**Многоуровневая архитектура клиент-сервер**

В нашей информационной системе, наша компания выбрала многоуровневую архитектуру клиент-сервер потому что:

**Имя шаблона:**

Многоуровневая архитектура клиент-сервер

**Описание:**

Разновидность архитектуры клиент-сервер, в которой функция обработки данных вынесена на один или несколько отдельных серверов. Это позволяет разделить функции хранения, обработки и представления данных для более эффективного использования возможностей серверов и клиентов.

Многоуровневая архитектур представлена на Рисунке 1

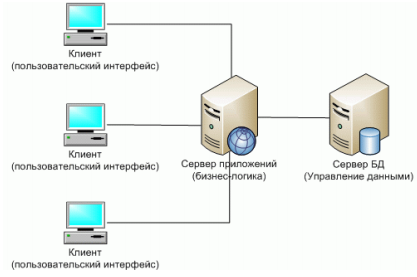


Рисунок 1. Многоуровневая архитектура клиент-сервер

**Пример:**

Сервер БД представлен MySQL-сервером; сервер приложений технологиями: ADO.NET, ASP.NET и web-сервером IIS; роль клиента выполняет любой web-браузер.

Браузер клиента *1->* Сервер IIS *2->* Исполняющая среда ASP.NET 2.0 *3->* Провайдер данных ADO.NET 2.0 *4->* Сервер MySQL *5->* Провайдер данных ADO.NET 2.0 *6->* Исполняющая среда ASP.NET 2.0 *7->* Сервер IIS *8->* Браузер клиента

1 - браузер клиента отправляет HTTP-запрос;

2 – на стороне сервера служба Web Internet Information Server (web-сервер IIS) определяет тип запрашиваемого ресурса, и для случая запроса \*.aspx (расширение файлов страниц ASP.NET) загружает соответствующее ему (запросу) расширение Internet Server Aplication Programming Interface (ISAPI). Для страниц aspx это расширение isapi\_aspnet.dll. IIS также осуществляет идентификацию и авторизацию пользователя от которого поступил запрос. В свою очередь расширение isapi\_aspnet.dll загружает фабрику обработчиков ASP.NET. Далее, фабрика обработчиков создает объектную модель запрашиваемой страницы и обрабатывает действия пользователя.

3 – в ходе генерации ответа приложению ASP.NET может потребоваться обращение к БД, в этом случае используя библиотеки классов провайдера данных ADO.NET 2.0, выполняющая среда обращается к серверу БД;

4 – провайдер данных ADO.NET 2.0 передает запрос на операцию с БД серверу MySQL;

5 – сервер MySQL осуществляет обработку запроса, выполняя соответствующие операции с БД ;

6 – провайдер данных ADO.NET 2.0 передает результаты запроса объекту страницы;

7 – объект страницы с учетом полученных данных осуществляет рендеринг графического интерфейса страницы и направляет результаты в выходной поток;

8 – сервер IIS отправляет содержимое сгенерированной страницы клиентскому браузеру.

**Когда используется:**

Суть многоуровневой архитектуры в том, что помимо сервера *БД* и приложений-клиентов дополнительно присутствует еще один или несколько серверов приложений. *Сервер приложений* является промежуточным уровнем, обеспечивающим организацию взаимодействия клиентов и сервера *БД*. *Сервер приложений* также называют брокером данных ( broker - посредник). Чаще всего используют трехуровневую модель. Прежде, чем мы двинемся дальше, давайте разберемся, что же такое уровень. Имеется три основных уровня:

Уровень данных. Этот уровень отвечает за хранение данных. Как правило, для этого уровня выделяется отдельный ПК, на котором устанавливают один из SQL -серверов, например, *InterBase*. Клиентские ПК непосредственно не имеют никакой связи с этим уровнем.

Бизнес-уровень. Этот уровень предназначен для получения данных с уровня данных, выполнения окончательной проверки данных, и служит посредником между клиентами и данными. Как правило, сервера приложений находятся именно на этом уровне.

Уровень представления данных. Этот уровень известен так же, как уровень графического интерфейса пользователя. На этом уровне полученные данные отображаются в таких компонентах вывода данных, как *DBGrid*, *DBEdit*, *DBMemo* и проч. Разумеется, этот уровень находится на клиентских ПК.

**Преимущества:**

многократность повторного использования общих функций обработки данных в множестве клиентских приложений при существенной экономии системных ресурсов

параллельность в работе сервера приложений и сервера баз данных, причём сервер приложений может быть менее мощным, чем сервер баз данных

оптимизация доступа к базе данных через сервер приложений из клиентских мест путём диспетчиризации выполнения запросов в вычислительной сети

повышение скорости и надёжности обработки данных в результате дублирования программного обеспечения на нескольких серверах приложений, которые могут заменять друг друга в сети в случае перегрузки или выхода из строя одного из них

перенос функций администрирования системы по проверке полномочий доступа пользователей с сервера баз данных на сервер приложений.

**Недостатки:**

Неработоспособность сервера может сделать неработоспособной всю вычислительную сеть.

Администрирование данной системы требует квалифицированного профессионала.

Высокая стоимость оборудования

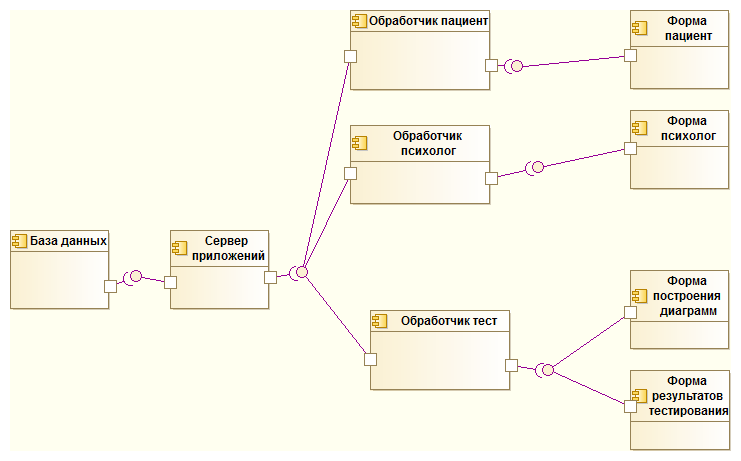


Рисунок 1. Диаграмма компонентов архитектуры ИС

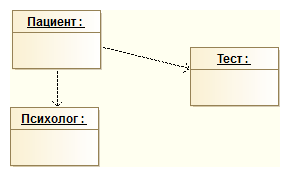


Рисунок 2.Концептуальня диаграмма классов.