



Московский ордена Ленина, ордена Октябрьской Революции
и ордена Трудового Красного Знамени.
ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н. Э. БАУМАНА

Факультет: Информатики и систем управления
Кафедра: Проектирование и технология производства электронной аппаратуры (ИУ4)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА к курсовому проекту

“Разработка АСУ кафедры ИУ4”
версия 2.0

Студент: Трофимов Д.А. ИУ4-101
(фамилия, инициалы) (индекс группы)

Руководитель: Власов А.И.
(фамилия, инициалы)

Москва
2003

Содержание

Содержание	2
Введение	3
1 Основные задачи и ключевые особенности АСУ	3
2 Функциональная модель	4
2.1 Диаграмма прецедентов	4
2.2 Диаграмма внедрения	4
2.3 Классы АСУ	5
2.4 Диаграммы действий	6
3 Информационная модель	7
3.1 "MENU"	7
3.2 "MODULES"	8
3.3 "FUNCTIONS"	8
3.4 "PARAMS"	8
3.5 "GROUPS"	8
3.6 "USERS"	9
3.7 "USER_MENU"	9
3.8 ER-модель	10
4 Общая структура	10
5 Описание модулей АСУ	11
5.1 Модуль "DB"	11
5.2 Модуль "Connect"	12
5.3 Модуль "Menu"	13
5.4 Модуль "Plugins"	13
5.5 Модуль "Admin"	14
6 Описание структуры плагинов	17
7 Системные требования	19

Введение

Целью разработки была разработка программного комплекса, способного решать задачи систематизации различных модулей, создаваемых на кафедре, разбиения их на группы, повышения информативности и простоты использования модулей путем их стандартизации.

Иными словами, необходимо было разработать систему, создающую для конкретного пользователей древовидную структуру, содержащую весь набор программ, необходимых для решения задач, стоящих перед данным пользователем, причем этот набор должен автоматически обновляться с появлением новых модулей.

Результатом разработки стала автоматизированная система управления кафедры ИУ-4.

АСУ представляет из себя каталог программных модулей(плагинов), выполняющих различные функции. Модули представляют из себя обычные запускаемые файлы или файлы динамических библиотек (*.exe или *.dll), параметром для запуска которых служит имя конкретной функции, которую должен выполнить данный модуль.

В качестве языка программирования был выбран Borland Delphi 6, поскольку, являясь достаточно простым, он позволяет эффективно решать все необходимые задачи.

На данном этапе разработки в качестве системы управления базой данных АСУ используется СУБД MySQL. Эта СУБД была выбрана благодаря своей компактности, большой скорости, устойчивостью и легкостью в использовании. MySQL поддерживает язык запросов SQL в стандарте ANSI 92, и кроме этого имеет множество расширений к этому стандарту, которых нет ни в одной другой СУБД.

Однако, АСУ создавалось с учетом легкого перехода с одной СУБД на другую. Для этого все функции, специфичные для данной СУБД вынесены в отдельный модуль, изменение которого позволяет осуществить переход на другую СУБД без изменения основного кода системы.

При разработке системы планируется разработать два модуля подключения к СУБД: для СУБД MySQL и для СУБД Oracle.

АСУ также обеспечивает независимость от операционной системы сервера БД.

1 Основные задачи и ключевые особенности АСУ

Можно выделить следующие задачи, решаемые АСУ:

- быстрый поиск нужного модуля
- разграничение по доступу к различным модулям для разных пользователей
- автоматическая загрузка новых и обновление старых модулей по сети

Ключевые особенности АСУ:

- расширенная поддержка динамических библиотек (DLL) в качестве модулей, что позволяет хранить несколько форм в одном модуле, вызывая их с помощью различных функций, экспортируемы модулем или различных параметров, передаваемых данному модулю.
- практически неограниченная вложенность дерева меню*
- модульная структура позволяет легко обновлять как плагины, так и саму АСУ
- одновременный доступ к системе нескольких пользователей

2 Функциональная модель

Первоначально необходимо разделить всех пользователей системы на три класса: модераторов, которые могут изменять информацию о доступных плагинах АСУ(глобального меню), создавать структуру меню для конкретного пользователя, администраторов которые могут добавлять и удалять других пользователей, и собственно пользователей, которые могут только запускать модули, к которым им предоставил доступ модератор.

Сама АСУ также, фактически будет состоять из двух частей – администраторской(Admin) и пользовательской или клиентской(Client).

Интерфейс администратоской части включает в себя:

- подсистему управления пользователями – для добавления/изменения/удаления учетных записей пользователей и назначения им прав доступа
- средства для изменения структуры глобального меню
- средства для сопоставления пунктов глобального меню конкретным пользователям, т.е. для изменения структуры пользовательского меню

* Ограничена объемом ОЗУ и используемой ОС(см. раздел Системные требования).

2.1 Диаграмма прецедентов

Данная диаграмма описывает прецеденты в системе. Каждый прецедент инициируется определенным актером или может быть вызван другим прецедентом, и в свою очередь вызывать какой-либо прецедент.

Актером может служить любой внешний субъект, который поддерживает связь с системой, но не является ее частью. В данном случае было выбрано три актера: администратор, модераторов и пользователь.

Диаграмма прецедентов для АСУ ИУ4 представлена на рисунке 1.

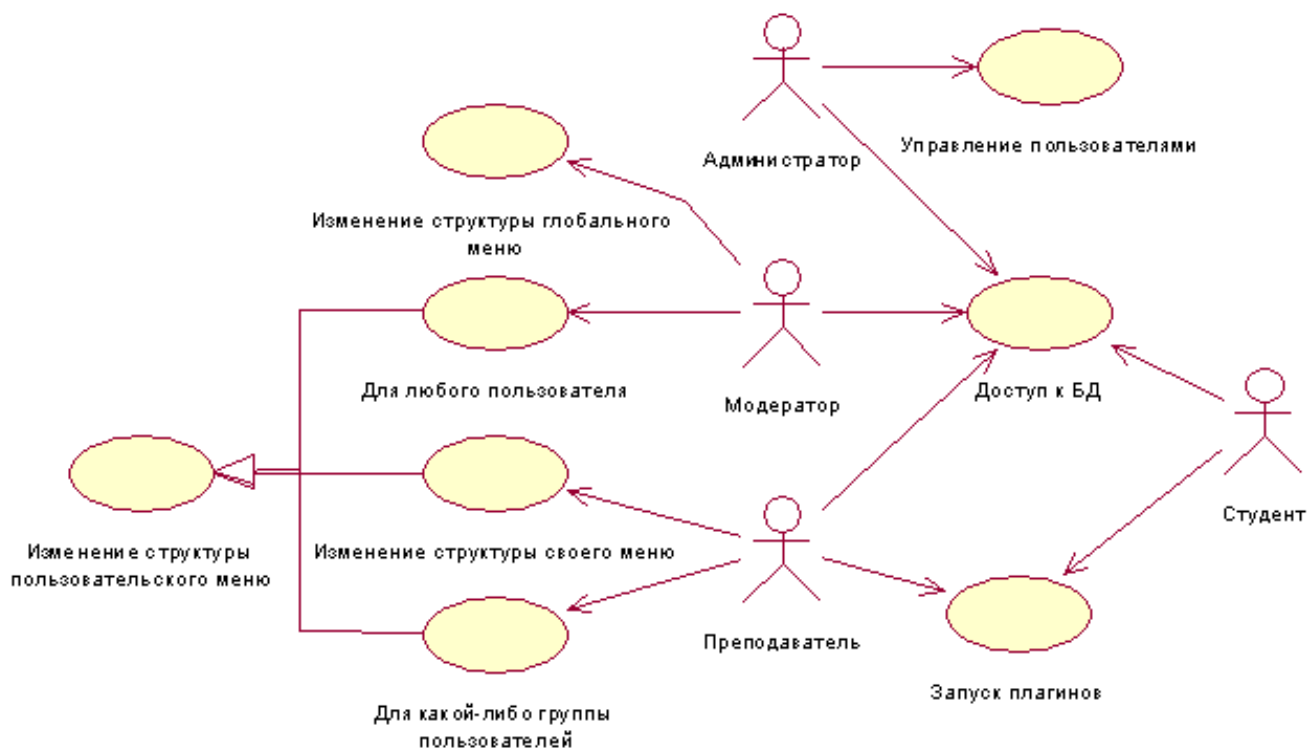


Рисунок 1 - Диаграмма прецедентов

2.2 Диаграмма внедрения

При создании АСУ была применена классическая Клиент-Серверная технология. При этом доступ к СУБД осуществляется посредством ODBC.

В системе одновременно могут работать несколько пользователей и администраторов, причем число подключений, фактически ограничено только возможностями сервера.

Диаграмма внедрения предназначена для отображения программных средств на узлы вычислительных систем. Вид диаграммы внедрения представлен на рисунке 2.

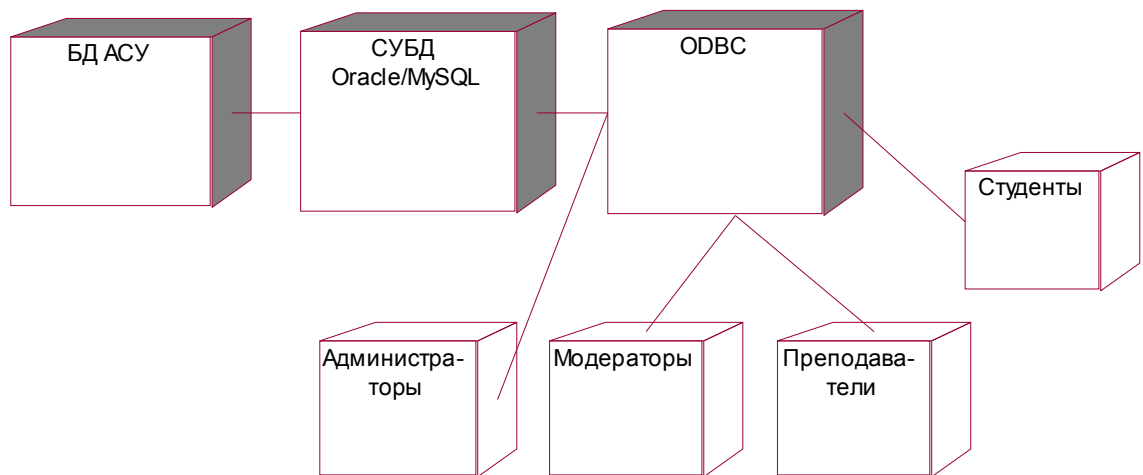


Рисунок 2 - Диаграмма внедрения

2.3 Классы АСУ

Исходя из задач АСУ и ее структуры, можно определить следующие три основополагающих класса АСУ:



Рисунок 3 - Классы АСУ

Здесь под пользователем понимается любой пользователь системы – в том числе модераторы. Фактически, вся система строится на взаимодействии этих трех классов: одни пользователи(администраторы) могут добавлять других, другие(модераторы) могут создавать различные структуры меню для конечных пользователей системы.

2.4 Диаграмма действий

Диаграмма действий(activity diagram) – это особое состояние конечного автомата, в котором все или большая часть состояний является состояниями деятельности или состояниями действия и в котором все, или большая часть переходов запускаются при завершении деятельности в исходных состояниях. На графе деятельности изображается поток работ или работа процедуры. Граф деятельности является завершенным блоком модели и отображается на диаграмме деятельности.

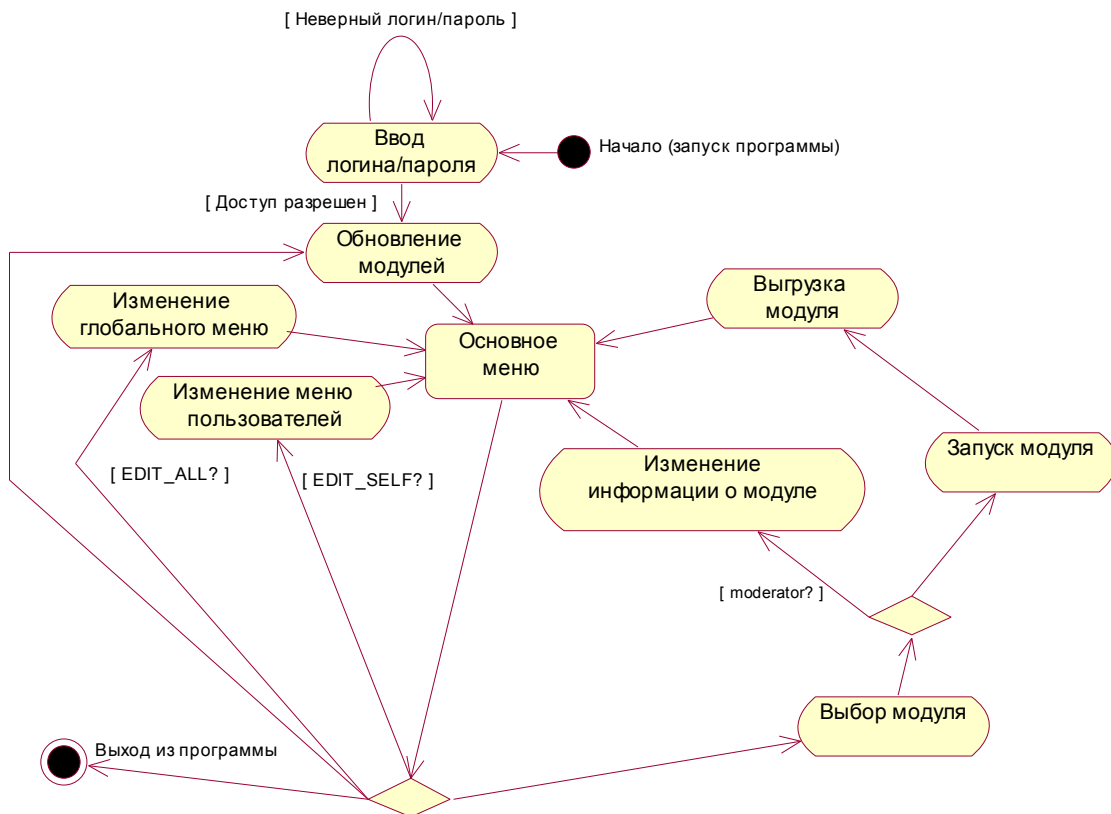


Рисунок 4 - Диаграмма действий в клиентском режиме

На рисунке 4 показана диаграмма действие при работе в клиентском режиме (модуль MENU). Войти в данный режим могут пользователи и модераторы, однако, как видно из диаграммы, набор доступных действий зависит от привилегий данного пользователя.

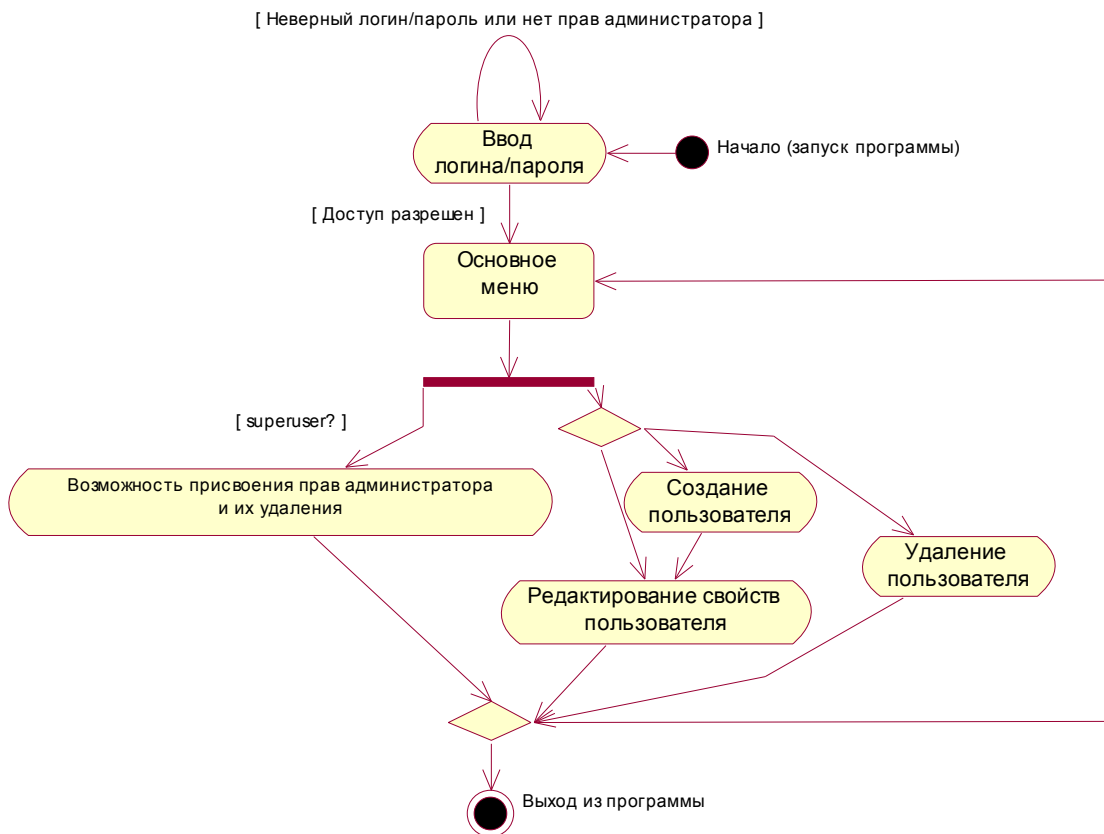


Рисунок 5 - Диаграмма действий режиме администрирования

На рисунке 5 показана модель действий администраторов. Как видно, администраторы могут создавать, редактировать и удалять пользователей, однако администраторы не могут производить никаких действий с учетными записями других администраторов. Такое право в системе имеют только администраторы с привилегией SuperUser.

3 Информационная модель АСУ

БД должна содержать следующие таблицы:

3.1 MENU

В таблице MENU хранится информация обо всех доступных плагинах системы, доступных в них функциях и параметрах запуска этих функций. Кроме того, таблица содержит дополнительные поля для построения дерева меню.

Структура таблицы представлена в табл. 1.

Таблица 1- Структура таблицы "MENU"

Поле	Тип	Null	Описание
MN_ID	mediumint(8) unsigned		Уникальный идентификатор
MD_ID	mediumint(8) unsigned	YES	Ссылка на модуль
FN_ID	mediumint(8) unsigned	YES	Ссылка на функцию
PR_ID	mediumint(8) unsigned	YES	Ссылка на параметр
MN_NAME	varchar(30)		Название пункта меню
MN_PARENT	mediumint(8) unsigned	YES	Ссылка на родительский пункт
MN_TYPE	tinyint(3) unsigned	YES	Тип модуля

3.2 MODULES

В таблице MODULES хранится информация о модулях системы. Структура таблицы представлена в табл. 2.

Таблица 2 - Структура таблицы "MODULES"

Поле	Тип	Null	Описание
MD_ID	mediumint(8) unsigned		Уникальный идентификатор
MD_NAME	varchar(30)		Название модуля
MD_CAPTION	varchar(64)	YES	Краткое описание
MD_DATE	date	YES	Дата
MD_AUTHOR	varchar(64)	YES	Автор
MD_DESCRIPTION	text	YES	Описание модуля

3.3 FUNCTIONS

В таблице FUNCTIONS хранится информация о функциях отдельных модулей. Структура таблицы представлена в табл. 3.

Таблица 3 - Структура таблицы "FUNCTIONS"

Поле	Тип	Null	Описание
FN_ID	mediumint(8) unsigned		Уникальный идентификатор
FN_MODULE	mediumint(8) unsigned	YES	Ссылка на модуль
FN_NAME	varchar(30)		Имя функции
FN_DESCRIPTION	text	YES	Описание функции

3.4 PARAMS

В таблице PARAMS хранится информация о параметрах, с которыми можно вызывать функции различных модулей.

Структура таблицы представлена в табл. 4.

Таблица 4 - Структура таблицы "PARAMS"

Поле	Тип	Null	Описание
PR_ID	mediumint(8) unsigned		Уникальный идентификатор
PR_FUNCTION	mediumint(8) unsigned	YES	Ссылка на функцию
PR_NAME	varchar(30)		Параметр
PR_DESCRIPTION	text	YES	Описание параметра

3.5 GROUPS

В таблице GROUPS хранится информация о группах пользователей и их привелегиях.

Структура таблицы представлена в табл. 5.

Таблица 5 - Структура таблицы "GROUPS"

Поле	Тип	Null	Описание
GR_ID	mediumint(8) unsigned		Уникальный идентификатор
GR_NAME	varchar(32)	YES	Имя группы
GR_EXEC	tinyint(1) unsigned		Право на запуск модулей
GR_EDITSELF	tinyint(1) unsigned		Право редактирования своего меню
GR_EDITALL	tinyint(1) unsigned		Право редактирования всех меню
GR_ADMIN	tinyint(1) unsigned		Право на создание и удаление пользователей с правами EXECUTE, EDITSELF или EDITALL, просмотр логов работы пользователей.
GR_SU	tinyint(1) unsigned		Право на создание и удаление пользователей с любыми правами. Полный доступ к БД.

Рекомендуется следующий набор групп:

Таблица 6 - Рекомендуемый набор групп

Название группы	Права				
	EXECUTE	EDITSELF	EDITALL	ADMIN	SU
USERS	X	-	-	-	-
POWERUSERS	X	X	-	-	-
MODERATORS	X	X	X	-	-
ADMINISTRATORS	-	-	-	X	-
SUPERUSERS	X	X	X	X	X

USERS – студенты; POWERUSERS - преподаватели

3.6 USERS

В таблице USERS хранится информация необходимая для проведения авторизации(login), права конкретного пользователя(admin/user), а также дополнительная информация о пользователях.

Структура таблицы представлена в табл. 7.

Таблица 7 - Структура таблицы "USERS"

Поле	Тип	Null	Описание
US_ID	int(11)		Уникальный идентификатор
GR_ID	mediumint(8) unsigned		ID группы
US_LOGIN	varchar(12)		Логин пользователя
US_NAME	varchar(70)	YES	Имя пользователя
US_NUMBER	varchar(7)	YES	Телефон
US_EMAIL	varchar(30)	YES	E-mail
US_BIRTHDAY	date	YES	День рождения
US_ADDITIONAL	text	YES	Дополнительная информация
US_BLOCK	char(1)	YES	Признак блокировки пользователя

3.7 USER_MENU

Фактически представляет из себя таблицу, осуществляющую связь МНОГИЕ-КО-МНОГИМ между таблицами MENU и USERS. Содержит структуру меню для конкретного пользователя.

Структура таблицы представлена в табл. 8

Таблица 8 - Структура таблицы "USER_MENU"

Поле	Тип	Null	Описание
USM_ID	int(10) unsigned		Уникальный идентификатор
US_ID	mediumint(8) unsigned		ID пользователя
MN_ID	mediumint(8) unsigned		ID пункта меню
USM_ORDER	smallint(5) unsigned	YES	Порядок пункта для пользователя
USM_NAME	varchar(80)	YES	Название пункта для пользователя

3.8 ER-модель

ER-модель БД представлена на рисунке 6.

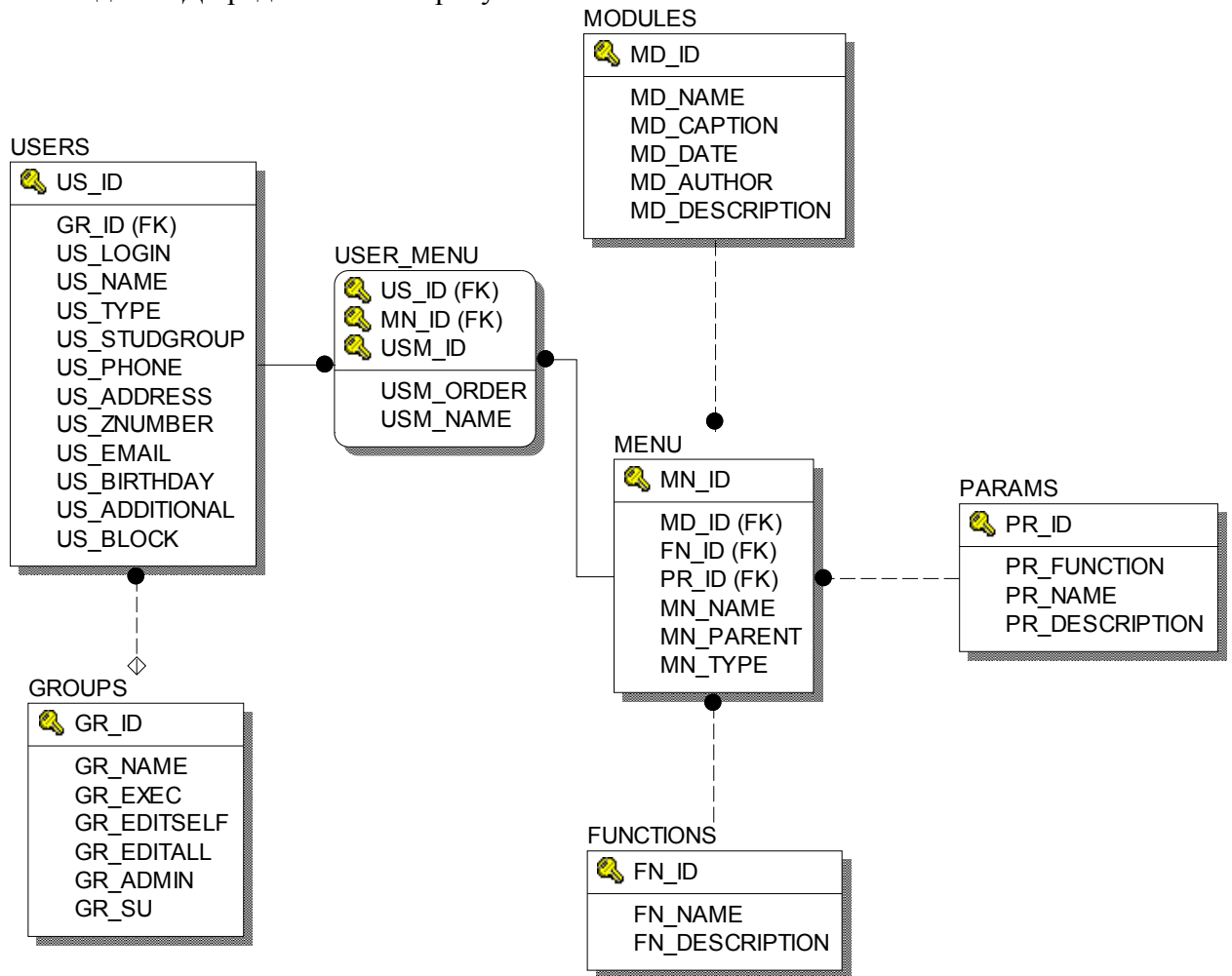


Рисунок 6 - ER-модель БД АСУ

4 Общая структура АСУ

АСУ состоит из 5-ти основных модулей:

1. Модуль "DB"

Модуль предназначен для создания подключения к БД и предоставления доступа к БД другим компонентам АСУ. Также, модуль содержит функции, позволяющие выполнять различные запросы, определять является ли пользователь администратором, менять пароли пользователей и создавать новых пользователей.

2. Модуль "Connect"

Данный модуль предназначен для обеспечения возможности подключения пользователей к БД АСУ. При этом вводятся имя пользователя и пароль.

3. Модуль "Menu"

Модуль предназначен для получения из БД структуры дерева программ для конкретного пользователя, ее отображения и запуска внешних модулей. Дополнительно в состав модуля включен браузер - аналог Windows Explorer с возможностью просматривать Интернет - контент. Браузер является стандартным объектом Delphi (TInternetBrowser), поэтому как отдельным модуль не рассматривается.

4. Модуль "Plugins"

Модуль сравнивает файлы с расширениями *.exe и *.dll в локальной и сетевой директориях, указанных в файле acs.ini и обновляет локальные файлы в случае необходимости(при отсутствии файлов в локальной директории, а также при несовпадении даты обновления файлов или их размера).

5. Модуль "Admin"

Модуль позволяет добавлять, удалять и изменять информацию в трех основных таблицах АСУ. Соответственно, модуль может работать в 3-х режимах:

- Пункты меню
- Пользователи
- Меню пользователей

Взаимосвязь модулей АСУ показана на рисунке 7.

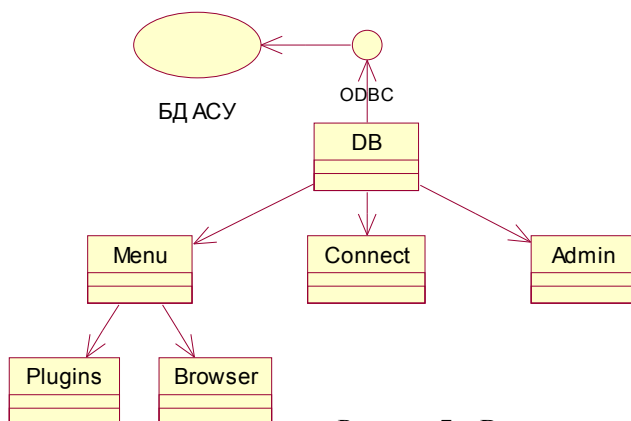


Рисунок 7 – Взаимосвязь модулей АСУ

5 Описание модулей АСУ

5.1 Модуль "DB"

Для взаимодействия с Базой данных АСУ в состав класса модуля включены 3 компонента:

1. ODBC (класс TADOConnection);
2. ADOQuery (класс TADOQuery);
3. ADOCommand (класс TADOCommand);

1-й компонент позволяет создавать соединение с БД, 2-й предназначен для выполнения SELECT – запросов, 3-й – для выполнения SQL – запросов, не возвращающих данные (INSERT, DELETE и т.д.).

Перечень процедур и функций модуля:

Процедуры:

Название: OpenQuery

Передаваемое значение: текст запроса(sqls: string)

Назначение: запуск SELECT – запросов

Название: ExecQuery

Передаваемое значение: текст запроса(sqls: string)

Назначение: запуск INSERT, DELETE, CREATE запросов

Название: SetPassword

Передаваемые значения: логин, пароль(login, pas: string)

Назначение: меняет пароль для пользователя <login> на <pass>

Название: CreateDBUser

Передаваемые значения: логин, пароль, флаг администратора(login,pas:string; admin:boolean)

Назначение: создает пользователя MySQL с логином и паролем, указанными в <login>/<pass>. Если флаг <admin> стоит в TRUE, то пользователю присваиваются права администратора: все права на БД ACS и право на UPDATE,CREATE и INSERT на системную БД.

Функции:

Название: UserIsAdmin

Передаваемое значение: логин, пароль(login, pas: string)

Возвращаемое значение: TRUE или FALSE (boolean)

Назначение: проверяет, является ли пользователь <login> администратором на основе информации из таблицы ACS.USERS

5.2 Модуль "Connect"

При запуске модуля появляется заставка, и отображается окно аутентификации пользователя(рисунке5). При этом заставка и окно аутентификации располагаются по центру экрана автоматически, в зависимости от разрешения.

Если при вводе пароля пользователь ошибается более трех раз, то программа закрывается.

При создании пользователя администратор может указать ему пустой пароль. В таком случае программа предложит пользователю ввести новый пароль(рисунке 6). Этот механизм можно применять также при создании новых пользователей, чтобы позволить им ввести пароль самим.

Для клиентской и администраторской части используются одинаковые модули Connect, однако в первом случае константу AllowUserLogin необходимо установить в TRUE, чтобы разрешить доступ пользователям, а во-втором, соответственно, в FALSE.

5.3 Модуль "Menu"

Модуль "Menu" – это основной модуль клиентской части.

После запуска вызывается в качестве модального окна модуль "Connect". В случае успеха аутентификации производится загрузка обновлений плагинов, после чего запускается основная форма приложения. Этот процесс можно представить следующим образом:



Загрузка меню

Загрузка меню осуществляется следующим образом: сначала выполняется запрос, получающий из БД все пункты меню для данного пользователя, для которых значение поля MN_PARENT равно NULL, то есть те, у которых нет предков.

Затем для каждого загруженного пункта проверяется поле MN_ISPARENT, т.е. пункт проверяется на наличие потомков и, если они есть, то в структуре дерева меню создается временный потомок, фактически являющийся меткой, по которой можно узнать, есть ли у данного пункта потомки. Затем, если пользователь откроет данную ветки меню, из БД будут автоматически подгружены потомки, причем для каждого из них будет проверяться условие, указанное выше. Таким образом обеспечивается практически неограниченная вложенность дерева меню.

5.4 Модуль "Plugins"

Обновление плагинов происходит на основе информации о местоположении локальной и сетевой директорий с плагинами. Эта информация находится в файле "acs.ini". Доступ к файлу осуществляется посредством стандартного объекта Delphi – TIniFile.

Если информация о местоположении локальной директории успешно прочитана, она передается в публичную переменную `plugdir` модуля "Menu" для дальнейшего использования.

Пути, содержащиеся в файле "acs.ini" можно поменять либо вручную, либо нажав соответствующую клавишу в ToolBar'e модуля "Menu". При это будет показана форма, содержащаяся в модуле "Plugins", и позволяющая изменить как локальный, так и сетевой путь к папкам с плагинами(рисунке 8). Аналогичное окно будет показано в случае, если файл "acs.ini" отсутствует или поврежден.

Алгоритм работы модуля представлен на рисунке 8.

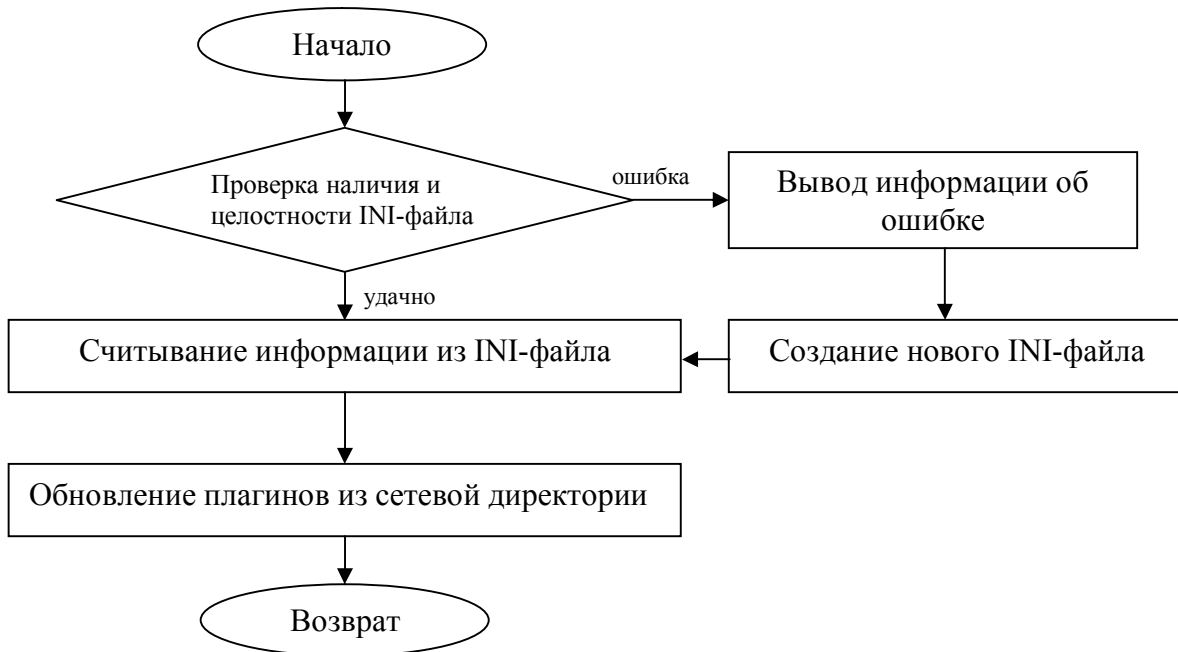


Рисунок 8 - Алгоритм работы модуля "Plugins"

5.5 Модуль "Admin"

Модуль может работать в 3-х режимах, каждый из которых позволяет изменять соответствующие таблицы. Переключение между режимами осуществляется при помощи стандартного компонента Delphi – TTabControl.

При этом в режиме "Меню пользователей" используются те же экземпляры компонентов, что и в предыдущих двух режимах(MenuTree и UsersGrid). При этом при выборе режима MenuTree просто смещается вправо.

Таким образом, взаимодействие между компонентами модуля и таблицами БД можно представить, как показано на рисунке 9.

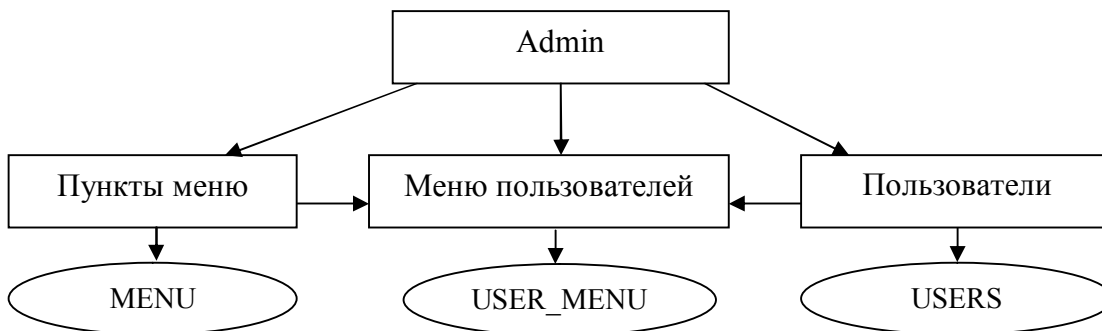


Рисунок 9 - Взаимодействие между компонентами модуля "Admin" и таблицами

Описание работы в различных режимах.

1. Пункты меню (рисунок 10)

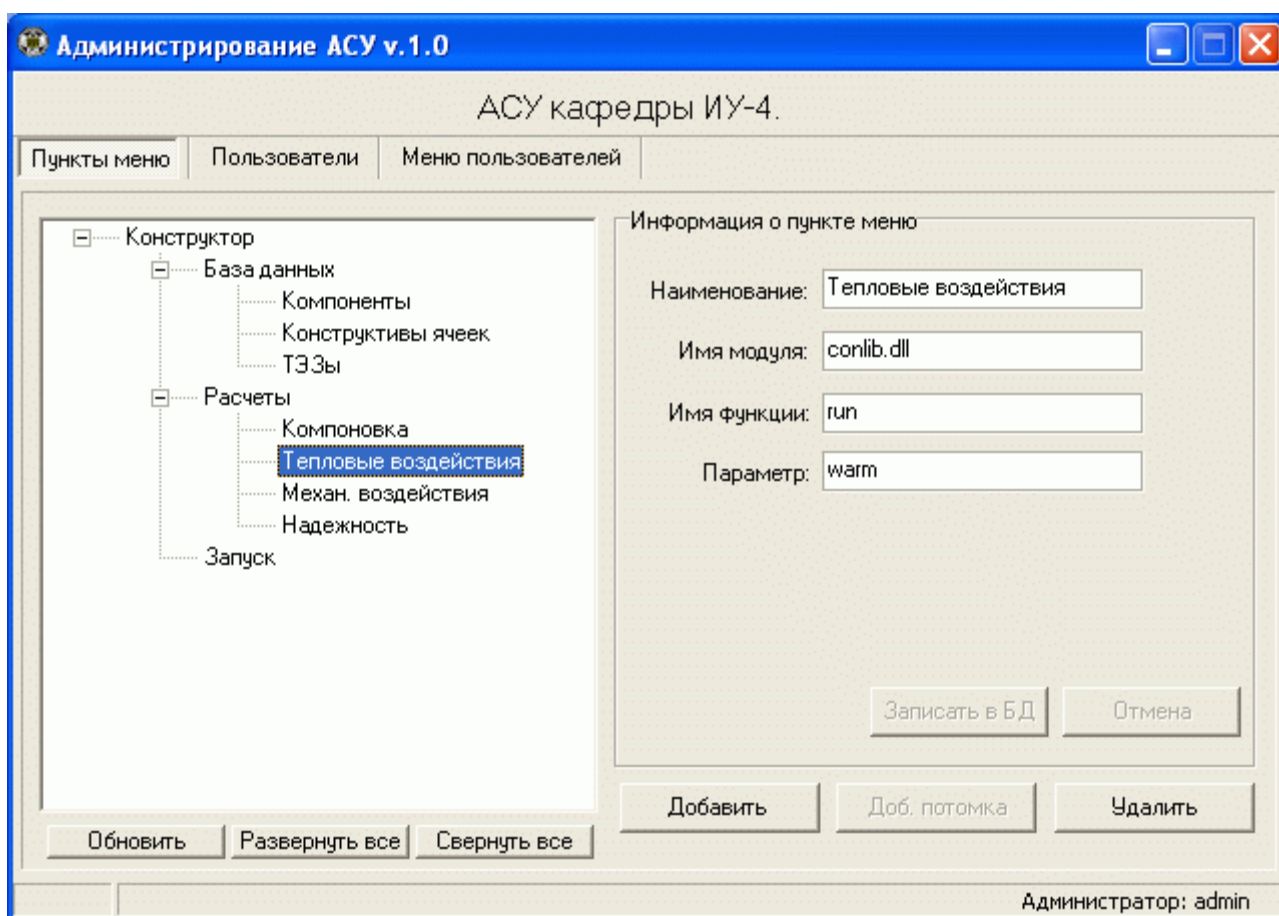


Рисунок 10 - Вид формы модуля "Admin" в режиме редактирования пунктов меню

Загрузка дерева меню из БД осуществляется динамически, по мере выбора отдельных ветвей.

Редактирование пунктов меню осуществляется при помощи обычных VCL(не VCLDB) - компонентов Delphi, поскольку в Delphi нет компонента TDBTreeView.

При выборе какого-либо пункта дерева меню, соответствующая информация появляется в правой части формы. Любой параметр можно поменять, при этом запись в БД произойдет только после нажатия соответствующей кнопки ('Записать в БД').

2. Пользователи (рисунок 11)

Используются компоненты VCLDB (кроме поля редактирования пароля).

Для повышения защищенности БД, пароли пользователей хранятся не в БД ACS, а только в системной таблице MYSQL. Кроме того, при отображении информации о пользователе в поле пароля за звездочками отображается не сам пароль, а некий стринг (TempPass), что практически исключает возможность получения паролей одних пользователей другими (ограничено защищенностью самой СУБД MySQL).

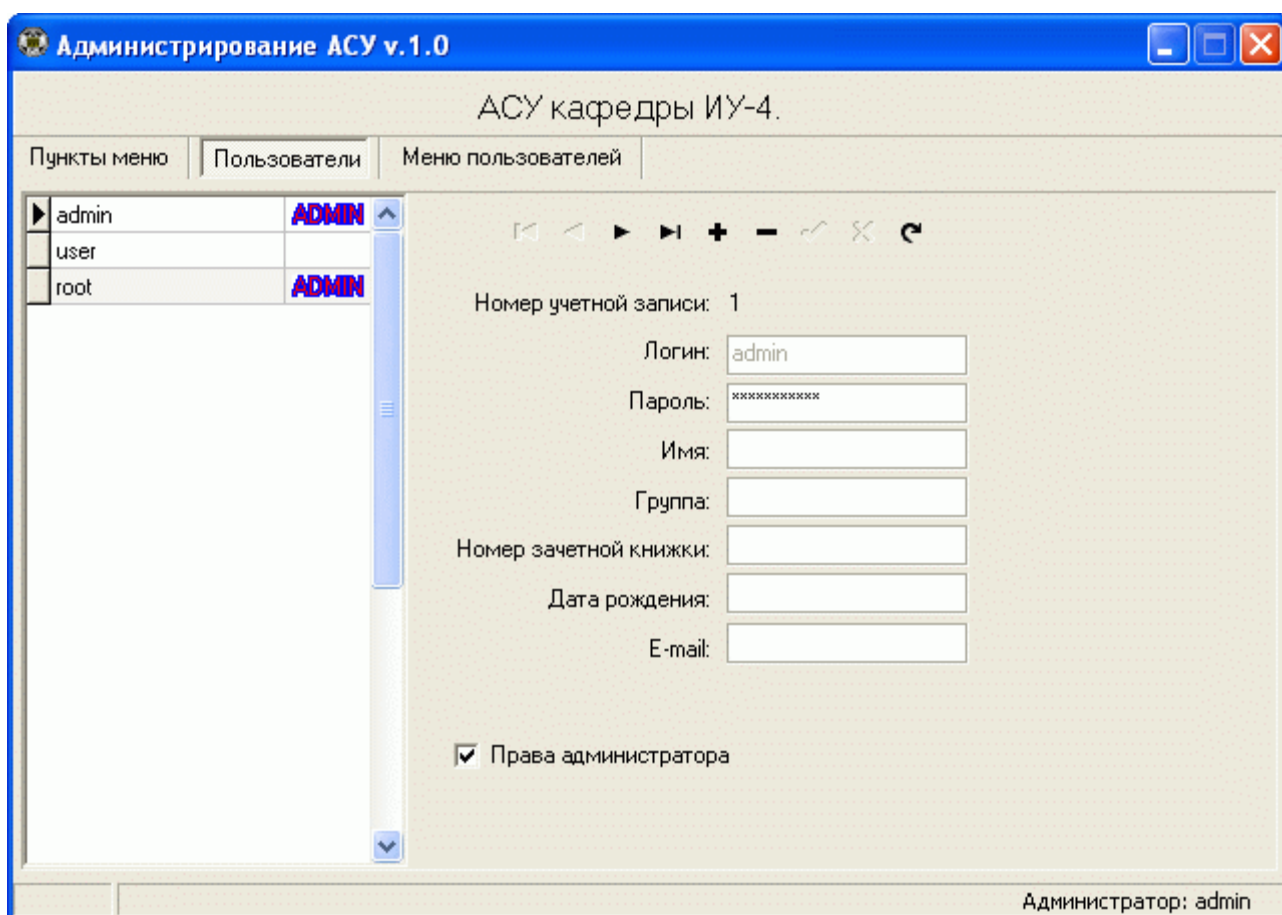


Рисунок 11- Вид формы модуля "Admin" в режиме редактирования учетных записей пользователей

3. Меню пользователей (рисунок 12)

Для выбора пунктов меню, доступных пользователям используется обычный CheckBox. При его установке/снятии флажка изменения записываются в БД автоматически. При выборе пункта, который является потомком другого пункта, родительский пункт выбирается автоматически.

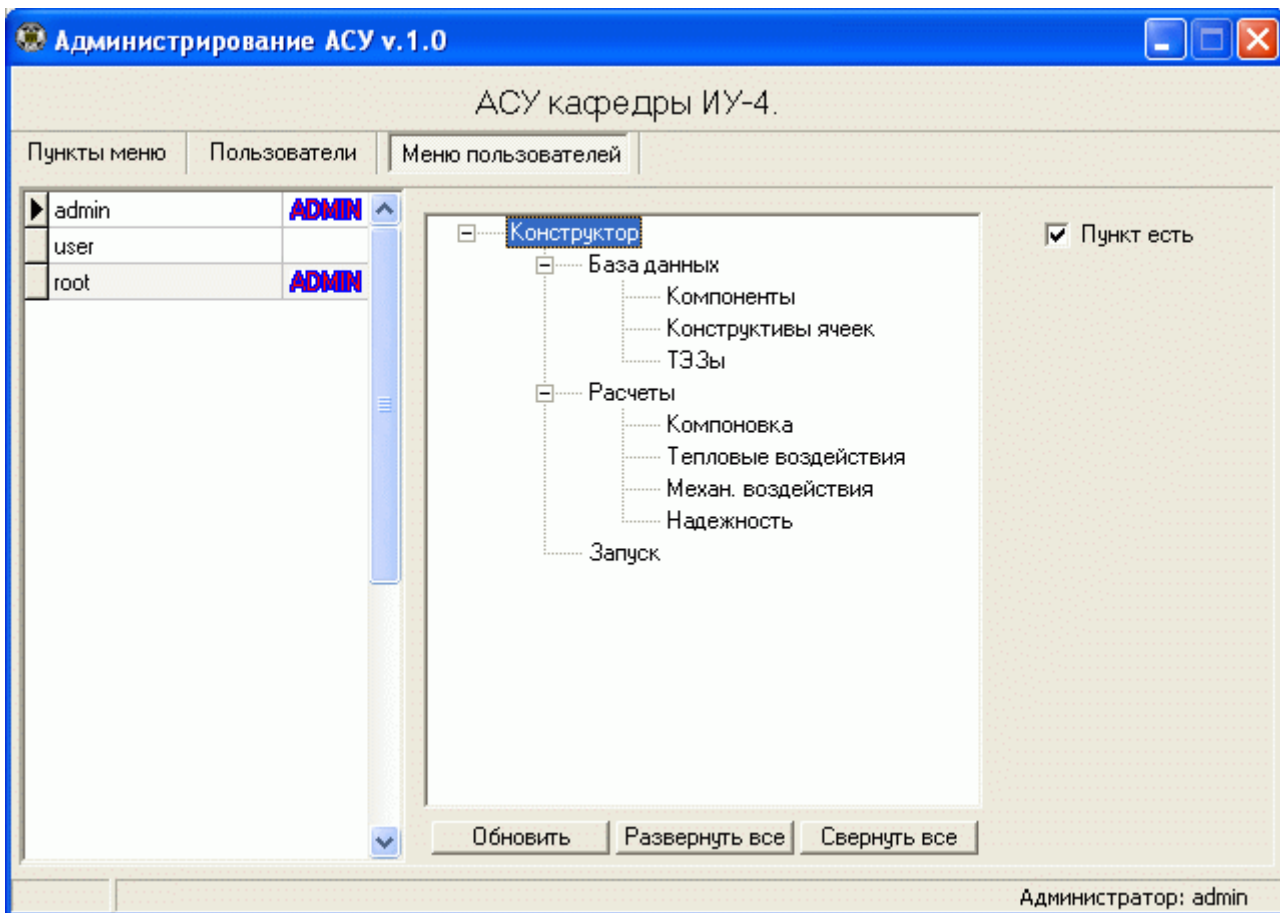


Рисунок 12 - Вид формы модуля "Admin" в режиме редактирования меню пользователей

6 Описание структуры плагинов

В данной системе плагины могут быть 2-х типов:

1. Обычные exe-файлы, имя запускаемой функции передается в качестве параметра.
2. Динамические библиотеки (dll) -должны иметь специфическую структуру, описание которой указано ниже.

Структура плагинов в виде dll

Плагины в dll должны экспортировать хотя бы одну функцию, которая может выполнять любые действия(например, создавать/отображать какую-нибудь форму). Этой функции также можно передать дополнительный параметр. Кроме того, при создании плагина нужно учитывать, что вызывающий его модуль(client) должен иметь возможность выгрузить dll по-окончании работы с плагином. Это реализуется с помощью передачи сообщения(WINDOWS MESSAGE) в форму client после закрытия главной формы dll.

При выходе из программы необходимо использовать событие OnCloseQuery, так как показано в примере ниже. Не позволяйте плагину самому закрывать свои окна! Это может привести к закрытию всех окон(в том числе и окна АСУ), а также к другим непредсказуемым последствиям.

Примерная структура библиотеки:

```
library mylib;
uses
  SysUtils, Classes, UfmTest in 'UfmTest.pas' {fmTest};
{$R *.res}
procedure init(PHandle,AHandle: THandle; StrParam: string); stdcall;
var id: longint;
begin
  fmTest.Init(PHandle,AHandle,StrParam);
  //PHandle - Handle вызывающей формы (client)
  //AHandle - Handle dll - нужно передать в client при закрытии dll
  //StrParam - Дополнительный строковый параметр

end;

exports
  Init;

begin
end.
```

Примерная структура формы в библиотеке:

```
unit UfmTest;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ComObj, ExtCtrls, Buttons;

const
  CM_DLLMUSTDIE = WM_USER + 1; //О п р е д е л я е м п е р е д а в а е м о е
  с о о б щ е н и е

type
  TfmTest = class(TForm)
  // С п и с о к с в о й с т в к л а с с а
    procedure Init(PHandle,AHandle: THandle; StrParam: string); stdcall;
  private
    paID, fmID: THandle;
  end;
var
  fmTest: TfmTest;

implementation
  {$R *.dfm}
  procedure TfmTest.Init(PHandle,AHandle: THandle; StrParam: string);
  var s: string;
  begin
    fmTest := TfmTest.Create(Application);
    fmTest.Show;
    //С о х р а н я е м H a n d l e s в г л о б а л ь н ы х п е р е м е н н ы х
    fmTest.paID := PHandle;
    fmTest.fmID := AHandle;
  end;

  procedure TfmTest.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
  begin
    CanClose := FALSE;
    //П е р е д а е м г л а в н о й п р о г р а м м е с о о б щ е н и е - б и б л и о т е к у
    м о ж н о в ы г р у ж а т ь
    PostMessage(PID,CM_DLLMUSTDIE,AID,0);
  end;
end.
```

7 Системные требования

Минимальные системные требования:

- Процессор: iPentium или совместимый с частотой не менее 100 МГц
- ОЗУ: не менее 32 Мб.
- Мб свободного дискового пространства(без плагинов).
- ОС Windows.

Рекомендуется использовать АСУ на компьютерах с >64 Мб ОЗУ и с установленной ОС Windows из семейства NT. В противном случае из-за особенностей Borland Delphi не гарантируется обеспечение вложенности дерева меню более 2-х.