

УДК 004.652.3

**БЕЗОПАСНОСТЬ ОРГАНИЗАЦИИ СИСТЕМЫ ОНЛАЙН ОПЛАТЫ  
КОММУНАЛЬНЫХ УСЛУГ****Чернобай Ю.А., Романченко П.К. Теплинский С. В.***Донецкий национальный технический университет**кафедра компьютерной инженерии**E-mail: [4er@bk.ru](mailto:4er@bk.ru)***Аннотация:**

**Чернобай Ю.А., Романченко П.К., Теплинский С. В. Безопасность организации системы онлайн оплаты коммунальных услуг. Рассмотрены и проанализированы подходы, которые могут быть использованы при создании аналогичных проектов. Обоснованы методы, выбранные для создания конкретной модели. Описан общий принцип работы модели.**

**Общая постановка проблемы**

В наше время трудно представить себе мир, в котором нет баз данных. Для хранения огромных массивов информации пока не было придумано ничего лучше. И очень актуальным вопросом является проблема защиты этих данных от несанкционированных действий посторонних лиц. Особенно остро вопрос защиты данных встает при написании баз данных хранящих информацию об использовании электронной валюты, в нашем случае системы онлайн оплаты коммунальных услуг.

**Решение поставленных в статье задач**

Наиболее распространенным методом работы с базами данных является создание системы управления базами данных (СУБД) — специализированная программа (чаще комплекс программ), предназначенная для организации и ведения базы данных. Но главным показателем эффективности СУБД является её защищенность. Технологии, как защиты так и взлома, не стоят на месте, и то что сегодня может считаться абсолютно безопасным завтра морально устареет. Существующие СУБД, кроме всего прочего, для защиты информации использующую несколько уровней доступа, определяющих права пользователей на доступ к данным. Однако хотя эта система и очень удобна, но в плане безопасности остается не идеальной. Пароль может быть разглашен самим пользователем или добыт любым другим путем. Как же тогда защитить свою базу? Первое что надо подумать, а стоит ли вообще давать пользователю возможность удаленного администрирования? В случае с созданием системы для онлайн оплаты коммунальных платежей пользователю достаточно лишь получить некоторые данные из базы, относящиеся к своей учетной записи, и, на основе полученных данных, произвести некоторые действия. Для администрирования можно разработать отдельную клиентскую программу, в которой будет возможность работать только локально - непосредственно на сервере с установленной базой. Но и этого не достаточно для полной защиты. Рассмотрим несколько способов, которые используются для организации взаимодействия клиента с базой данных.

Первый способ взаимодействия – двухуровневая архитектура «клиент-сервер». В этом случае база данных помещается на сетевом сервере, однако программа клиента лишена возможности прямого доступа к БД. Доступ к БД регулируется специальной программой – сервером БД.

Взаимодействие сервера БД и клиента реализуется с помощью SQL-запросов, которые формирует и отправляет серверу клиент. Сервер, приняв запрос, выполняет его и возвращает результат клиенту. В клиентском приложении в основном осуществляются интерпретация полученных от сервера данных, реализация пользовательского интерфейса, а также

реализация части бизнес-правил. Двухуровневая архитектура клиент сервер представлена на рисунке 1.

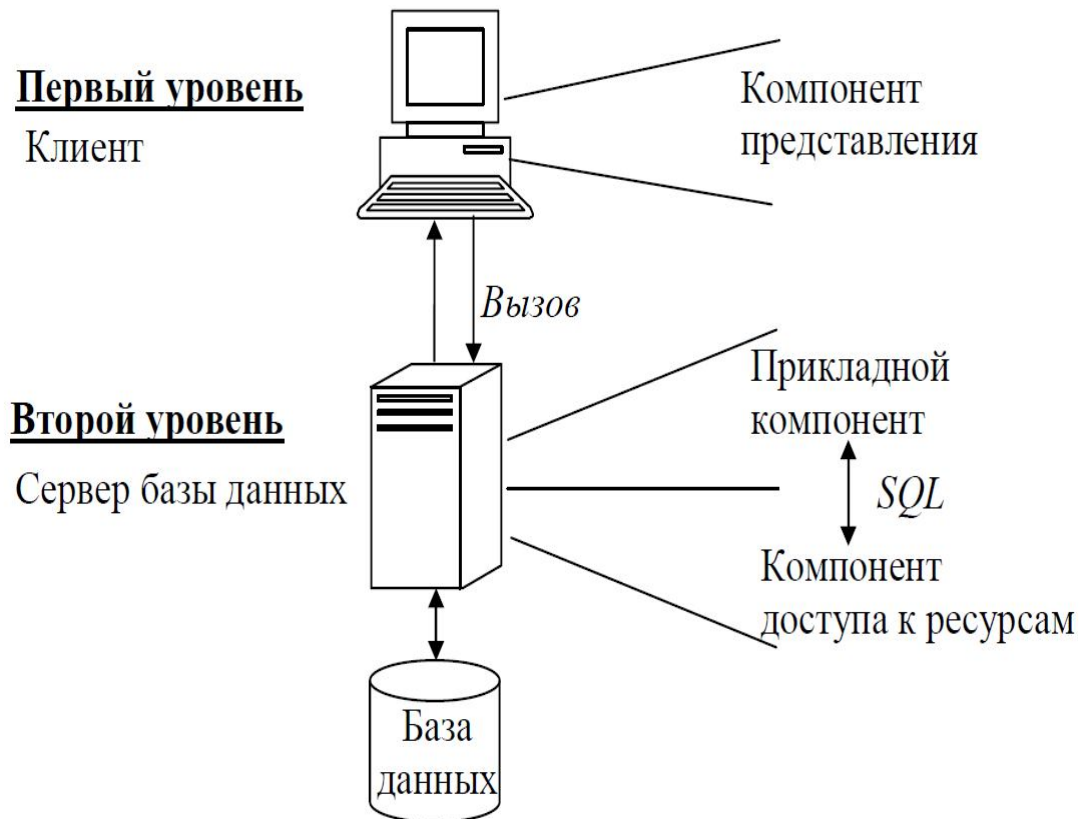


Рисунок 1 - Двухуровневая архитектура «клиент-сервер».

Второй способ взаимодействия – трехуровневая архитектура «клиент-сервер». В этой модели процесс, выполняющийся на клиенте, отвечает за интерфейс с пользователем. Обращаясь за выполнением услуг к прикладному компоненту.

Прикладной компонент реализован как группа процессов, выполняющих прикладные функции, и называется сервером приложения. Все операции над информационными ресурсами баз данных выполняются компонентами доступа к ресурсам. Таким образом, в отличие от предыдущей архитектуры, обмен между клиентом и сервером осуществляется посредством специально разработанного набора команд (API), а не SQL запросам. Набор этих команд определяется разработчиком, что позволяет ему ограничить набор допустимых действий с данными доступным клиенту. Трехуровневая архитектура клиент сервер представлена на рисунке 2.

#### **Реализация цели**

При разработке системы онлайн оплаты коммунальных услуг целесообразней использовать трехуровневую архитектуру клиент-сервера. В нашем случае клиентская программа, написанная на actionscript 3, загружается на компьютер пользователя как обычный флеш файл с сервера. Запускаясь, эта программа подключается к серверу, написанному на C#, использующую базу данных MS SQL. Запросы клиента передаются в формате XML, который на стороне сервера расшифровывается и на основе полученных данных формируется SQL запрос к базе. При этом сервер перед формированием запроса проверяет все входящие команды. Набор команд обмена между сервером и клиентом очень ограничен и позволяет выполнять только определенный набор действий с базой

данных. Однако такой подход для создания узко специализированных СУБД оправдывает малую функциональность большой защищенностью базы данных.

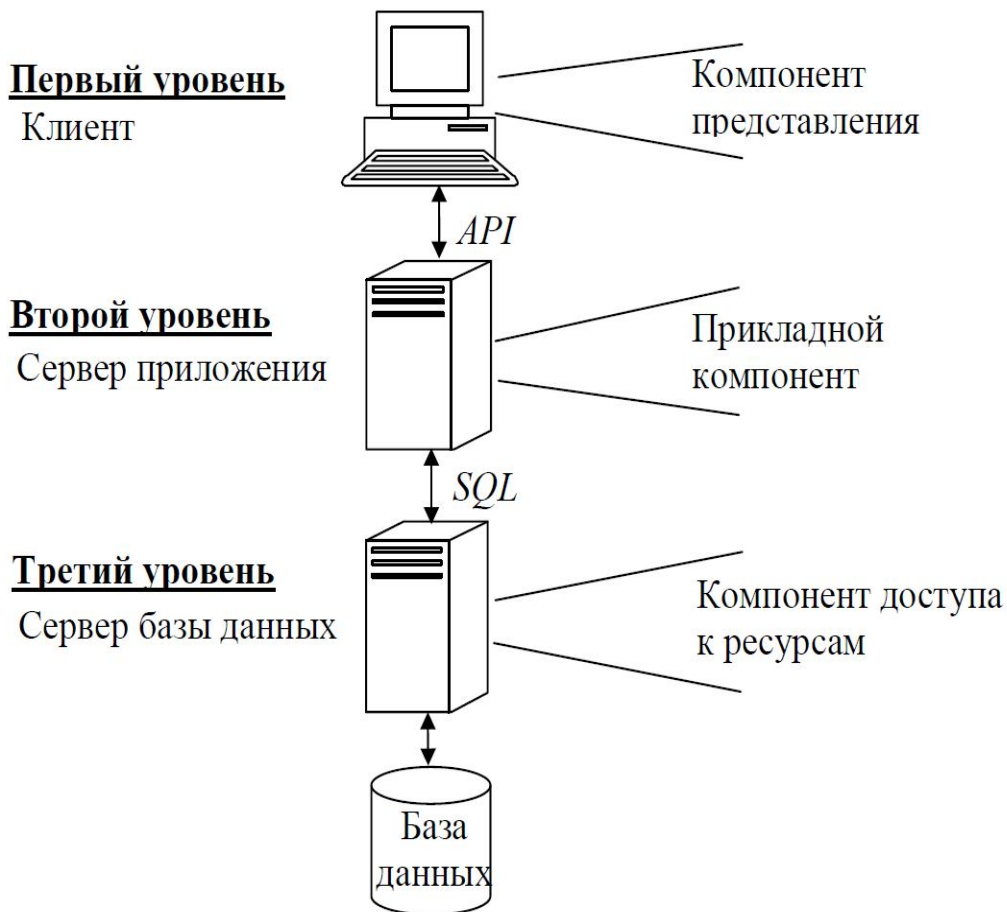


Рисунок 2 - Трехуровневая архитектура «клиент-сервер».

Формат XML очень удобен для передачи данных, но он является открытым и понятным даже для обычного человека. Так что для защиты данных пользователей при передаче по сети используются стандартные методы шифрования передаваемых данных.

Реализуемая программа-клиент будет располагаться на компьютере с установленным http сервером, и будет доступна любому пользователю, имеющему доступ в Интернет. Загружаться на компьютер пользователя и запускаться она будет как обычный флеш-ролик через браузер, и не будет требовать дополнительной установки. Для работы программы-клиента на компьютере пользователя должен будет установлен любой браузер и Flash Player версии не ниже 10.0.

Программа сервер для большей безопасности может располагаться на любой другой серверной машине без установленных http и ftp серверов и защищенной firewall-ом. Такая локализация частей программ значительно повышает безопасность сервера с базами данных и исключает возможность удаленного доступа к файлам программы и базы данных.

При переходе по ссылке пользователь загружает и запускает программу-клиента. При запуске программы предлагается ввести данные для осуществления подключения к учетной записи пользователя. Программа предлагает ввести «Лицевой счет квартиры», Фамилию, имя, отчество», и «Номер квартиры». Поле «Лицевой счет квартиры» известно только пользователю и исполняет роль пароля. Окно ввода данных пользователем представлено на рисунке 3.

**Введите данные**

Лицевой счет квартиры

Фамилия, имя, отчество

Номер квартиры

Рисунок 3. - Окно ввода данных пользователем.

После авторизации пользователю предлагается квитанция в которую он вносит данные про использование им коммунальных услуг. Окно квитанции представлено на рисунке 4. При нажатии на кнопку «Оплатить» внесенные данные отправляются на сервер где проходят проверку на корректность и вносятся в базу.

**Заполните квитанцию**

**КВИТАНЦИЯ** Код ЕРЦ 1234567890

Абонент- Иванов Иван Иванович Адрес Артема 65

Вид оплаты	Сумма платежа	Период оплаты	Пеня	Показания счетчика		
				Кон.	Нач.	Разн.
Квартплата	30	1.04.10				
Эл. энергия	45			3	1	2
Газ	30					
Вода	45			3	1	2
Стоки	45			3	1	2
Отопление	30					
Горячая вода	45			3	1	2
Вывоз мусора	30					
<b>Итого</b>	<b>300</b>					

Рисунок 4. - Окно квитанции

Также существует возможность просмотра текущих тарифов при нажатии на кнопку «Тарифы». Окно тарифов представлено на рисунке 5.

## Тарифы

Наименование платежа	Без счетчика	Со счетчиком
Квартплата	10	15
Электроэнергия	10	15
Газ	10	15
Вода	10	15
Стоки	10	15
Отопление	10	15
Горячая вода	10	15
Вывоз мусора	10	15

Выход

Рисунок 5. – Окно тарифов

**Выводы**

Такой подход к защите данных идеален для программ такого рода как система оплаты коммунальных услуг. Написание собственной СУБД использующей клиент серверную архитектуру, где для взаимосвязи клиента с сервером, используется низкий уровень сетевого взаимодействия, наиболее оптимален для данной задачи.

Этот подход к защите данных может быть использован во многих областях жизни от социальной и промышленной сферы до игровой индустрии. Однако написание таких приложений это трудоемкий и длительный процесс. В случае если защита не так важна, или необходима возможность удаленного администрирования, можно использовать стандартные подходы организации работы с базами данных.

**Список литературы**

1. А.В. Маркин. *Построение запросов и программирование на SQL*, Рязань 2008. - 312с.
2. Роберт Вийера, "Программирование баз данных Microsoft SQL Server 2005 для профессионалов", 2008. – 256с.
3. Джеффри Д. Ульман, Дженнифер Уидом: Введение в системы баз данных, 2005. – 457с.
4. Интернет-ресурс. - Режим доступа : [http://www.cyberinfo.ru/3/1826\\_1.htm](http://www.cyberinfo.ru/3/1826_1.htm). - Загл. с экрана.
5. Интернет-ресурс. - Режим доступа : <http://www.diwaxx.ru/hak/vzlompass1.php> . - Загл. с экрана.