

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

СОГЛАСОВАНО

Генеральный директор  
ЗАО «АйТи»

  
Бакиев О.Р.  
"23" сентября 2011 г.

УТВЕРЖДАЮ

Ректор НИУ ИТМО

  
Васильев В.Н.  
2011 г.

МНОГОПРОФИЛЬНАЯ ИНСТРУМЕНТАЛЬНО-  
ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА СОЗДАНИЯ  
И УПРАВЛЕНИЯ РАСПРЕДЕЛЕННОЙ СРЕДОЙ  
ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ CLAVIRE

ПРОГРАММНЫЙ КОМПОНЕНТ ИНФОРМАЦИОННОГО  
ПОРТАЛА CLAVIRE/PORTAL

ОПИСАНИЕ ПРОГРАММЫ

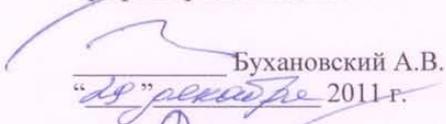
ЛИСТ УТВЕРЖДЕНИЯ

RU.СНАБ.80066-06 13 31-ЛУ

Ине.№ подл.	Подп. и дата
Взам.ине.№	Подп. и дата
Ине.№ дубл.	Подп. и дата
Ине.№ подл.	Подп. и дата

Представители  
Организации-разработчика

Руководитель разработки,  
профессор НИУ ИТМО

  
Бухановский А.В.  
"23" сентября 2011 г.

Ответственный исполнитель,  
с.н.с. НИУ ИТМО

  
Луценко А.Е.  
"23" сентября 2011 г.

Нормоконтролер  
ведущий инженер НИУ ИТМО

  
Позднякова Л.Г.  
"23" сентября 2011 г.

2011

УТВЕРЖДЕН  
RU.СНАБ.80066-06 13 31-ЛУ

**МНОГОПРОФИЛЬНАЯ ИНСТРУМЕНТАЛЬНО-  
ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА СОЗДАНИЯ  
И УПРАВЛЕНИЯ РАСПРЕДЕЛЕННОЙ СРЕДОЙ  
ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ CLAVIRE**

**ПРОГРАММНЫЙ КОМПОНЕНТ ИНФОРМАЦИОННОГО  
ПОРТАЛА CLAVIRE/PORTAL**

**ОПИСАНИЕ ПРОГРАММЫ**

RU.СНАБ.80066-06 13 31

ЛИСТОВ 24

2011

Инв.№ подл.	Подп. и дата	Взам. инв.№	Инв.№ дубл.	Подп. и дата

## **АННОТАЦИЯ**

Программный компонент информационного портала CLAVIRE/Portal RU.СНАБ.80066-06 01 31 предназначен для информационной поддержки пользователей МИТП, организации высокоуровневого сотрудничества в рамках виртуального профессионального сообщества (в рамках центров компетенции), а также интеграции базовых функциональных элементов управления МИТП в составе единого информационно-программного web-решения. Настоящая версия информационного портала включена в состав макета МИТП, разработанного на этапе эскизного проекта, и предназначена для анализа базовых подходов и функциональных возможностей, предполагаемых к реализации в составе МИТП.

## СОДЕРЖАНИЕ

1.	ОБЩИЕ СВЕДЕНИЯ .....	4
2.	ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ .....	4
3.	ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ .....	5
3.1.	Принципы функционирования и программная архитектура .....	5
3.2.	Описание исполняемых модулей представления системы управления контентом..	11
3.3.	Основные классы компонента.....	15
3.3.1.	Класс mod_auth .....	15
3.3.2.	Класс mod_config_ws.....	16
3.3.3.	Класс mod_db .....	17
3.3.4.	Класс mod_safe .....	17
3.3.5.	Класс mod_content.....	18
4.	ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА .....	19
5.	ВЫЗОВ И ЗАГРУЗКА.....	20
6.	ВХОДНЫЕ ДАННЫЕ.....	21
7.	ВЫХОДНЫЕ ДАННЫЕ .....	22
	ПЕРЕЧЕНЬ СОКРАЩЕНИЙ .....	23

## 1. ОБЩИЕ СВЕДЕНИЯ

Программный компонент информационного портала CLAVIRE/Portal RU.СНАБ.80066-06 01 31 предназначен для информационной поддержки пользователей МИТП, организации высокоуровневого сотрудничества в рамках виртуального профессионального сообщества (в рамках центров компетенции), а также предоставления доступа к интерфейсу МИТП средствами компонента CLAVIRE/Ginger. Наиболее значимую роль данный компонент играет при необходимости поддержки открытого доступа к возможностям МИТП, коллаборативной работы, а также высокоуровневой интеллектуальной и информационной поддержки. В первую очередь такой подход востребован в рамках технологических платформ МИТП-Ц и МИТП-Д, отличающихся высокой степенью открытости и широкими возможностями по интеграции разнородных ресурсов (как вычислительных, так и информационных).

## 2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

Основное функциональное назначение компонента информационного портала на данном этапе – организация интеллектуальной информационной и технологической поддержки пользователя в процессе функционирования МИТП. Компонент позволяет решать следующие задачи:

- 1) создание условий для поиска и изучения материалов по использованию возможностей доступных сервисов, изложенных в доступной форме в виде информационных тематических блоков;
- 2) обеспечение возможности консультации с экспертами в предметной области применения МИТП;
- 3) поддержка виртуального профессионального сообщества в области знаний, для которой используется МИТП;
- 4) предоставление интерактивной справки по вычислительным ресурсам и сервисам, входящим в состав МИТП.

При этом, представляя собой один из компонентов базового ядра МИТП, информационный портал выступает в качестве элемента отдельных вариантов реализации платформы.

- В составе МИТП-К, МИТП-М и МИТП-Э информационный портал играет второстепенную роль, выполняя лишь базовые функции информационной поддержки и интеграции управляющего интерфейса МИТП.
- В составе МИТП-Ц информационный портал приобретает большое значение в качестве платформы функционирования виртуального профессионального сообщества пользователей МИТП: экспертов, модераторов, привилегированных и непривилегированных пользователей. При этом существенную значимость приобретают информационная поддержка пользователей с различным уровнем компетентности, а также возможности взаимодействия пользователей друг с другом, проведения совместных исследований и пр.
- В составе МИТП-Д информационный портал играет важную роль, позволяя централизованно управлять источниками данных, анализируемыми системными сервисами данной платформы при сборе информации. Поскольку во многих случаях большинство источников представляют собой децентрализованные информационные структуры в сети Интернет (социальные сети, новостные ленты и пр.), обеспечение возможности их интеграции, мониторинга, оповещения пользователей о возникновении определенных событий, а также координации процесса сбора информации становится важной задачей, решаемой порталом.

### **3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ**

#### **3.1. Принципы функционирования и программная архитектура**

Ядром программного компонента является система управления содержимым (CMS), разработанная с использованием архитектуры MVC (Model–View–Controller, Модель–Представление–Контроллер). Такая архитектура позволяет:

- 1) организовать повторное использование кода, создавая различные интерфейсы для доступа к функциональности приложений. Это важно, например, в случае работы с web-сервисами;
- 2) реализовать дополнительные интерфейсы доступа к сторонним web-ресурсам при помощи фреймов HTML или к web-сервисам посредством обмена данными по протоколу SOAP к существующему поисковому сервису (эта задача сводится к описанию нового формата представления, а также созданию объекта, отвечающего за вывод результатов в новом формате, при этом не требуется каких-либо изменений

существующей функциональности, а весь процесс отличается прозрачностью и простотой реализации);

- 3) изменять любую часть кода, не затрагивая при этом верстку, или изменять шаблоны без изменения программной части.

Архитектура включает следующие блоки.

- 1) *Модель* – определяет бизнес-логику, содержит программный код, определяющий типы данных, их свойства и взаимосвязи, а также дополнительные модули, содержащие набор функций, которые позволяют сайту работать с базой данных посредством SQL-запросов, а также с web-сервисами при помощи SOAP-запросов.
- 2) *Контроллер* – формирует логику ввода, интерпретирует данные, введенные пользователем, манипулирует сущностями Модели так, чтобы их можно было превратить в Представление.
- 3) *Представление* – определяет логику интерфейса, генерирует HTML-страницы портала из данных, полученных от Модели, интерпретированных Контроллером на основе запросов пользователя. К Представлению относятся шаблоны страниц, файлы стилей, Javascript-сценарии, приложения Flash и Silverlight.

Благодаря такой архитектуре логика ввода, бизнес-логика и логика интерфейса разделяются, но достаточно тесно взаимодействуют друг с другом. Это разделение позволяет работать со сложными структурами при создании программного компонента, так как обеспечивает одновременную реализацию только одного аспекта.

Типовую процедуру взаимодействия пользователя с интерфейсом можно описать следующим образом (рис. 3.1).

- Пользователь обращается к ресурсу через браузер, после чего браузер отправляет с помощью метода GET или POST, обрабатываемый Контроллером запрос, который, в свою очередь, обращается к Модели.
- Модель запрашивает данные у сервера базы данных с помощью SQL-запросов или у web-сервисов при помощи SOAP-запросов.

Получив необходимые данные через Модель, Контроллер выводит нужную информацию пользователю через Представление, генерируя HTML-страницу.

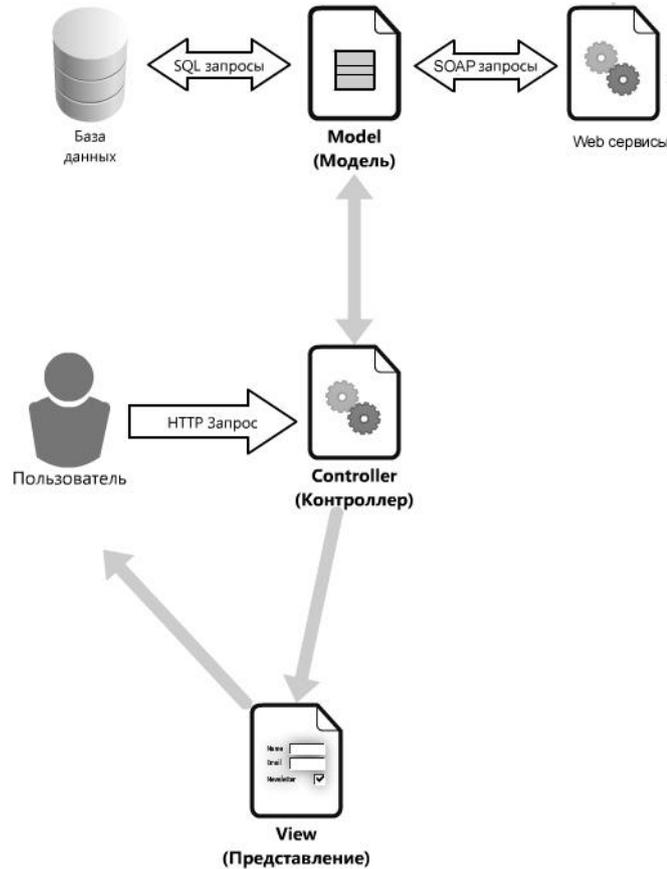


Рисунок 3.1 – Архитектура системы управления контентом

Система управления содержимым информационного портала имеет спроектированную блочную схему. Это обстоятельство позволяет, не внедряясь в ядро системы, разрабатывать новую функциональность в виде новых блоков. Количество блоков не ограничено. Система разработана таким образом, что в случае модернизации и добавления функциональности портала любой дополнительно разработанный блок может быть оперативно в нее установлен.

Базовым элементом информационного портала является услуга (сервис), предоставляемая его пользователям. Портал в этом случае представляет собой совокупность связанных сервисов.

Для реализации различных типов взаимодействия сервисов предусмотрены различные формы представления результатов работы пользователю. Ниже перечислены основные формы блока Представление для информационных сервисов. На рис. 3.2 приведены диаграммы, иллюстрирующие использование перечисляемых форм представления в составе страниц портала.

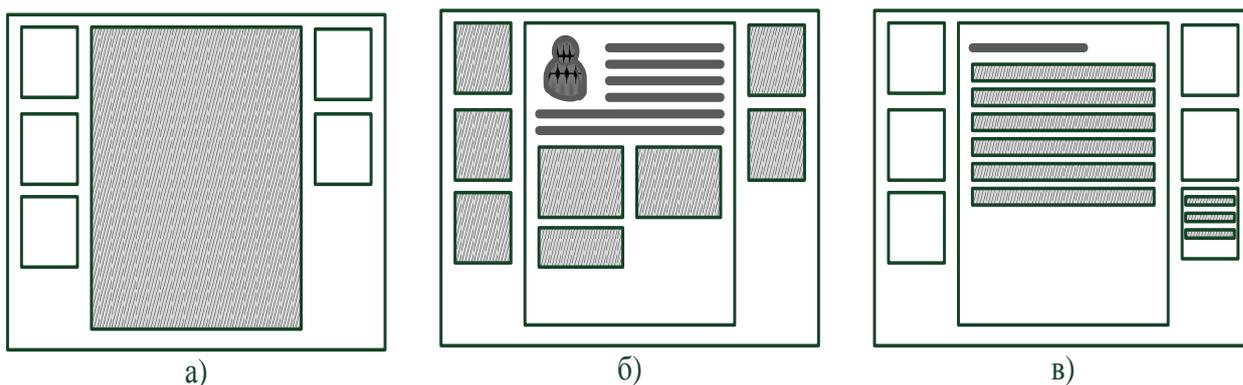


Рисунок 3.2 – Формы представления информационных сервисов

- 1) *Развернутое* (рис. 3.2а) – наиболее полное представление информации, доступной посредством сервиса. В составе просматриваемой пользователем страницы портала основную часть составляет развернутое представление, оно может включать в себя другие формы представления для данных, полученных от используемых сервисов.
- 2) *Сокращенное* (рис. 3.2б) – компактный вариант вывода информации, предназначенный для встраивания как в базовую структуру страницы портала, так и в развернутое представление других сервисов.
- 3) *Списочное* (рис. 3.2в) – предназначено для вывода краткой ссылочной информации в составе списков однотипных элементов (статей, дипломов, проектов и т.п.). Такие элементы могут быть использованы в составе других форм представления (списков публикаций, новостей и пр.).

Структура портала представлена несколькими группами, каждая содержит набор сервисов, в данном случае: хранения; мониторинга; вычислительные и персональные сервисы. Пример системы базовых сервисов портала приведен на рис. 3.3.

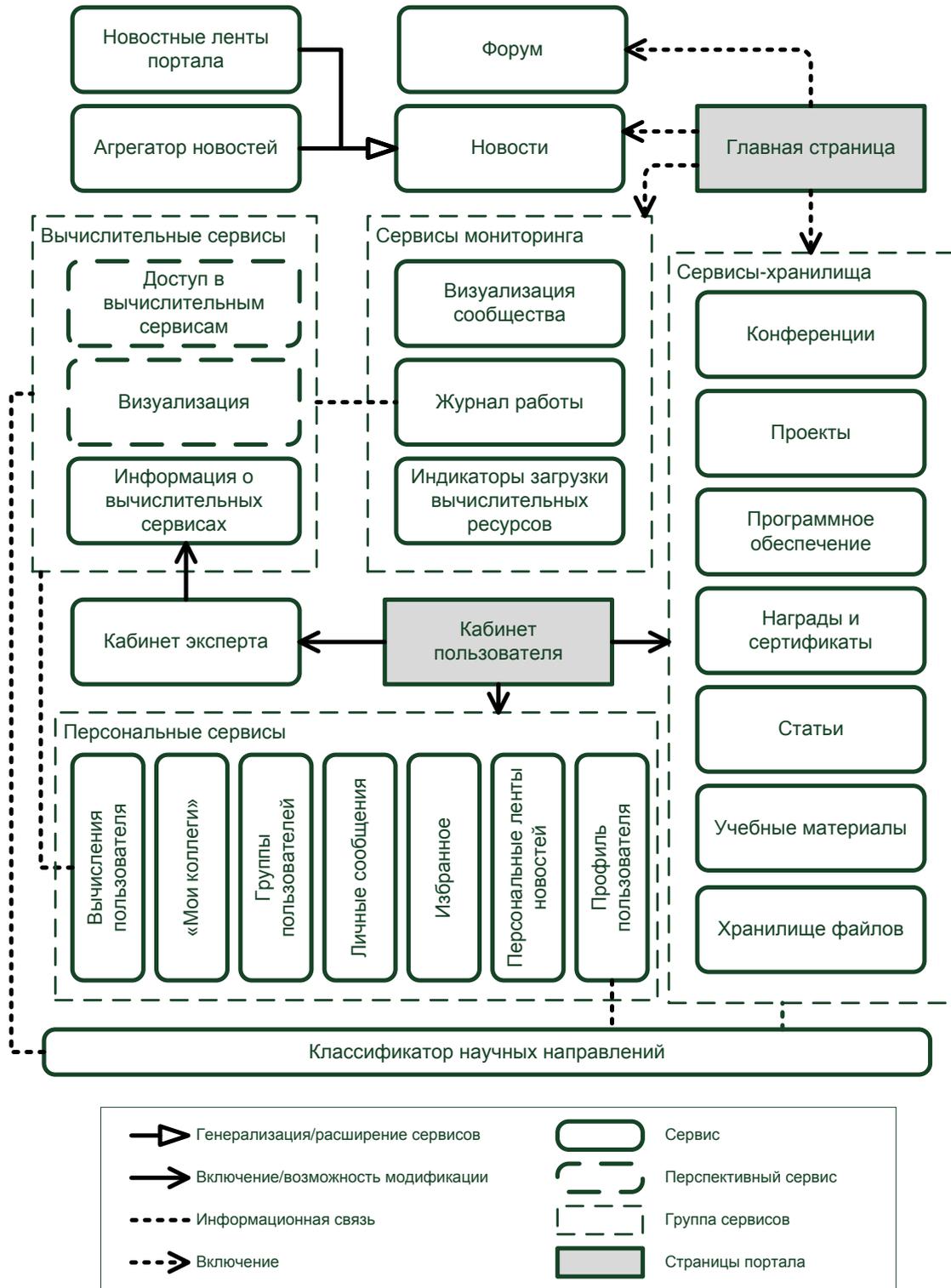


Рисунок 3.3 – Система базовых сервисов портала

Сервисы управляются блоком Контроллер, а для хранения и получения внешних данных используют блок Модель.

**Сервисы хранения.** Представляют собой интерфейс доступа к массивам однотипных данных: статей, сведений о конференциях, дистрибутивов программных средств и пр. Предоставляя возможность каждому из пользователей работать со своим

собственным набором объектов, сервисы этой группы в то же время обеспечивают доступ ко всей базе аналогичных объектов. При этом пользователю предоставляется возможность классификации объектов как по ключевым словам, так и с использованием сервиса классификатора научных направлений. Последний в этом случае представляет собой иерархическую структуру, служащую основой для группировки объектов заданного класса. Для каждого из объектов классификации можно рассматривать: сервис доступа к отдельным статьям, ориентированный, в первую очередь, на развернутую форму представления, и сервис каталогизации, использующий табличное представление предыдущего сервиса для агрегации информации обо всей базе статей (с возможной фильтрацией).

**Персональные сервисы** – доступны, они соответствуют основным сценариям поведения пользователей. Доступ к сервисам этой группы осуществляется посредством меню личного кабинета пользователя. В зависимости от режима доступа к конкретному сервису форма его представления может быть различной.

**Сервисы мониторинга** предназначены для отслеживания процесса выполнения собственных задач, а также для демонстрации активности портала и вычислительного комплекса. С помощью данных сервисов собирается статистика, ведутся журналы работы пользователей с ресурсами.

**Вычислительные сервисы** предназначены для поддержки пользователя в процессе работы с вычислительными ресурсами. Представлены информационные сервисы, предоставляющие аккумулированные знания о процедурах использования комплекса в процессе решения конкретных задач, а также сервисы интеграции клиентского интерфейса программного комплекса и средства визуализации.

Предусмотрена работа в системе в следующих пользовательских режимах.

- 1) **Гостевой вход** – разрешен доступ на портал для ограниченного просмотра содержимого: только к разделам с общей или ознакомительной информацией, чтобы составить представление о назначении ресурса, а также зарегистрироваться, чтобы получить доступ ко всем разделам.
- 2) **Режим пользователя** – разрешен доступ на портал для просмотра всех разделов, содержащих публичную информацию, а также доступ к частным данным в случае разрешения их владельца. Кроме того, пользователь может публиковать свои данные на портале, а также осуществлять доступ к пользовательским сервисам.
- 3) **Режим администратора** – разрешен доступ на портал для просмотра и редактирования содержимого всех разделов, содержащих публичную информацию,

также существует возможность корректировать и проводить ревизию материалов, публикуемых пользователями портала, управлять учетными записями пользователей, просматривать статистику по посещениям портала и использованию пользовательских сервисов.

- 4) **Режим разработчика** – разрешен доступ к техническим разделам портала: коду, шаблонам, таблицам базы данных. Этот вариант доступа предназначен только для квалифицированного персонала, обслуживающего ресурс: программистов, специалистов по подготовке шаблонов верстки, специалистов по продвижению портала в сети Интернет, администраторов базы данных.

### **3.2. Описание исполняемых модулей представления системы управления контентом**

В текущей реализации компонента информационного портала используется десять модулей представления системы управления контентом. Данные модули обеспечивают исполнение технологического функционала.

- 1) *Модуль отображения сохраненных данных с версткой.* Предоставляет функционал для вывода данных (форматированных, с сохранением верстки), сохраненных в базе данных.
- 2) *Модуль работы с документацией.* Предоставляет функционал доступа к сохраненной документации, разграничивает доступ для различных категорий пользователей. Поддерживает вывод разных форматов документов и видов ее представления.
- 3) *Модуль работы с материалами для самостоятельного тестирования.* Предоставляет функционал для создания материалов с целью самотестирования одной категории пользователей, а также с целью предоставления доступа к этим материалам другой категории пользователей (рис. 3.4).



Рисунок 3.4 – Пример интерфейса модуля работы с материалами для самостоятельного тестирования

- 4) *Модуль работы с учебными материалами.* Предоставляет функционал для создания учебных материалов одной категории пользователей, а также для предоставления доступа к этим материалам другой категории пользователей.
- 5) *Модуль работы с учебными модулями.* Предоставляет функционал для создания учебных модулей с возможностью автоматического конвертирования данных модулей в формат SCORM.
- 6) *Модуль электронных тьюторов.* Предоставляет функционал для демонстрации на портале электронных тьюторов в формате Flash.
- 7) *Модуль авторизации в системе.* Предоставляет функционал для авторизации пользователя на сайте. В качестве входных параметров предоставляются параметры учетной записи пользователя (имя пользователя и пароль), после чего происходит запрос к базе данных о существовании учетной записи с такими параметрами и получение сведения о правах пользователя в системе. В случае совпадения пары «логин–пароль», а также наличия у пользователя прав администратора системы происходит успешная авторизация и предоставляется доступ к интерфейсу администратора. Авторизованному пользователю данный модуль позволяет выйти из своей учетной записи.
- 8) *Модуль запуска инструмента коллективной работы.* Инструмент коллективной работы представляет собой репозиторий аннотированных композитных приложений на основе сервисов многопрофильной инструментально-технологической

платформы, предназначенных для решения конкретных прикладных задач. Предоставляет функционал для запуска инструмента коллективной работы с удаленного web-сервера посредством отображения во всплывающем фрейме содержимого данного ресурса (рис. 3.5), представленного компонентом CLAVIRE/Ginger.

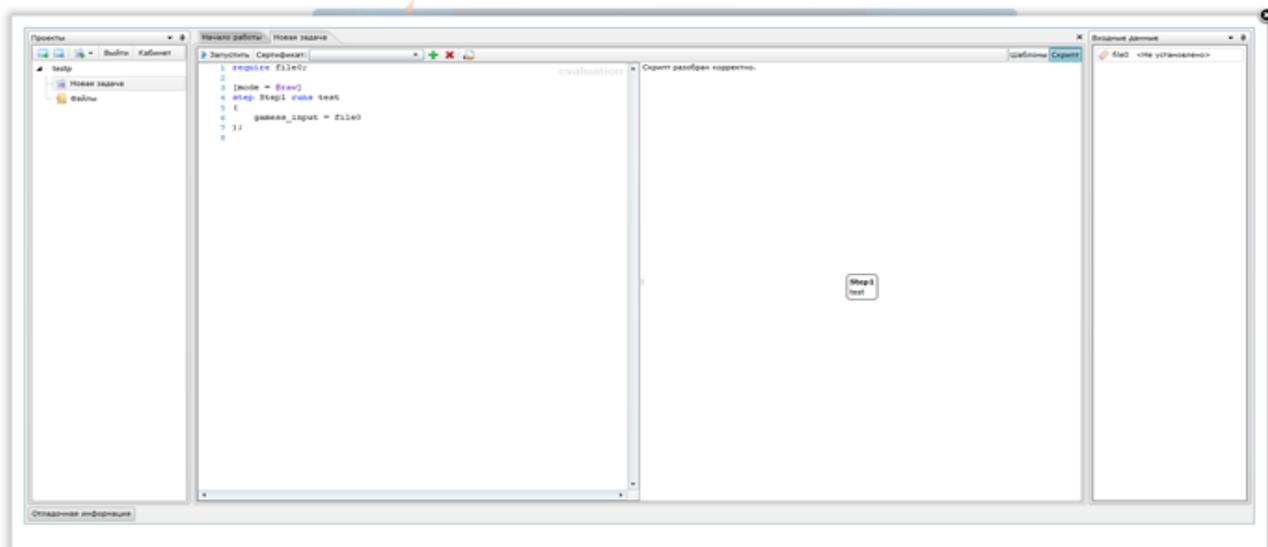


Рисунок 3.5 – Пример интерфейса доступа к компоненту CLAVIRE/Ginger

Схема работы следующая: один пользователь размещает свою работу (проект) в репозитории для обеспечения к ней доступа со стороны других пользователей (рис.3.6).

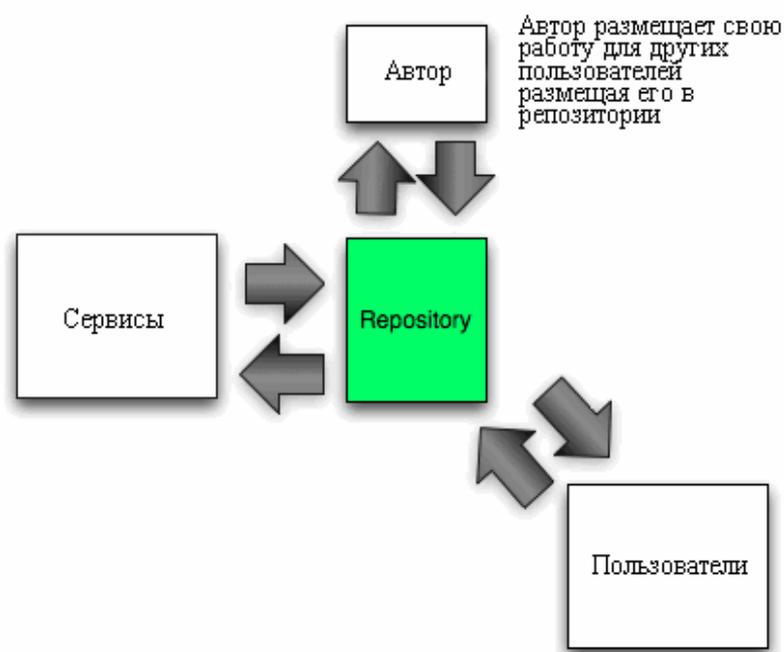


Рисунок 3.6 – Пример интерфейса модуля профессионального сообщества

- 9) *Модуль социальной сети.* Предоставляет функционал для отображения схемы социальной сети, ее представителей и связей между ними в виде анимации.
- 10) *Модуль виртуального профессионального сообщества.* Предоставляет функционал для создания, редактирования и просмотра профилей представителей сообщества, а также обмена сообщениями и материалами между ними (рис. 3.7).

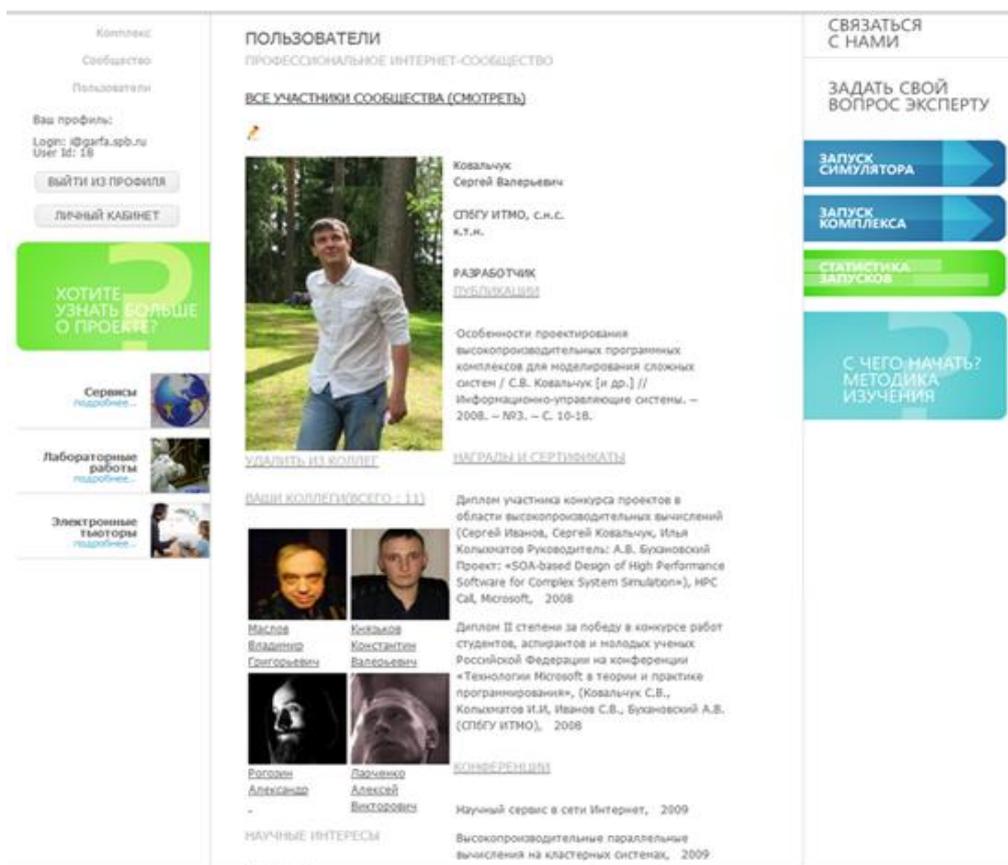


Рисунок 3.7 – Пример интерфейса модуля профессионального сообщества

### 3.3. Основные классы компонента

Ниже приводятся сокращенные описания структуры и методов основных классов компонента информационного портала.

#### 3.3.1. Класс *mod\_auth*

Реализация процесса авторизации пользователей в системе, а также механизмов, связанных с созданием параметров учетных записей пользователей.

##### Открытые методы

- `rand_symb` – генерирует случайный символ
  - a) входной параметр `type` (тип: `string`) – тип символа;
  - b) возвращает значение символа (`string`).
- `randomize_string_lenght` – генерирует случайный набор символов
  - a) входной параметр `from` (тип: `int`) – случайное число символов минимальное;
  - b) входной параметр `to` (тип: `int`) – случайное число символов максимальное;
  - c) возвращает значение строки (`string`).

- `mod_auth_gen_code` – генерация уникального случайного кода, идентифицирующего пользователя, который пытается авторизоваться через форму на портале
  - a) возвращает код (тип: string).
- `mod_auth_start` – механизм входа пользователя в систему (авторизация)
  - a) входной параметр `login` (тип: string) – имя пользователя;
  - b) входной параметр `pass` (тип: string) – пароль пользователя;
  - c) входной параметр `code` (тип: string) – уникальный код пользователя;
  - d) входной параметр `remember` (тип: int) – ключ сохранения имя пользователя и пароля в системе;
  - e) возвращаемое значение отсутствует (void).
- `mod_auth_stop` – механизм выхода пользователя из системы (деактивация)
  - a) возвращаемое значение отсутствует (void).
- `mod_user_audit` – механизм протоколирования действий пользователя в системе
  - a) входной параметр `action` (тип: string) – действие;
  - b) возвращаемое значение отсутствует (void).
- `is_ie` – определение типа браузера
  - a) возвращает код браузера (int).
- `is_mobile` – определение типа устройства (мобильное или стационарное)
  - a) возвращает код типа устройства (int).

### 3.3.2. Класс `mod_config_ws`

Реализация исполняемых модулей представления системы.

#### Открытые методы

- `select_ws_engine_rename` – возвращает идентификатор текущего модуля представления системы по наименованию
  - a) входной параметр `ws` (тип: string) – наименование текущего модуля представления системы;
  - b) возвращает идентификатор исполняемого модуля представления системы (int).
- `select_ws_engine` – возвращает разрешенный для данных параметров авторизации и типа системы идентификатор модуля представления системы
  - a) входной параметр `from` (тип: int) – идентификатор текущего модуля представления системы;

- b) возвращает наименование разрешенного системой исполняемого модуля представления системы (string).
- select\_ws\_default – возвращает модуль представления системы по умолчанию для данных параметров авторизации и типа системы
  - a) возвращает наименование разрешенного системой исполняемого модуля представления системы (string).

### 3.3.3. Класс *mod\_db*

Реализация функций работы с базой данных.

#### Открытые методы

- open\_base – открывает базу данных для подключений
  - a) возвращаемое значение отсутствует (void).
- close\_base – закрывает базу данных для подключений
  - a) возвращаемое значение отсутствует (void).
- db\_objects – возвращает объекты массива элементов результата запроса к БД
  - a) входной параметр query (тип:string) – запрос к БД;
  - b) возвращает массив элементов (array).
- db\_query – возвращает результат запроса к БД
  - a) входной параметр query (тип:string) – запрос к БД;
  - b) возвращает массив элементов (array).
- escape\_string – подготовка данных запроса к БД, экранирование специальных символов в строке запроса
  - a) входной параметр query (тип:string) – запрос к БД;
  - b) возвращает обработанную строку (string).

### 3.3.4. Класс *mod\_safe*

Реализация функций подготовки параметров ввода-вывода.

#### Открытые методы

- pget\_param – возвращает значение POST-запроса с преобразованием специальных символов
  - a) входной параметр value (тип:string) – переменная;
  - b) входной параметр default (тип:string) – значение по умолчанию;
  - c) возвращает значение запроса (string).

- `pget_param_fck` – возвращает значение POST-запроса с преобразованием специальных символов для формы `fckeditor`
  - a) входной параметр `value` (тип: `string`) – переменная;
  - b) входной параметр `default` (тип: `string`) – значение по умолчанию;
  - c) возвращает значение запроса (`string`).
- `pget_param_full` – возвращает значение POST-запроса с полным экранированием и преобразованием специальных символов
  - a) входной параметр `value` (тип: `string`) – переменная;
  - b) входной параметр `default` (тип: `string`) – значение по умолчанию;
  - c) возвращает значение запроса (`string`).
- `get_param` – возвращает значение GET-запроса с преобразованием специальных символов
  - a) входной параметр `value` (тип: `string`) – переменная;
  - b) входной параметр `default` (тип: `string`) – значение по умолчанию;
  - c) возвращает значение запроса (`string`).
- `pget_param_digit` – возвращает значение POST-запроса с преобразованием специальных символов для числа
  - a) входной параметр `value` (тип: `string`) – переменная;
  - b) входной параметр `default` (тип: `string`) – значение по умолчанию;
  - c) возвращает значение запроса (`string`).

### 3.3.5. Класс *mod\_content*

Реализация функций работы с контентом.

#### **Открытые методы**

- `check_auth` – устанавливает, имеет ли авторизованный пользователь идентификатор
  - a) возвращает идентификатор (`id`).
- `check_auth_admin` – устанавливает, имеет ли авторизованный пользователь доступ к контенту администратора
  - a) возвращает значение запроса (`boolean`).
- `interpr_ws` – возвращает наименование исполняемого модуля представления системы
  - a) входной параметр `ws` (тип: `string`) – идентификатор исполняемого модуля представления системы;
  - b) возвращает наименование исполняемого модуля представления системы (`string`).

- `interpr_lang` – возвращает наименование исполняемой языковой локализации системы
  - a) входной параметр `lang` (тип: `string`) – идентификатор исполняемой языковой локализации системы;
  - b) возвращает наименование исполняемой языковой локализации системы (`string`).
- `interpr_local` – возвращает наименование исполняемой географической локализации системы
  - a) входной параметр `local` (тип: `string`) – идентификатор исполняемой географической локализации системы;
  - b) возвращает наименование исполняемой географической локализации системы (`string`).
- `change_pass` – реализует форму смены пароля пользователя
  - a) возвращаемое значение отсутствует (`void`).
- `login_button` – реализует форму деактивации сеанса пользователя в системе
  - a) возвращаемое значение отсутствует (`void`).

#### 4. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

Информационный портал работает на серверной ЭВМ на базе ОС Linux с установленным web-сервером Apache, установленной средой PHP (версии 5.2 и выше) с поддержкой стандартного набора библиотек и характеристиками, обеспечивающими стабильную работу ОС и web-сервера в процессе организации последним доступа к сервисам. Для запуска и работы основных модулей требуется x86-совместимый компьютер с тактовой частотой процессора не ниже 500 МГц и оперативной памятью не менее 256 МБ, на котором должен быть установлен интернет-браузер Google Chrome, Mozilla Firefox, Opera или Safari (последних версий) с включенной поддержкой JavaScript.

Блоки Модель и Контроллер информационного портала реализованы с помощью языка программирования PHP 5.2, блок Представление реализован с помощью технологий Javascript, html DOM, Ajax (библиотека jquery). Связь с web-сервисами осуществляется на основе стандарта SOAP. Для хранения данных используется сервер базы данных MySQL. Для связи приложений с сервером базы данных используется язык запросов SQL. Хостинг сервисов осуществляется с помощью web-сервера Apache на UNIX-платформе.

Дополнительно для организации системы управления содержимым программного компонента используются сервисы сторонних разработчиков, такие как: CKEditor,

FSKEditor компании CKSource и модуль FancyBox. Данные модули являются программным обеспечением, свободно распространяемым по лицензиям GPL, LGPL или MPL.

При разработке портала использовано решение, которое обеспечивает высокий уровень безопасности при работе с данными. Система использует строгую фильтрацию запросов и проверку пользовательского ввода. Фильтры применяются для обработки запросов до и после вызова методов контроллера. Переданные параметры валидируются для каждого запроса, причем система автоматически удаляет неопознанные параметры, тем самым значительно снижая риск SQL-инъекций и аналогичных атак. Все пароли и ключи хранятся в зашифрованном виде, используется 128-битный алгоритм хеширования.

Оболочка портала использует стандартные протоколы обмена данными и стандартные средства разработки. Данное обстоятельство исключает несовместимость форматов при организации обмена данными со сторонними web-сервисами, подключении баз данных, доработке модулей системы, разработке и подключении дополнений к системе. Стоит отметить, что перечисленные сторонние средства должны также поддерживать стандартные средства разработки и использовать стандартные протоколы обмена данными.

## 5. ВЫЗОВ И ЗАГРУЗКА

Установка и настройка работы компонента осуществляется на базе web-сервера Apache (версии 2.2.16 и выше) со средой PHP (версии 5.2 и выше), с поддержкой стандартного набора библиотек, установленными стандартным образом в соответствии с документацией на указанные продукты. Для корректной работы компонента необходимо наличие входящих в дистрибутив компонента:

- конфигурационного файла `config.php`, содержащего базовые настройки запуска компонента;
- файла `index.php` – входная точка запуска компонента ;
- директории `cms`, содержащей файлы и директории, описывающие модель системы управления контентом:
  - `classes`– основные классы модели системы;
  - `mod` – реализация модели системы;
  - `scripts` – дополнительные внешние сервисы системы;

- директории content, содержащей файлы и директории, описывающие представление системы управления контентом:
  - classes – основные классы представления системы;
  - files – хранилище пользовательских файлов;
  - frames – реализация фреймов исполняемых модулей представления системы;
  - mod – реализация представления системы;
  - pages – реализация исполняемых модулей представления системы;
  - popup – реализация всплывающих окон исполняемых модулей представления системы;
  - scripts – реализация модулей интерфейса и их параметров на языке JavaScript;
  - tpl – реализация вида системы: макеты шаблонов, стили и элементы дизайна интерфейса.

Запуск компонента осуществляется при помощи интернет-браузера посредством ввода в строку адреса полного URL пути к файлу входной точки запуска index.php без указания имени самого файла.

После успешного запуска компонента для доступа к функциям администратора или пользователя необходимо авторизоваться. Если пользователь не может авторизоваться, он сможет воспользоваться только функциями гостя (неавторизованного пользователя). Для авторизации пользователю необходимо иметь учетную запись, наделяющую его правами доступа к системе, которая состоит из двух параметров: имя пользователя и пароль.

## 6. ВХОДНЫЕ ДАННЫЕ

Все входные и выходные данные программного компонента передаются по протоколу http. Для обмена входными данными используются два основных метода доступа: get, post.

При использовании метода get клиент сообщает адрес ресурса (URL), который он хочет получить, версию протокола http, MIME-типы документов, которые он поддерживает, версию и название клиентского программного обеспечения. Все эти параметры указываются в заголовке http-запроса. Тело запроса не передается. В ответ сервер сообщает версию http-протокола, код возврата, тип содержания тела сообщения, размер тела сообщения и ряд других необязательных директив http-заголовка. Сам ресурс передается в теле отклика. При обмене входными данными по методу post в запросе

клиента присутствует тело http-сообщения. Это тело формируется из данных, сформированных в html-форме. В отклике присутствуют заголовок и тело http-сообщения.

## **7. ВЫХОДНЫЕ ДАННЫЕ**

Сервер, приняв запрос от клиента, преобразует часть информации заголовка http-запроса в переменные окружения, которые доступны для анализа. Если запрос обладает телом, то оно становится доступным обработчику через поток стандартного ввода. Далее php-интерпретатор обрабатывает форму: переменные get-запроса попадают в массив `$_GET`, переменные post-запроса – в массив `$_POST`, серверные переменные – в массив `$_SERVER`. PHP передает полученные данные в модель обработки, далее происходит обработка данных на сервере, осуществляются обмен с сервером базы данных и с web-сервисами и отправка браузеру результата работы в виде заголовка, а в качестве ответа PHP формирует html-код, который является выходными данными.

## ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

AJAX	Asynchronous Javascript and XML (асинхронный JavaScript и XML)
CMS	Content Management System (Система управления содержимым)
DOM	Document Object Model (Объектная модель документа)
GPL	General Public License (Открытое лицензионное соглашение GNU)
HTML	HyperText Markup Language (Язык разметки гипертекста)
LGPL	Lesser General Public License (Стандартная общественная лицензия ограниченного применения GNU)
MIME	Multipurpose Internet Mail Extension (Многоцелевое расширение почты)
MPL	Mozilla Public License (Лицензия на свободное ПО)
MVC	Model–View–Controller (Модель–представление–контроллер)
PHP	Personal Hypertext Preprocessor (Препроцессор гипертекста)
SOAP	Simple Object Access Protocol (Простой протокол доступа к объектам)
SQL	Structured Query Language (Язык структурированных запросов)
URL	Uniform Resource Locator (Единый указатель ресурсов)
БД	База данных
МИТП	Многопрофильная инструментально-технологическая платформа
ОС	Операционная система
ПО	Программное обеспечение
ЭВМ	Электронно-вычислительная машина

