Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

Высшего профессионального образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной

инженерии

**Второй этап курсовой работы**

по дисциплине «Распределенные программные системы»

«Структура бизнес-логики и веб-интерфейса системы.»

**Выполнил:**

Ст. гр. ИСТ-113

Чугуев А.Н.

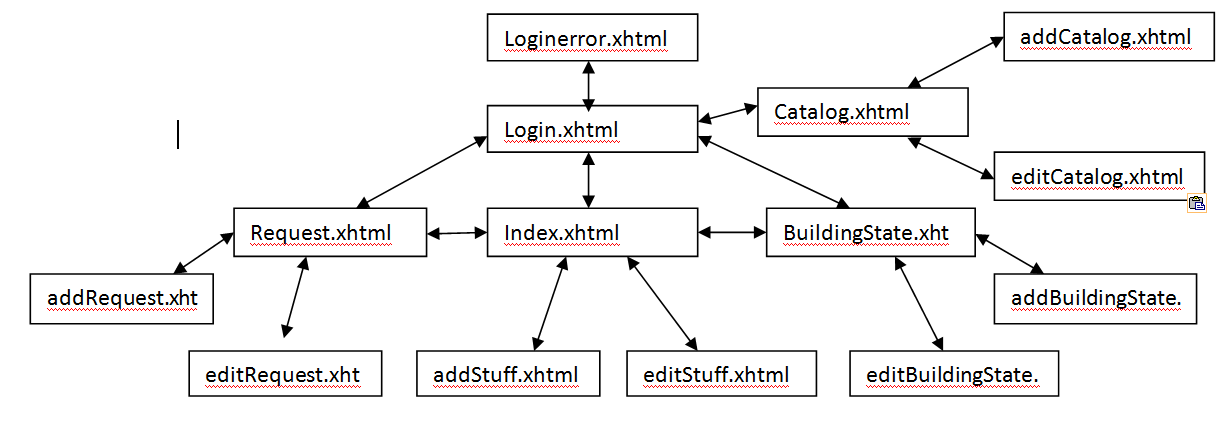
**Принял:**

Тимофеев А. А.

Владимир 2016

Схема веб интерфейса системы

Веб-интерфейс курсового проекта по теме «Автоматизация учёта расходных материалов в строительной фирме» приведен на рисунке 1.

Рисунок1. Схема веб-интерфейса системы

**Описание веб-страниц.**

1. index.xhtml – стартовая страница приложения. Она является также страницей обзора расходных материалов, на ней находятся ссылки на страницы Запроса и Объекта . Пользователь попадает на нее при открытии приложения, а так же при переходе на нее с других страниц, нажав соответствующую ссылку.

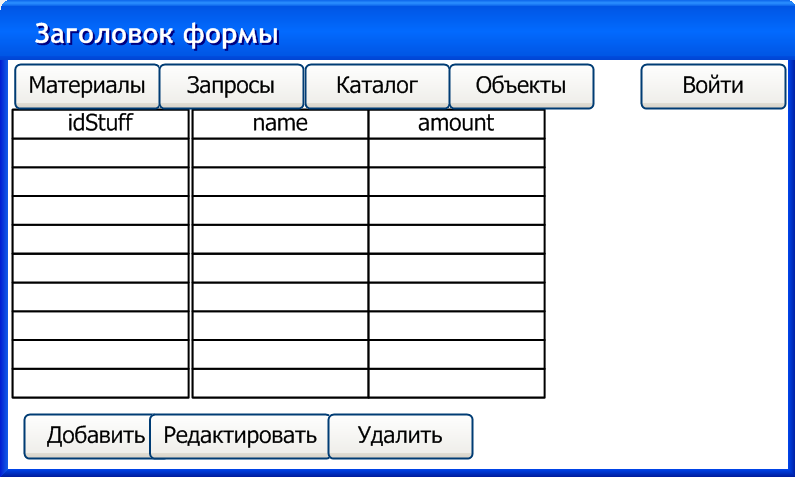


Рисунок2. Главная страница приложения

1. login.xhtml – страница авторизации. На ней расположены поля для ввода логина и пароля и ссылка на главную страницу. Пользователь попадает на эту страницу с главной при нажатии на кнопку «Войти в систему».

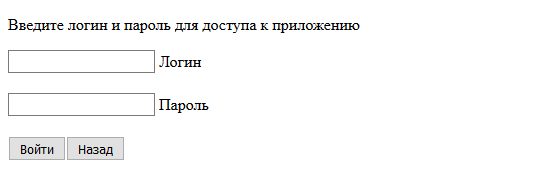


Рисунок3. Страница входа в систему

1. loginerror.xhtml – страница ошибки авторизации. На ней расположено сообщение о неверно указанной пользователем комбинации логина и пароля, а также ссылка на страницу авторизации. Пользователь попадает на эту страницу, если введенной им комбинации пользователя и пароля не обнаружилось в базе данных.
2. allUsers.xhtml – страница зарегистрированных пользователей системы. На ней находится таблица, в которой содержится информация о логинах пользователей и их ролях в системе, а также ссылка на страницу добавления нового пользователя. На данную страницу пользователь попадает, нажав на ссылку на главной странице, находясь в системе под ролью администратора.
3. addUser.xhtml – страница регистрации нового пользователя. На ней находятся поля для ввода информации о новом пользователе, а также ссылки на переход на главную страницу и на страницу со списком пользователя. Незарегистрированный пользователь попадает на данную страницу путем нажатия кнопки «Зарегистрироваться» на главной странице.



Рисунок5. Страница регистрации нового пользователя

1. addStuff.xhtml – страница добавления расходных материалов. На ней находятся поля для ввода информации о добавляемом материале и кнопка для подтверждения или кнопка назад. Пользователь попадает на эту страницу ссылкой со страницы расходных материалов.

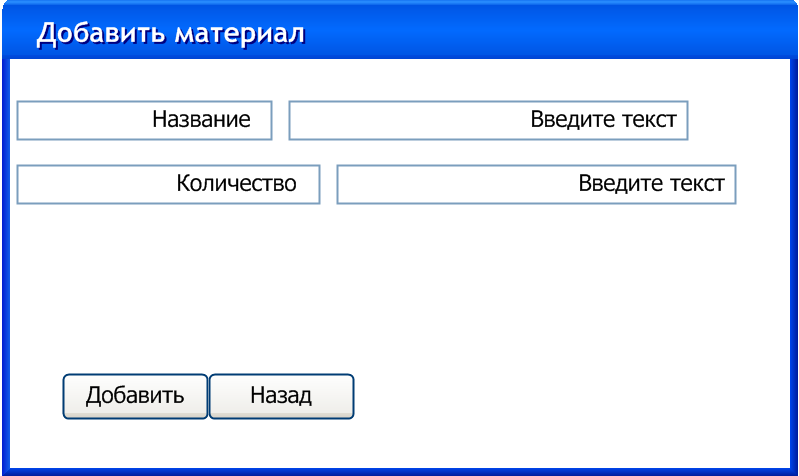


Рисунок7. Страница добавления нового расходного материала

1. editStuff.xhtml – страница редактирования расходного материала. На ней находятся поля для ввода информации о редактируемом материале и кнопка для подтверждения или кнопки назад. Пользователь попадает на эту страницу ссылкой со страницы расходных материалов.

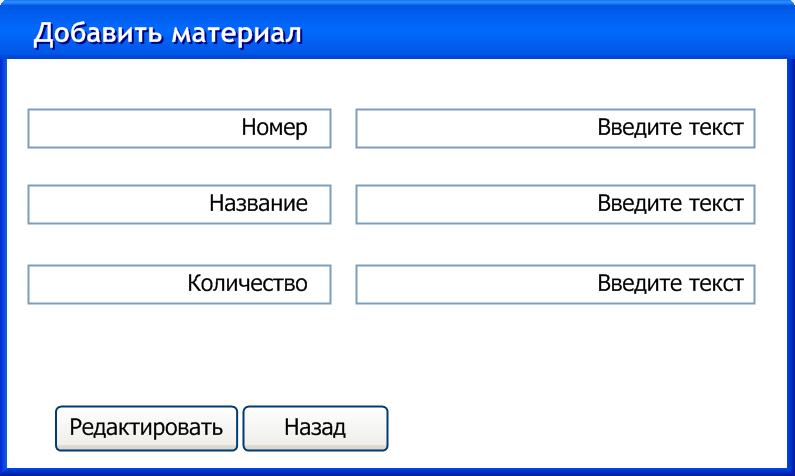


Рисунок8. Страница редактирования расходных материалов

1. Request.xhtml – страница запросов. На ней находится таблица, содержащая номер запрашиваемого материала, количество номер запроса и согласование для данного запроса. Пользователь попадает на эту страницу, нажав нужную ссылку на странице Запроса.

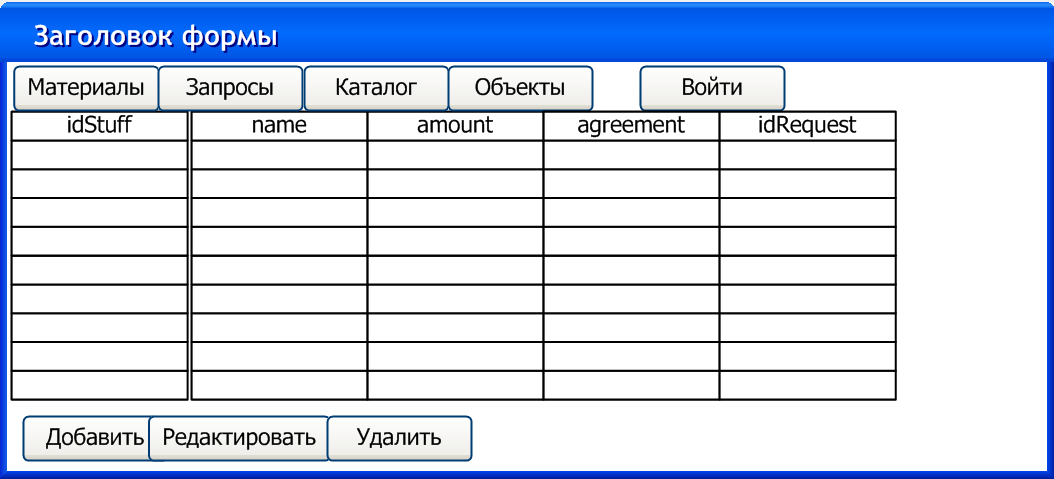


Рисунок10. Страница всех запросов

1. addRequest.xhtml – страница добавления запроса. На ней находятся поля для ввода информации о добавляемом запросе и кнопка для подтверждения или кнопки назад. Пользователь попадает на эту страницу ссылкой со страницы запроса.
2. editRequest.xhtml – страница редактирования запроса. На ней находятся поля для ввода информации о редактируемом запросе и кнопка для подтверждения или кнопки назад. Пользователь попадает на эту страницу ссылкой со страницы запроса.
3. BuildingState.xhtml – страница объекта. Содержит список, состоящий из номера объекта, названия, адреса, номера приписываемого акта и номера запросов с данного объекта. Пользователь может попасть на эту страницу по ссылке с главной страницы .

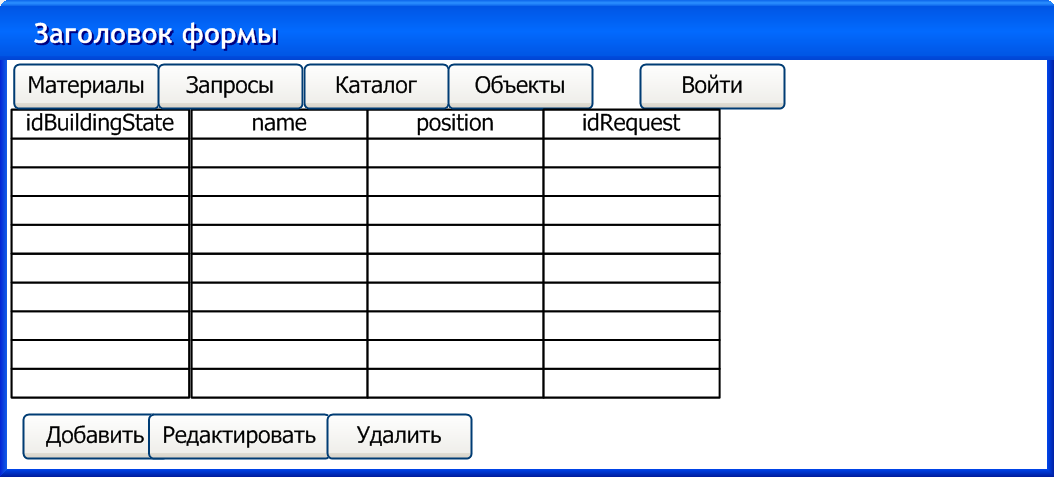


Рисунок13. Страница просмотра всех объектов.

1. addBuildingState.xhtml – страница добавления объекта. На ней находятся поля для ввода информации о добавляемом объекте и кнопка перехода на страницу всех остановок. Пользователь попадает на эту страницу ссылкой со страницы объекта.
2. edit BuildingState.xhtml – страница редактирования объекта. На ней находятся поля для ввода информации о редактируемом объекте. Пользователь попадает на эту страницу ссылкой со страницы объекта.
3. errorPage.xhtml – страница ошибок. Содержит сообщение о возникновении ошибки, дыб пользователь не был в неведении. На данную страницу пользователь будет попадать в случае возникновения в приложении ошибок рода 404,500 и т.д.
4. Catalog.xhtml – страница использования расходных материалов. Содержит список, состоящий из номера материала, номера запроса, названия, количества и даты . Пользователь может попасть на эту страницу по ссылке с главной страницы .

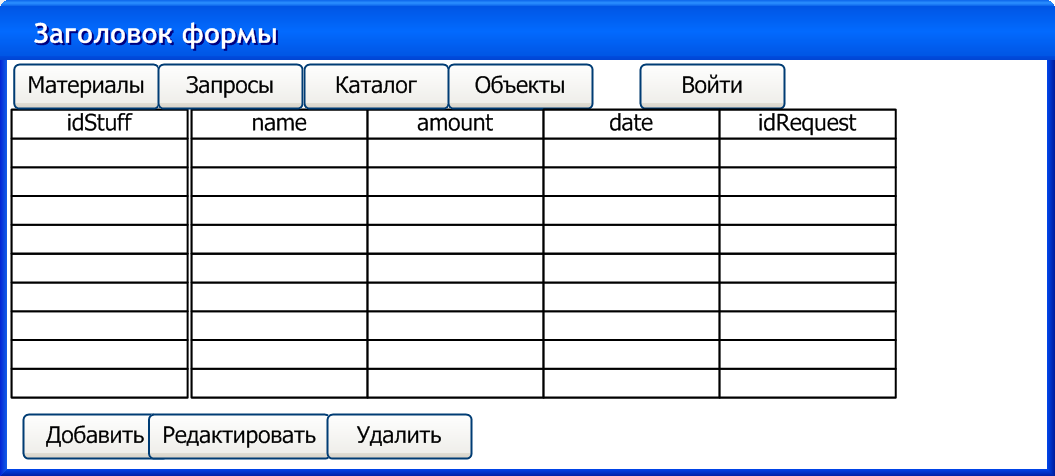


Рисунок13. Страница просмотра всех материалов.

1. addCatalog.xhtml – страница добавления материалов. На ней находятся поля для ввода информации об используемом материале и кнопка перехода обратно. Пользователь попадает на эту страницу ссылкой со страницы каталога.
2. edit Catalog.xhtml – страница редактирования материалов. На ней находятся поля для ввода информации о редактируемом материале. Пользователь попадает на эту страницу ссылкой со страницы каталога.

Диаграмма классов сущностей.

Диаграмма классов сущностей информационной системы «Автоматизация учёта расходных материалов в строительной фирме» приведена на рисунке 16.

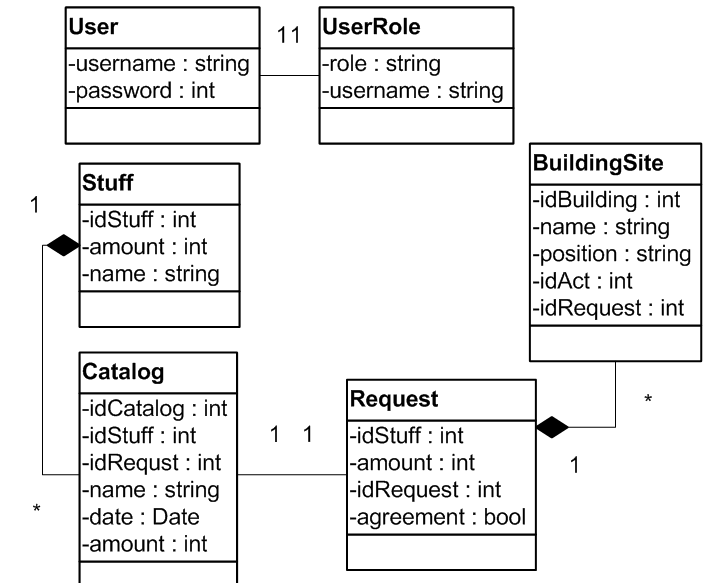


Рисунок16. Диаграмма классов сущностей.

**Описание классов.**

1. Класс Пользователь (myuser) определяет основные атрибуты, которыми обладает пользователь системы:

Атрибуты: логин (username), пароль (password);;

1. Класс роли пользователей (myuser\_roles) определяет соответствие пользователей с их ролями в системе:

Атрибуты: логин (username), роль (role);;

1. Stuff – класс, содержащий перечень расходных материалов на складе.

Атрибуты: номер материала(idStuff), количество материала(amount), название материала(name).

1. Catalog – справочник использованных прорабом материалов на текущем объекте за текущий день.

Атрибуты: Номер акта(idAct), Номер материала(idStuff), Номер запроса на этот день (idRequest), название (name), дата использования(date), количество(amount).

1. Request –класс запроса, котором приведён перечень необходимых прорабу расходных материалов.

Атрибуты: номер запроса(idRequest), номер материала(idStuff), количество(amount), соглашение (agreement).

1. BuildingState – класс объекта на котором проводятся работы.

Атрибуты: номер объекта(idBuildingState), название объекта(name), адрес(position), номер запроса(idRequest).

Диаграмма последовательностей.

**Расширенное описание прецедента «Составить отчет о доходах»**

* Название: «**Формирование документов**»
* Предусловие: Пользователь авторизован в системе как прораб и находится на странице с запросом.
* Действующее лицо: прораб.
* Основной поток: Составление запроса. Сотрудник нажимает кнопку добавить запрос на расходные материалы, после чего он попадает на форму создания запроса. Система предлагает сотруднику ввести материал и количество данного материала. После чего прораб нажимает кнопку «Создать» и система заносит данные в базу данных.
* Альтернативный поток: Другие сотрудники не могут составлять запросы, кроме сотрудника с ролью прораб. В случае попытки составить отчет не прорабом, сотрудник будет оповещен системой о невозможности ввести данные по причине недостатка прав.
* Постусловие: Система обрабатывает данные и вносит изменения в таблицу маршрутов. Сотрудники видят обновленную информацию на странице доходов маршрутов.

Диаграмма последовательностей прецедента «Формирование Документов» приведена на рисунке 17.

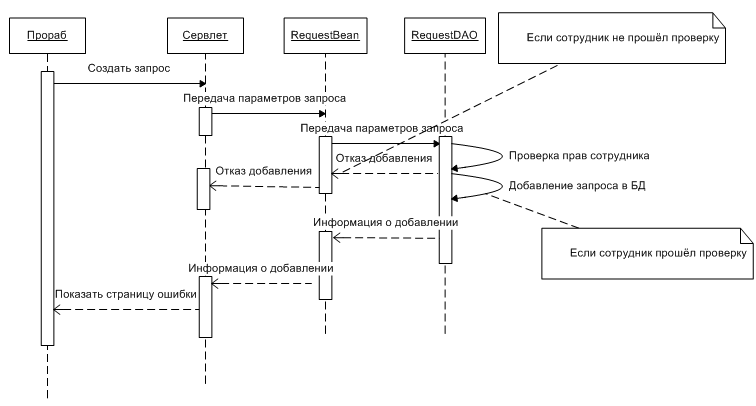


Рисунок17. Диаграмма последовательности для прецедента «Формирование документа»

 Сводная диаграмма классов/компонентов приложения с указанием связей между компонентами различных слоев.

Слой представления - это единственный слой, непосредственно связанный с пользователем. Слой представления используется для получения данных от пользователя и передачи их слою бизнес-логики для дальнейшей обработки, и когда данные получены в объекте-значении, он отвечает за представление объекта-значения в надлежащей форме, которая понятна пользователю.

Слой бизнес-логики (BUS) работает как мост между уровнем представления и DAO. Все значения пользователя, полученные от уровня представления, передаются в BUS. Слой бизнес-логики содержит всю бизнес-логику программы.

Слой доступа к базе данных (DAO) строит запрос на основе параметров, полученных от слоя бизнес-логики и используя соединение с базой данных, выполняет требуемые операции. Затем происходит возврат результатов из DAO классов в слой бизнес-логики.

Сводная диаграмма классов/компонентов приложения приведена на рисунке 18.

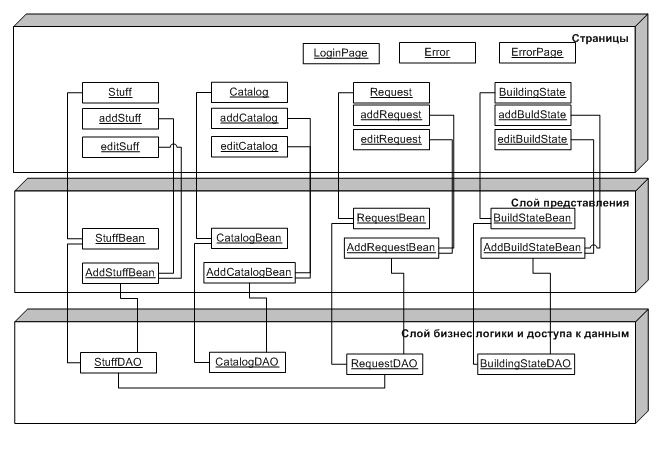


Рисунок18. Диаграмма компонентов и их взаимодействия

Спецификация компонентов системы.

**Слой бизнес-логики.**

1. **StuffDAO**
   1. List<Stuff>getAllStuff() – метод для вывода всех расходных материалов
   2. Void addStuff(String name, Int amount) – метод для добавления нового материала в БД.
   3. Void deleteStuff (List<Stuff>selectedList) – удаление выбранных материалов из БД. Входные параметры – список объектов.
   4. Void editStuff(int idStuff, int amount, String name) – метод для редактирования уже существующего отчета. Входные параметры – номер материала и количество.
2. **CatalogDAO**
   1. List<Catalog>getAllCatalog() – метод для вывода всех использованных материалов.
   2. Void addCatalog(int idStuff, String name, date Date, int amount, idRequest) – добавление использованного материала в базу данных. Входные параметры – номер материала, его количество, дата и номер запроса в котором был казан материал.
   3. Void deleteCatalog(List<Catalog>selectedList) – удаление выбранных материалов из БД. Входные параметры – список материалов.
   4. Void editCatalog(int idStuff, String name, date Date, int amount, idRequest) – метод для редактирования расхожного материала..
3. **BuildingStateDAO**
   1. List<BuildingState>getAllBuildingState() – метод для вывода всех объектов.
   2. Void addBuildingState (int idBuildingState, int idRequest, String name, String position) – добавление объекта в базу данных.
   3. Void deleteBuildingState (List<BuildingState>selectedList) – удаление выбранных объектов из БД. Входные параметры – список объектов.
   4. Void editBuildingState (int idBuildingState, int idRequest, String name, String position) – метод для редактирования уже существующего объекта.
4. **UserDAO**
   1. List<User>getAllUsers() – метод для вывода всех зарегистрированных пользователей системыиз БД на страницу.
   2. voidaddUser(Stringusername, Stringpassword, Stringrole) – метод для добавления нового пользователя в БД, регистрация. Входные параметры – логин, пароль и роль пользователя. Использует метод addUsername и addRole.
   3. voidaddUsername(Stringusername, Stringpassword) – метод для добавления пользователя в таблицу myuser в БД. Входные параметры – логин и пароль пользователя.
   4. voidaddRole(Stringusername, Stringrole) – метод для добавления роли пользователю. Входные параметры – логин и роль пользователя.
5. **RequestDAO**
   1. List<Report>getAllRequest() – метод для вывода всех запросов.
   2. Void addRequest(int idStuff, int amount, int idRequest, bool agreement) – добавление запроса в базу данных.
   3. Void deleteRequest (List<Request>selectedList) – удаление выбранных запросов из БД. Входные параметры – список запросов.

**Слой представления.**

1. **UserBean**
   1. List<User>getAllUsers() – метод для вывода на страницу списка пользователей. Использует метод UserDAO.getAllUsers.Возвращает список пользователей.
2. **AddUserBean**
   1. Stringlogout() – метод для выхода пользователя из авторизованного состояния. Возвращает ссылку на главную страницу.
   2. StringaddNewUser() – метод добавления пользователя в БД. Использует методы UserDAO.addUser. Возвращает ссылку на главную страницу.
3. **AddRequestBean**
   1. String AddRequest() – метод для добавления нового запроса. Использует метод RequestDAO.addRequest.
   2. String cancelAddRequest () – метод отмены создания запроса. Использует метод RequestDAO.cancelAddRequest.
4. **RequestBean**
   1. Void actionListener(ActionEventevent) – метод для удаления выбранных запросов из БД. Использует метод RequestDAO.deleteRequest.
5. **BuildingStateBean**
   1. Void actionListener(ActionEventevent) – метод для удаления выбранных строительных объектов из БД. Использует метод BuildingStateDAO.deleteBuildingState.
6. **AddBuildingStateBean**
   1. String addBuildingState () – метод для добавления строительного объекта в БД. Использует метод BuildingStateDAO.addBuildingState. Возвращает ссылку на страницу со всеми объектами.
   2. String editBuildingState (int idBuildingState) – метод для редактирования объектов. Использует метод BuildingState.editBuildingState. Возвращает ссылку на страницу со всеми объектами.
7. **StuffBean**
   1. List<Stuff>getAll Stuff () – метод для вывода на страницу списка всех материалов. Использует метод Stuff DAO.getAllStuff. Возвращает список материалов.
   2. String edit Stuff (int id Stuff) – метод для запоминания id редактируемого материала в сессии. Возвращает ссылку на страницу редактирования материала.
8. **AddStuffBean**
   1. String addStuff() – метод для добавления материала в БД. Использует метод StuffDAO.addStuff. Возвращает ссылку на страницу со всеми материалами.
   2. String editStuff (int idStuff) – метод для редактирования материала. Использует метод StufftDAO.editStufft. Возвращает ссылку на страницу со всеми материала.