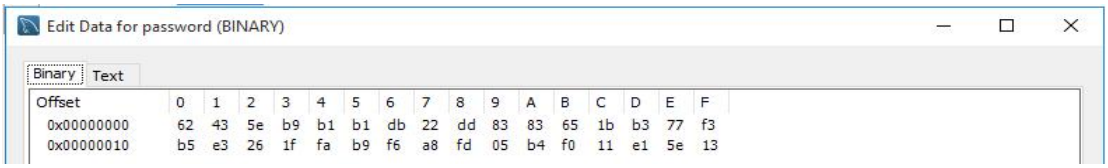

Рис. 5. Хранение хеша паролей

Что касается основной таблицы, содержащей данные о заказах, которые непосредственно обрабатываются в программе, то особенность работы с ней заключается в использовании индексов и составлении запросов с учетом селективности.

Помимо прочего разработанная информационная система в своей работе задействует такой инструмент СУБД MySQL как хранимые процедуры, чье использование повысила быстродействие соответствующего раздела программы примерно на 70 %. [3]

Учитывая все вышеописанные технологические особенности реализации инфраструктуры базы данных MySQL, можно констатировать, что мероприятия по ее разработке и оптимизации должны не ограничиваться простым созданием таблиц, а включать в свой перечень комплекс специфических методов и принципов, направленных на обеспечение безопасности и производительности обработки данных для созданной информационной системы.

Принимая во внимание все положительные эффекты от использования при разработке вышеописанных технологий и практик, можно сделать вывод, что для достижения высоких уровней производительности, отказоустойчивости и безопасности информационных систем, необходимо детально рассматривать все аспекты работы программы и для каждого из них подбирать наиболее подходящую технологическую реализацию, так как применение современных технологий должным образом положительно сказывается на опыте эксплуатации программы и повышает эффективность от внедрения данной информационной системы в деятельность предприятия.

Библиографический список

1. Рихтер Дж. CLR via C#. Питер, 2012. 928 с.
2. Надеин Н.В., Тюкавкин Н.М. Услкги делового характера сервисных организаций, // Вестник Самарского государственного университета. 2015. № 5 (127). С. 117–122.
3. Сараев А.Л. Динамическая многофакторная модель модернизации производственного предприятия. Вестник Самарского государственного университета. 2015. № 5 (127). С. 224–232.
4. Сараев А.Л. Закономерности взаимодействия потребителей и производителей в условиях непрерывного конкурентного рынка/ Сараев А.Л., Сараев Л.А.// В сборнике: Актуальные проблемы развития финансово-экономических систем и институтовматериалы и доклады I Международной научно-методической конференции: в 2 ч. Самарский государственный университет; под общ. ред. А.Н. Сорочайкина. 2010. С. 58–68.
5. Скорниченко Н.Н. Развитие сферы услуг в современной экономической системе: монография / Агаева Л.К., Арисова М.Б., Башкан Е.А., Безлепкина Н.В., Васильчук О.И., Гарькина Н.Г., Голдобина М.В., Гоман И.В., Каширина М.В., Ковтуненко А.В., Кононова Е.Н., Курносова Е.А., Манукян М.М., Медведева Е.В., Мельников М.А., Мокина Л.С., Насакина Л.А., Оруч Т.А., Прыткова Н.И., Скорниченко Н.Н. и др.// Самара, 2016.
6. Библиотека классов .NET Framework. https://msdn.microsoft.com/ru-ru/library/mt472912(v=vs.110).aspx (дата обращения: 04.03.2016)
7. Документация MySQL. URL: http://dev.mysql.com/doc/refman/5.7/en/ (дата обращения: 11.03.2016)

References

1. Richter George. CLR via C#. Peter, 2012. 928 S.
2. Nadein N. In. Ulchi business service organizations / N/In. Nadein, N.M. Tyukavkin // Vestnik of Samara state University. 2015. No. 5 (127). P. 117–122.

3. Saraev A.L. Dynamic multifactor model of modernization production enterprise. Vestnik of Samara state University. 2015. No. 5 (127). P. 224–232

4. Saraev A.L. regularities of interaction between consumers and producers in the context of continuous competitive marke t / A.L. Sheds, L.A. Barnes // In collection: Actual problems of development of financial-economic systems and institutionally and the reports of the I International scientific-practical conference: in 2 parts. Samara state University; Under the General editorship of A. N. Sorochkina. 2010. P. 58–68.

5. Kornichenko, N.N. The development of the service sector in modern economic system: monograph / L.K. Agayev, Arisawa M.B., Governor E.A., Bezlepkina N. In. Vasilchuk O. I., Garkina N.G., Goldobin, M.V., Goman, I.V., Kashirina M.V., Kovtunenko V.A., Kononov E.N., Kurnosov, E.A., Manoogian, M.M., Medvedev E.V., Melnikov M.A., Mokina HP, Nasakin L.A., Oruch T.A., Prytkova N. And. Kornichenko, N.N. . Samara, 2016.

6. Class library .NET Framework. https://msdn.microsoft.com/ru-ru/library/mt472912(v=vs.110).aspx (date accessed: 04.03.2016)

7. Documentation MySQL. URL: http://dev.mysql.com/doc/refman/5.7/en/ (accessed: 11.03.2016)

A.V. Mantulenko, A.O. Shokhin*

DESIGN AND DEVELOPMENT OF INFORMATION SYSTEM USING MODERN TECHNOLOGIES

This article discusses the using of some modern software technologies and describes their features and impacts in developing information systems according current requirements of reliability, security and performance.

Key words: information system, database MySQL, modern technologies, Microsoft Azure, cryptographic algorithm RSA, hashing SHA-2.

Статья поступила в редакцию 17/I/2016.
The article received 17/I/2016.

* *Mantulenko Alexei Vyacheslavovich* (mantulenko83@mail.ru), Department of Mathematics and Business Informatics, Samara University, 34, Moskovskoye shosse, Samara, 443086, Russian Federation.

Shokhin Aleksandr Olegovich (int.47@yandex.ru), Department of Mathematics and Business Informatics, Samara University, 34, Moskovskoye shosse, Samara, 443086, Russian Federation.