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

Сербин А.А.

**Принял:**

Тимофеев А. А.

Владимир 2016

Схема веб интерфейса системы

Современные концепции проектирования распределенных информационных систем корпоративного уровня предполагают общение пользователя системы с базой данных посредством браузера и технологий Internet. Для обеспечения возможности взаимодействия пользователя с браузером применяется Веб-интерфейс — это совокупность средств, при помощи которых пользователь взаимодействует с веб-сайтом или любым другим приложением через браузер.

Веб-интерфейс моего курсового проекта по теме «Оптовый склад» приведен на рисунке 1.



Рисунок1. Схема веб-интерфейса системы

**Описание веб-страниц.**

1. index.xhtml – стартовая страница приложения. На ней располагаются приветственное сообщение, а также ссылки на страницу материалов, заявок, зарегистрированных пользователей системы, табеля, регистрацию и авторизацию. Пользователь попадает на нее при открытии приложения, а так же при переходе на нее с других страниц, нажав соответствующую ссылку.

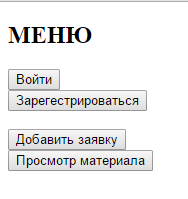


Рисунок2. Главная страница приложения

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».

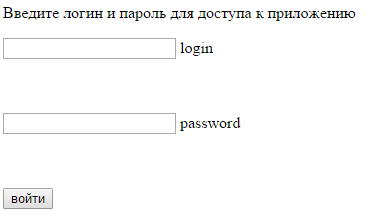


Рисунок3. Страница входа в систему

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, администратор может попасть на эту страницу со страницы всех пользователей для добавления нового пользователя с нужной ему ролью.

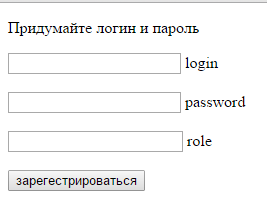


Рисунок4. Страница регистрации нового пользователя

1. AllMat.xhtml – страница склада материалов. На ней располагается таблица со сведениями о всех материалах, находящихся на складе.

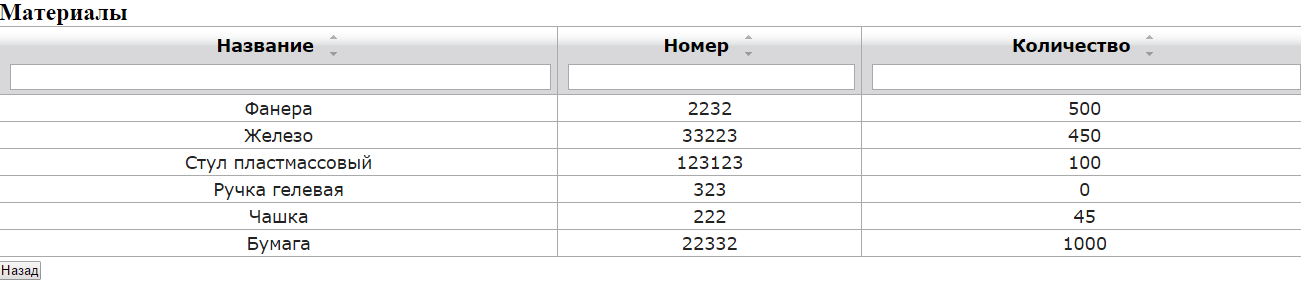


Рисунок5. Страница всех материалов

1. AddMaterial.xhtml – страница добавления нового материала. На ней находятся поля для ввода информации о добавляемом материале и кнопка добавления.

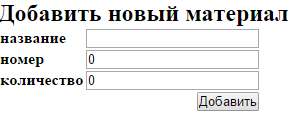


Рисунок6. Страница добавления нового материала

1. EditMaterial.xhtml – страница редактирования материала. На ней находятся поля для ввода информации о редактируемом материале и кнопка сохранения изменений.

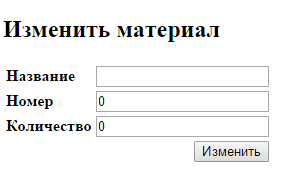


Рисунок7. Страница редактирования материала

1. AddAppProviders.xhtml – страница создания заявки на получение материала на склад от поставщиков. Поля для ввода информации, для получения заявки, кнопка для расчета общей суммы за товар, а также кнопка для просмотра поставщиков и какие товары за какую цену они поставляют.

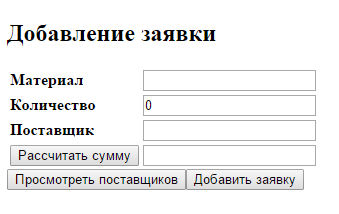


Рисунок8. Страница добавления заявки поставщикам

1. AllTs.xhtml – страница просмотра табеля. На ней находятся таблицы, описывающие выполнение заявок, на получение или на выдачу материалов, также находится информация по состоянию капитала самого склада и выплаты пользователей.

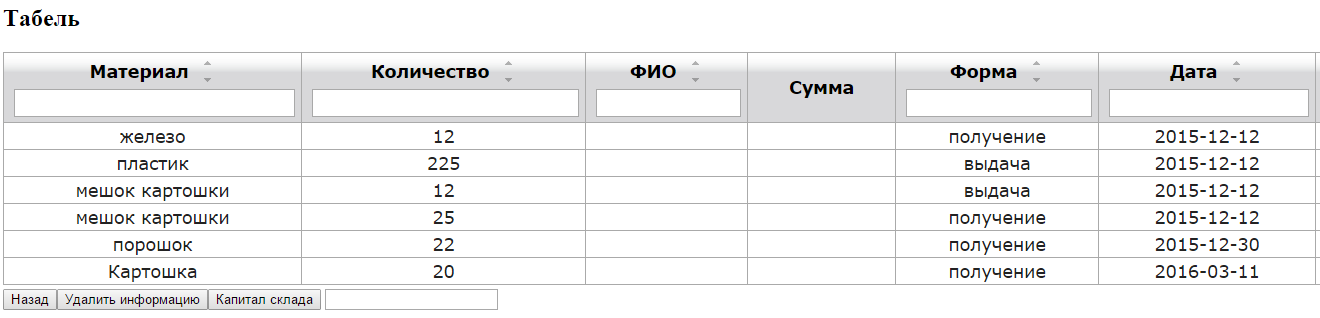


Рисунок9. Страница табеля

1. AddApp.xhtml – страница заполнения заявки на получения материала со склада. На ней располагается поля для ввода информации, а также есть возможность рассчитать общую сумму за получаемый материал.

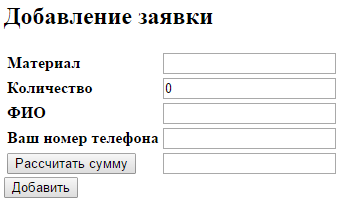


Рисунок10. Страница добавления заявки

1. EditApplication.xhtml – страница редактирования заявок остановок. На ней находятся поля для ввода информации о редактируемой заявке и кнопка сохранения изменений

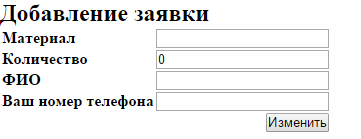


Рисунок11. Страница редактирования заявки.

1. errorPage.xhtml – страница ошибок. Содержит сообщение о возникновении ошибки, чтобы пользователь не был в неведении. На данную страницу пользователь будет попадать в случае возникновения в приложении ошибок рода 404, 500 и т.д.

Диаграмма классов сущностей.

Диаграммы классов показывают статическую структуру системы, то есть определяют типы объектов системы и различного рода статические связи и отношения между ними. Диаграммы классов содержат набор статических элементов, как, например, классы, типы, их связи, объединенные в граф.

Диаграмма классов сущностей информационной системы «Оптовый склад» приведена на рисунке 12.



Рисунок12. Диаграмма классов сущностей.

**Описание классов.**

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

1) Логин (String username);

2) Пароль (String password);

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

1) Логин (String username);

2) Роль (String role);

1. Класс Заявка (Application) представляет собой форму заполнения заявки, на получение материала со склада. Класс содержит следующие атрибуты:

1) Номер заявки (idApplication);

2) Название материала (Material)

3) Количество материала (Quantity)

4) ФИО клиента (FIO)

5) Общая сумма (Summ)

1. Класс Материалы (Material) представляет собой список доступных материалов на складе, обладает следующими атрибутами:

1) Номер в таблице (idMat)

2) Название материала (Name)

3) Цифровой номер (number)

4) Количество (Quantity)

1. Класс Табель (TimesHeet) содержит информацию по выполненным заявкам. Содержит атрибуты:
2. Номер (Id)
3. Название материала (Material)
4. Количество (Quantity)
5. ФИО Клиента (FIO)
6. Сумму, которую заплатил клиент (Price)
7. Дата, когда была выполнена заявка (Date)
8. Класс Заявка для поставщиков (ApplicationProviders) – заявка на получение товара на склад от поставщиков, содержит атрибуты:

1)Название материала (material)

2)Количество (quantity)

3)Название поставщика (providers)

4)Общая сумма за товар (allsum)

1. Класс Поставщики (Providers) – информация о поставщиках, их товарах и ценах. Содержит атрибуты:
2. Название поставщика (NameProvider)
3. Название материала (Material)
4. Цена за этот материал (Price)

Диаграмма последовательностей.

**Расширенное описание прецедента «Заказать материал на склад»**

* Название: «Заказать материал на склад».
* Предусловие: Пользователь авторизован как администратор.
* Действующее лицо: Администратор
* Основной поток: Администратор просматривая страницу с материалами, смотрит какие отсутствует и их надо заказать, смотрит поставщиков, у кого какой материал дешевле и выгоднее купить и заполняет заявки на получение этого материала.
* Альтернативный поток: Материал присутствует на складе и его не надо заказывать.
* Постусловие: Администратор перенаправляется на страницу заполнения заявки на получение материала.

Диаграмма последовательности является одной из разновидности диаграмм взаимодействия и предназначена для моделирования взаимодействия объектов Системы во времени, а также обмена сообщениями между ними.

Диаграмма последовательностей прецедента «Заказать материал на склад» приведена на рисунке 13.



Рисунок13. Диаграмма прецедента «Заказать материал на склад»

 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.



Рисунок14. Диаграмма компонентов и их завимодействия

Спецификация компонентов системы.

**Слой бизнес-логики.**

1. **TsDAO**
   1. List<TimesHeet> selectTimesHeet – метод для вывода всей информации по табелю
   2. public void addTimesHeet – метод добавления новой информации в табель
   3. public void deleteTs – метод удаления информации из табеля
   4. public void editTs – метод редактирования информации в табеле
2. **MatDAO**
   1. List<Material> selectMaterial – метод для вывода списка всех материалов
   2. public void addMaterial – метод добавления нового материала
   3. public void deleteMaterial – метод удаления материала
   4. public void editMaterial – метод редактирования материала
3. **AppDAO**
   1. List<Application> selectApplication – метод для вывода списка всех заявок
   2. public void addApplication – метод для добавления новой заявки
   3. public void deleteApplication – метод для удаления заявки
   4. public void editApp –метод для редактирования заявки
   5. public void Save – метод для выполнения заявки с занесением информации в табель
4. **DAOUser**
   1. public void addUser – метод для добавления нового пользователя
   2. public List<User> selectUser() – метод для вывода всех пользователей
   3. public void deleteUser – метод для удаления пользователя
5. **ProviderDAO**
   1. public List<Providers> selectProviders() – метод для вывода всех поставщиков с их материалами и ценами.
   2. Public void AddAppProvider – метод для добавления новой заявки для получения материала на склад от поставщиков.
   3. Public void editAppProvider – метод для редактирования заявки
   4. Public void deleteApp – метод для удаления заявки

**Слой представления.**

1. **TsBean**
   1. void actionListener(ActionEvent event) – метод для удаления выбранной информации табеля из БД.
   2. List<Stop> getAllMat() – метод для вывода всей информации по табелю
2. **AddTsBean**
   1. String AddApp() – метод для добавления новой информации в табель
   2. String editTs() – метод для редактирования информации
3. **AppBean**
   1. void actionListener(ActionEvent event) – метод для удаления выбранной заявки из БД.
   2. List<Route> getAllApp() – метод для вывода всех заявок
4. **AddAppBean**
   1. String AddApp() – метод для добавления новой заявки
   2. String editApp() – метод для редактирования заявки1
5. **MatBean**
   1. void actionListener(ActionEvent event) – метод для удаления выбранных материалов из БД.
   2. List<Stop> getAllMat() – метод для вывода всех материалов
6. **AddMatBean**
   1. String addNewMat() – метод для добавления материала
   2. String editMat() – метод для редактирования материала
7. **ProviderBean**
   1. void actionListener(ActionEvent event) – метод для удаления выбранных материалов из БД.
   2. List<Stop> getAllProviders() – метод для вывода всех поставщиков
8. **AddProviderBean**
   1. String addNewAppProv() – метод для добавления заявки
   2. String editAppProv() – метод для редактирования заявки
9. **UserBean**
   1. List<User> getAllUsers() – метод для вывода на страницу списка пользователей. Использует метод UserDAO.getAllUsers. Возвращает список пользователей.