

УДК 004.657

## **Разработка интернет - ориентированного клиент-серверного приложения для студенческого общежития**

*Мурчиков А.М., студент*

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,  
кафедра «Системы обработки информации и управления»*

*Научный руководитель: Белоусова В.И., доцент*

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,  
кафедра «Системы обработки информации и управления»*

[chernen@bmstu.ru](mailto:chernen@bmstu.ru)

### **Введение**

В наше время, в век информационных технологий, бумажная работа уходит в прошлое, и на замену бумаге приходят компьютеры. Сейчас почти вся бумажная работа заменена работой с различными программами или интернет - ориентированными клиент - серверными приложениями. Возникают задачи разработки программных приложений для автоматизации административной работы в разных организациях и учреждениях.

Не составляют исключения и вузы, для которых созданы целые системы, например, «Электронный университет»[1]. Для автоматизации работы отделов и подразделений вузов используются различные технологии, в том числе веб-сайты с методическими материалами, платформы 1С для отдела кадров[2], офисные пакеты для кафедральных систем [3].

Разработка автоматизированных систем для подразделений вуза имеет свои особенности, поскольку процессы, подлежащие автоматизации, весьма специфичны [4] и требуют правильной формализации с учетом всех тонкостей [5].

Обозначенная задача является актуальной и для студенческих общежитий. Практически во всех общежитиях все данные хранятся в бумажном виде, вся важная информация размещена на сотнях листах бумаги. Перед комендантом общежития возникает проблема быстрого поиска нужной информации, сортировки информации и, как следствие, предоставление нужных отчетов занимает намного больше времени, чем могло бы.

В статье предложен способ решения данной проблемы – внедрение интернет – ориентированного клиент – серверного приложения, которое позволит автоматизировать работу коменданта, хранить информацию и предоставлять ее в удобном виде, сортировать

и быстро искать информацию по многим критериям. Также данное приложение будет доступно и студентам, проживающим в общежитии. Студентам будет доступна лента объявлений, они смогут посмотреть необходимую информацию о себе, например, наличие медицинской справки, оплату за общежитие. Также студенты смогут посмотреть списки должников и списки на выселение.

Наличие аналогичных проблем в смежных областях [6] и поиск их решений свидетельствует об актуальности задачи. Альтернативных систем очень мало, что говорит об важности данного проекта. Автору не удалось найти сведения о подобных проектах, которые были бы разработаны специально для общежитий, поэтому было решено разработать данный проект.

### **Пользователи и возможности системы**

Данное клиент – серверное приложение рассчитано на пользователей трех разных групп: комендант, завхоз и сами студенты. У каждой группы пользователей есть свои права, которые удовлетворяют основным потребностям каждой группы.

Рассмотрим основные функции, которые предоставляет данный проект каждой группе пользователей.

Общие возможности, предоставляемые всем группам пользователей:

- Просмотр объявлений;
- Просмотр ближайших мероприятий;
- Просмотр списка задолжников по оплате;
- Просмотр плана общежития;
- Просмотр адреса и индекса общежития.

Возможности, предоставляемые только пользователю Комендант:

- Заселение студентов (подразумевает под собой ввод всей информации о студенте, выбор конкретной комнаты для заселения, простановки даты заселения и занесения всех этих данных в базу);
- Выселение студента (проставление метки о выселении студента и даты выселения. Данные о студенте не удаляются из базы данных для ведения журнала заселений и выселений);
- Поиск студентов по различным критериям отбора с возможностью редактировать данные о студенте;
- Просмотр списка выселенных студентов;

- Просмотр статистики по комнатам (число свободных мест в общежитии, число пустых комнат и список этих комнат, число занятых комнат и список этих комнат);
- Просмотр журнала заселений и выселений студентов (ФИО студента, статус – заселен или выселен, число операции);
- Заполнение объявлений и ближайших мероприятий (для этого разработаны 2 таблицы, в которые будут заноситься данные, а именно для Объявлений – ID объявления и само объявление, для Мероприятий – ID мероприятия, название мероприятия и дата проведения мероприятия).

Возможности, предоставляемые только пользователю Завхоз:

- Просмотр инвентаря студентов;
- Выдача инвентаря студентам;
- Удаление инвентаря у студента.

На рис.1 представлена DFD –диаграмма (диаграмма потоков данных) данного клиент – серверного приложения. Она показывает основные функции, которые доступны каждой из трех групп пользователей.

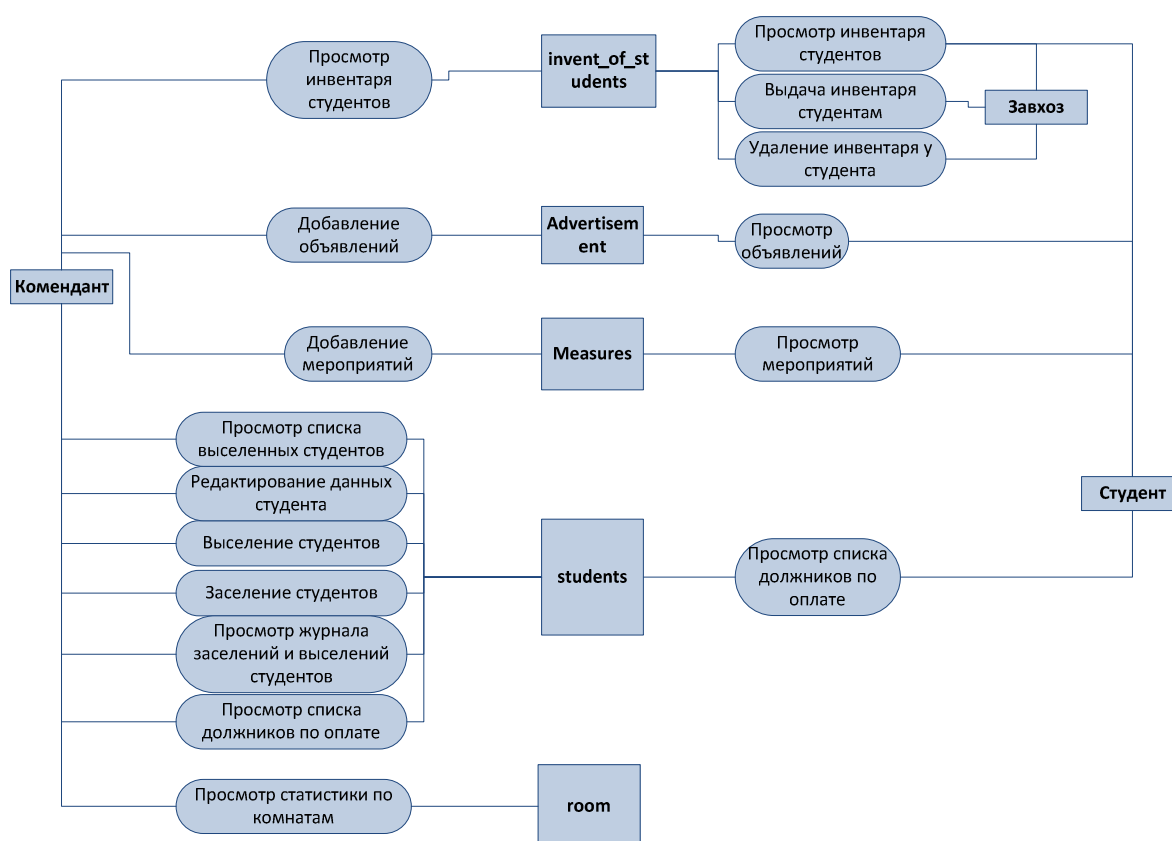


Рис. 1. DFD-диаграмма приложения

Для повышения удобства работы с сайтом был проработан его пользовательский интерфейс. После захода пользователя на сайт все данные о нем появляются справа в колонке. Это обеспечивает дополнительное удобство, особенно для студентов. Это избавляет студентов от поисков своей фамилии в различных списках. В этой колонке будут приведены такие данные, как ФИО, дата рождения, комната, дата заселения, оплатил ли студент за проживание или нет, наличие медицинской справки, отметка о выселении («Проживает» или «Выселен»), есть ли у студента нарушения и список этих нарушений, если они есть, а также находится ли студент в списке на выселение или нет.

### Архитектура приложения

Проект осуществлен на основе WAMP-технологии. Уровень данных представлен базой данных под управлением СУБД MySQL. Уровень бизнес – логики представлен программой на веб – ориентированном языке программирования PHP. В качестве веб – сервера использовался Apache HTTP-сервер. При разработке также использовался язык разметки HTML и язык описания внешнего вида документа CSS. Для обеспечения динамичности приложения и разгрузки сервера был использован язык JavaScript. Сценарии на языке JavaScript используются для обработки событий, например, для проверки правильности введенных данных в форму перед отправкой на сервер. Данная проверка выполняется не на серверной стороне приложения, а на клиентской, что позволяет снизить нагрузку на сервер.

Для обеспечения переходов между страницами были использованы POST и GET переменные, в которых передавалась информация, необходимая для наполнения страниц.

На рис. 2 представлена схема архитектуры приложения. На схеме представлены основные компоненты приложения и связи между ними.

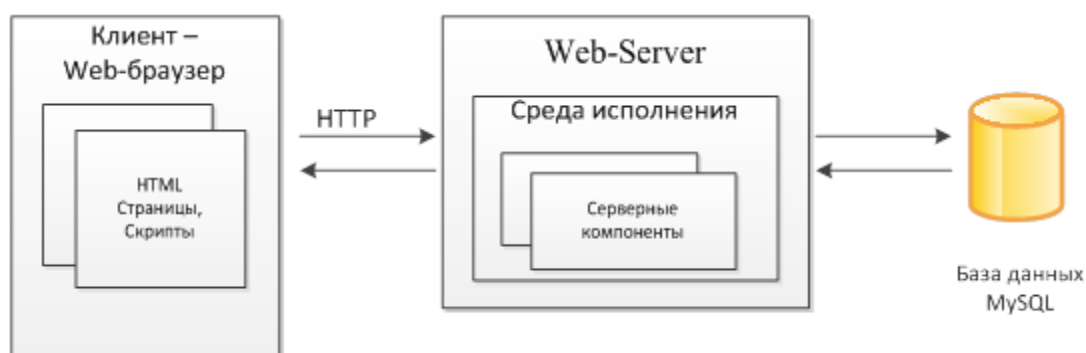


Рис. 2. Схема архитектуры приложения

Взаимодействие между клиентской стороной и серверной происходит по протоколу HTTP. HTTP (англ. *Hyper Text Transfer Protocol* — «протокол передачи гипертекста») — протокол прикладного уровня передачи данных. Основой HTTP является технология «клиент-сервер», то есть предполагается существование потребителей (клиентов), которые инициируют соединение и посылают запрос, и поставщиков (серверов), которые ожидают соединения для получения запроса, производят необходимые действия и возвращают обратно сообщение с результатом.

Разграничение прав доступа осуществляется путем аутентификации при заходе в приложение и использованием сессий в PHP. Все пользователи находятся в таблице Users. Если авторизоваться под пользователем, которого нет в этой таблице, то доступ к приложению будет закрыт. Для разграничения прав были использованы *сессии*, в которых хранилось имя пользователя системы. В зависимости от имени пользователя выдавались различные права на пользование системой. Так, на каждой странице перед отображением элементов просматривается переменная \$\_SESSION, и в зависимости от ее содержимого отображаются или скрываются некоторые элементы на странице.

### Структура базы данных

На рис.3 представлена графическая диаграмма даталогической модели базы данных.

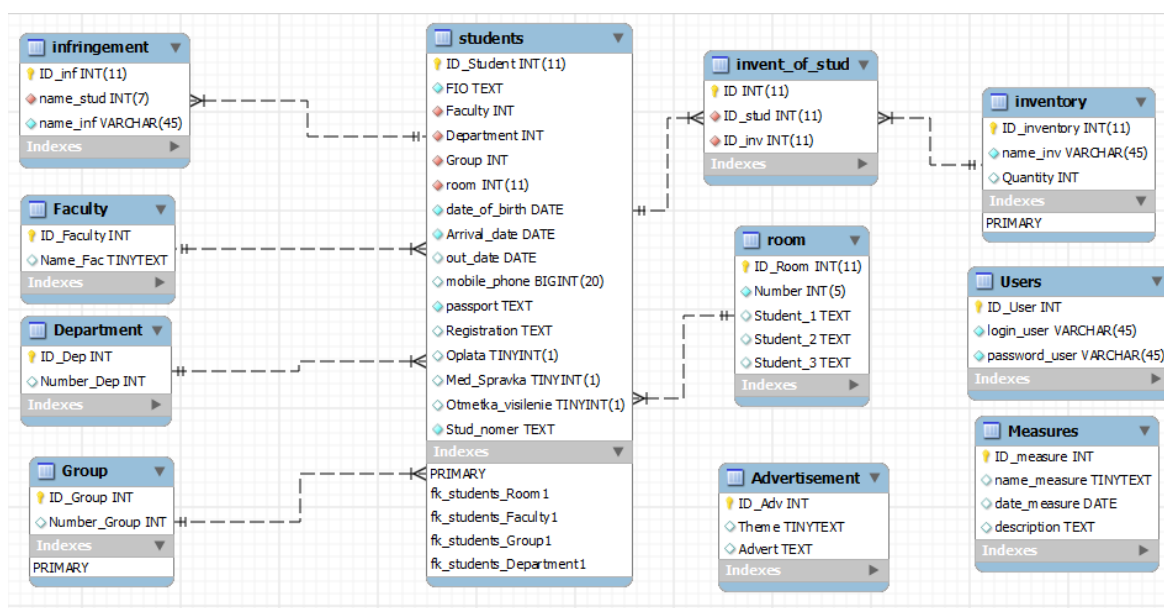


Рис. 3. Графическая диаграмма даталогической модели БД

Центральное место в представленной модели занимает таблица *students*. Она содержит всю необходимую информацию о студенте. Поле *out\_date* (дата выселения) при заселении остается пустым и заполняется, когда студент выселяется из общежития. Поля *FIO, Faculty, Department, Group, room, date\_of\_birth, Arrival\_date* (дата заселения), *passport* (серия и номер паспорта), *Stud\_nomer* (номер студенческого билета) – являются обязательными для заполнения.

Таблица *room* содержит номер комнаты и три места для студентов. При заселении в комнату поля *Student\_1, Student\_2 и Student\_3* заполняются по мере занятости. Если в комнате все места заняты, то при заселении она не отображается в списке доступных комнат. Таблицы *room* и *students* связаны связью 1:M.

Таблица *inventory* (инвентарь) содержит ID инвентаря и его название. Для того, чтобы одному студенту можно было выдать несколько предметов, и чтобы предмет с одинаковым названием можно было выдать нескольким студентам, создана таблица *invent\_of\_stud*, которая помогает реализовать связь M:M (многие ко многим).

Для фиксации нарушений создана таблица *infringement*. Она хранит ФИО студента – нарушителя и название нарушения. ФИО студента берется из таблицы *students* путем связи 1:M, а поле *name\_inf* (название нарушения) заполняется комендантом.

Для авторизации пользователей создана таблица *Users*, которая хранит логин и пароль пользователя. Если пользователя нет в данной таблице, то он не сможет получить доступ к приложению. Логин для студентов является номер их студенческого билета, который извлекается из таблицы *students*. Логин студента заносится в таблицу *Users* при заселении студента автоматически, и нет необходимости делать это вручную.

Для удобного выбора группы студента было создано 3 таблицы: *Faculty* – факультет, *Department* – кафедра, *Group* – номер группы. Они позволяют выполнять поиск по различным критериям и обеспечивают нормализованную структуру данных.

## **Заключение**

Применение предложенного в статье веб – ориентированного клиент – серверного приложения позволяет автоматизировать основную деятельность коменданта и завхоза студенческого общежития. Также данное приложение удобно для студентов, поскольку каждый студент может быстро и удобно получить все необходимые для него сведения.

Разработанное приложение может быть использовано в студенческом общежитии любого вуза. Его применение позволяет не только облегчить работу коменданта и завхоза, но и держать студентов в курсе всех объявлений и новостей.

### Список литературы

1. Балдин А.В., Тоноян С.А., Елисеев Д.В. Обработка архива кадровых данных средствами 1С // Инженерный журнал: наука и инновации. 2013. № 11 (23). Режим доступа: <http://engjournal.ru/catalog/it/hidden/1009.html> (дата обращения 01.03.2015).
2. Виноградов В.И., Виноградова М.В. Возможности программирования libreoffice/openoffice для создания программ обработки документов // Инженерный журнал: наука и инновации. 2014. № 1. Режим доступа: <http://engjournal.ru/catalog/it/hidden/1060.html> (дата обращения 01.03.2015).
3. Кулешов Д.С., Чёрненький В.М. Метод оперативного управления образовательными и реабилитационными ресурсами в условиях инклюзивного высшего профессионального образования // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 2014. № 3. С. 105-115.
4. Белоус В.В., Спиридонов С.Б., Постников В.М. Формализация процесса реализации цикла выпуска электронных периодических изданий // Наука и образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2013. № 11. DOI: 10.7463/1113.0658430.
5. Гонсалес Х.К., Галкин В.А. Анализ и синтез программного обеспечения сбора технических данных конечного интернет – пользователя // Инженерный вестник. МГТУ им. Н.Э. Баумана. Электрон. журн. 2014. № 11. Режим доступа: <http://engbul.bmstu.ru/doc/735760.html> (дата обращения 01.03.2015).