

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра «Распределенные вычисления и компьютерные сети»

ОТЧЕТ

по курсовой работе

по дисциплине: «Программное обеспечение распределенных вычислительных систем»

Тема работы: «Система управления распределенным коллективом»

Работу выполнил студент

63507/1 *Иванов К.Г.*

Преподаватель

_____ *Стручков И.В.*

Санкт-Петербург
2016

Оглавление

1	Введение	2
2	Анализ задания	2
2.1	Формулировка задания	2
2.2	Роли	2
2.3	Варианты использования	3
2.4	Модель предметной области	5
2.5	Диаграмма последовательности	6
3	Реализация	8
3.1	Слой хранения данных	8
3.2	Слой бизнес-логики	8
3.3	Слой представления	8
3.4	Тестирование	10
3.5	Инструкция системного администратора	11
3.6	Инструкция пользователя	12
4	Выводы	12

1 Введение

В рамках курса было необходимо разработать приложение, позволяющее продемонстрировать применение основных принципов разработки программного обеспечения. В частности, приложение должно содержать следующие компоненты:

- слой бизнес-логики;
- слой хранения данных;
- слой представления.

Целью работы являлось разработать систему управления распределенным коллективом, в которой взаимодействуют менеджеры и работники. Работники должны иметь доступ к списку своих задач с возможностью отмечать их, как выполненные. Менеджеры выдают задания работникам с назначением приоритета, отмечают их, как выполненные, и удаляют. Задачи могут создавать и те, и другие.

В ходе работы требовалось выполнить следующие задачи:

1. Изучить разработку web-приложений с помощью языка Java.
2. Изучить компоненты системы EJB, необходимые для создания, развертывания и функционирования распределенных приложений Java масштаба предприятия.
3. Овладеть базовыми навыками разработки распределенных корпоративных Java-приложений в рамках технологии EJB.
4. Исследовать использование технологии хранения данных (persistence) Java EE 6 для размещения данных приложения в СУБД.

2 Анализ задания

2.1 Формулировка задания

Требуется реализовать система управления распределенным коллективом. Рабочее место менеджера позволяет выдавать задания работникам с назначением приоритета. Рабочее место работника позволяет получать список задач в порядке убывания приоритета и отмечать выполненные задания. Операции удаленного объекта: создать задание, просмотреть список заданий для определенного работника, отметить задание, как выполненное. Сериализуемый объект: описание задания с приоритетом.

2.2 Роли

В проекте может быть выделено две роли: менеджер и работник.

1. Менеджер:

- Смотрит список свободных задач
- Смотрит список задач работника
- Назначает задачу работнику с указанием приоритета

- Снимает задачу с работника
- Добавляет задачу
- Отмечает задачу, как выполненную
- Удаляет задачу

2. Работник:

- Смотрит свои задачи по убыванию приоритета
- Добавляет задачу
- Отмечает задачу, как выполненную

2.3 Варианты использования

Создать задание

1. Пользователь авторизуется и выбирает кнопку «создать задание»
2. Система отображает окно «Новое задание»
3. Пользователь вводит описание задания и нажимает кнопку «ОК»
4. Система уведомляет об успешном добавлении задания

Альтернатива: этап 1 Пользователя с таким логином/паролем не существует: система выдает уведомление и просит ввести еще раз

Просмотреть список задач (работник)

1. Работник авторизуется
2. Система показывает список заданий, на которые он назначен, в порядке убывания приоритета

Альтернатива: этап 1 Работника с таким логином/паролем не существует: система выдает уведомление и просит ввести еще раз

Просмотреть список задач работника и снять задачу (менеджер)

1. Менеджер авторизуется
2. Менеджер выбирает работника из списка
3. Система показывает список задач, закрепленных за работником
4. Менеджер выбирает задачу и нажимает кнопку «снять»
5. Система убирает задачу из списка закрепленных за работником

Альтернатива: Этап 1 Менеджера с таким логином/паролем не существует: система выдает уведомление и просит ввести еще раз

Удалить задачу (менеджер)

1. Менеджер авторизуется
2. Менеджер выбирает работника из списка
3. Система показывает список задач, закрепленных за работником
4. Менеджер выбирает задачу и нажимает кнопку «Удалить»
5. Система удаляет задачу

Альтернатива: Этап 1 Менеджера с таким логином/паролем не существует: система выдает уведомление и просит ввести еще раз

Альтернатива: Этап 2 Менеджер выбирает свободную задачу и нажимает кнопку «Удалить»

Назначить задание работнику

1. Менеджер авторизуется
2. Система показывает список свободных задач
3. Менеджер выбирает задачу и нажимает кнопку «Назначить»
4. Система показывает окно «Назначение задачи» со списком работников и полем назначения приоритета
5. Менеджер выбирает работника и приоритет, нажимает кнопку «ОК»
6. Система уведомляет об успехе процедуры

Альтернатива: Этап 1 Менеджера с таким логином/паролем не существует: система выдает уведомление и просит ввести еще раз

Пометить задание, как выполненное (работник)

1. Работник авторизуется
2. Система показывает список заданий, на которые он назначен, в порядке убывания приоритета
3. Работник выбирает задание и нажимает кнопку «Задача готова»
4. Система уведомляет об успехе операции

Альтернатива: Этап 1 Работника с таким логином/паролем не существует: система выдает уведомление и просит ввести еще раз

Пометить задание, как выполненное (менеджер)

1. Менеджер авторизуется
2. Менеджер выбирает работника из списка
3. Система показывает список задач, закрепленных за работником
4. Менеджер выбирает задачу и нажимает кнопку «Задача готова»

Альтернатива: Этап 1 Менеджера с таким логином/паролем не существует: система выдает уведомление и просит ввести еще раз

Альтернатива: Этап 2 Менеджер выбирает свободную задачу и нажимает кнопку «Задача готова»

2.4 Модель предметной области

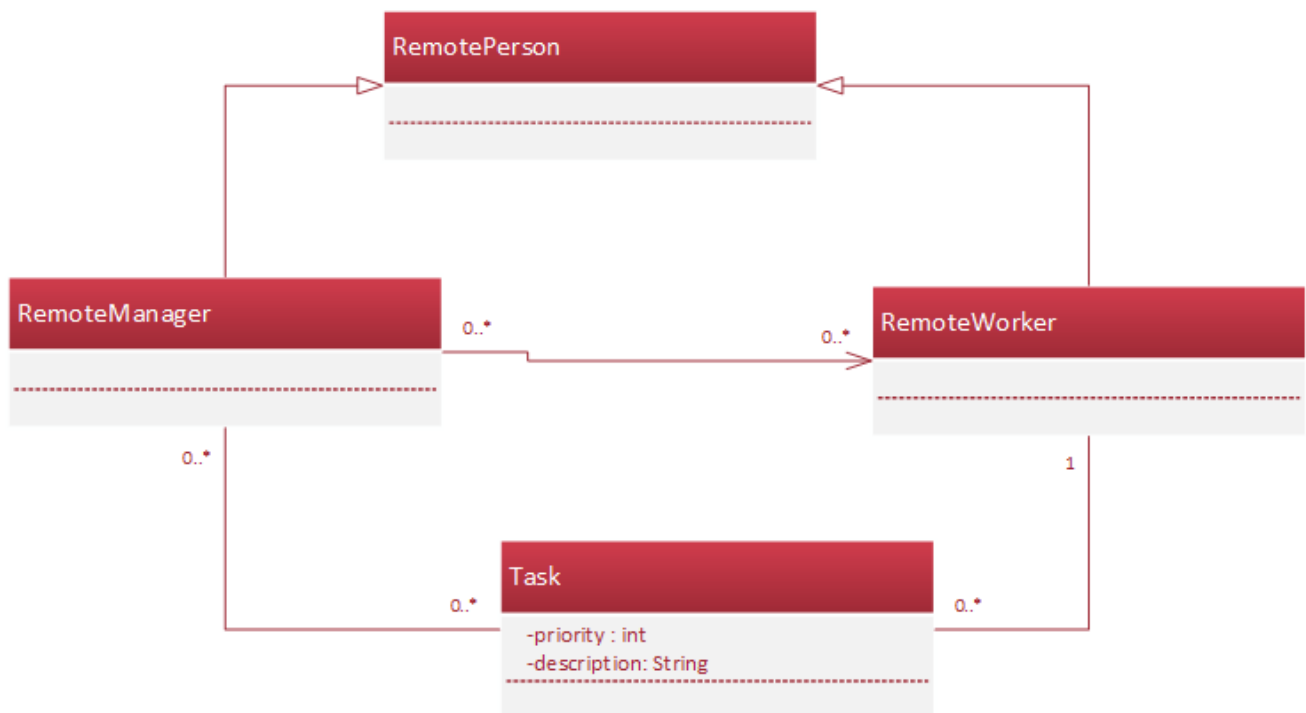


Рис. 1: Модель предметной области

2.5 Диаграмма последовательности

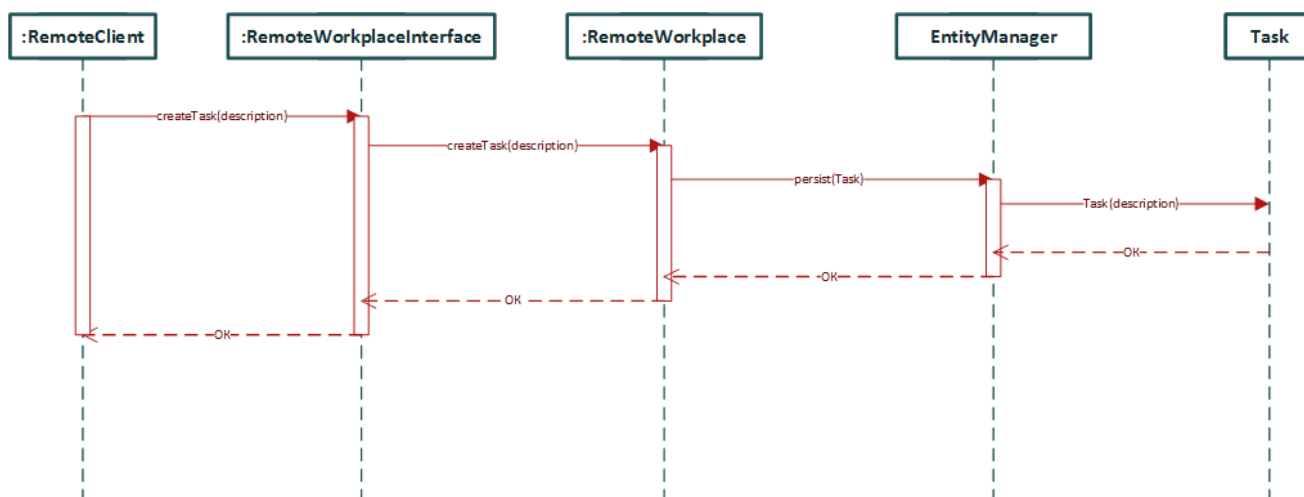


Рис. 2: Добавление задачи

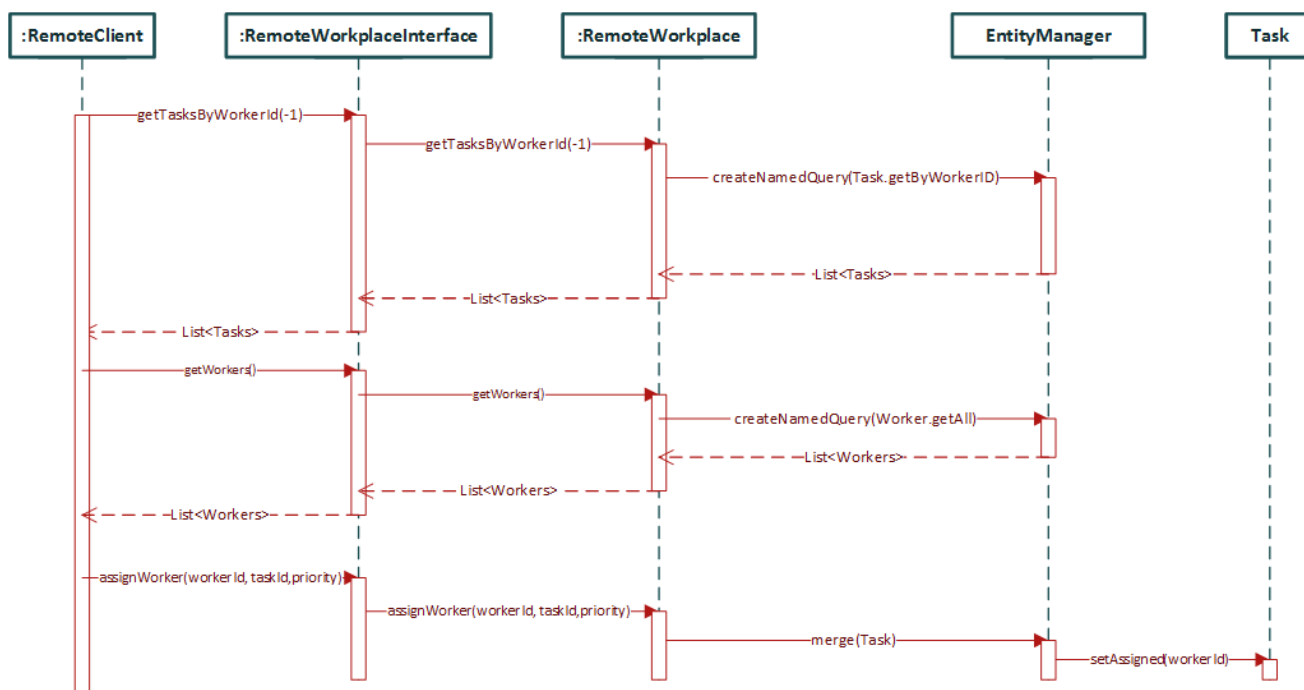


Рис. 3: Назначение задачи

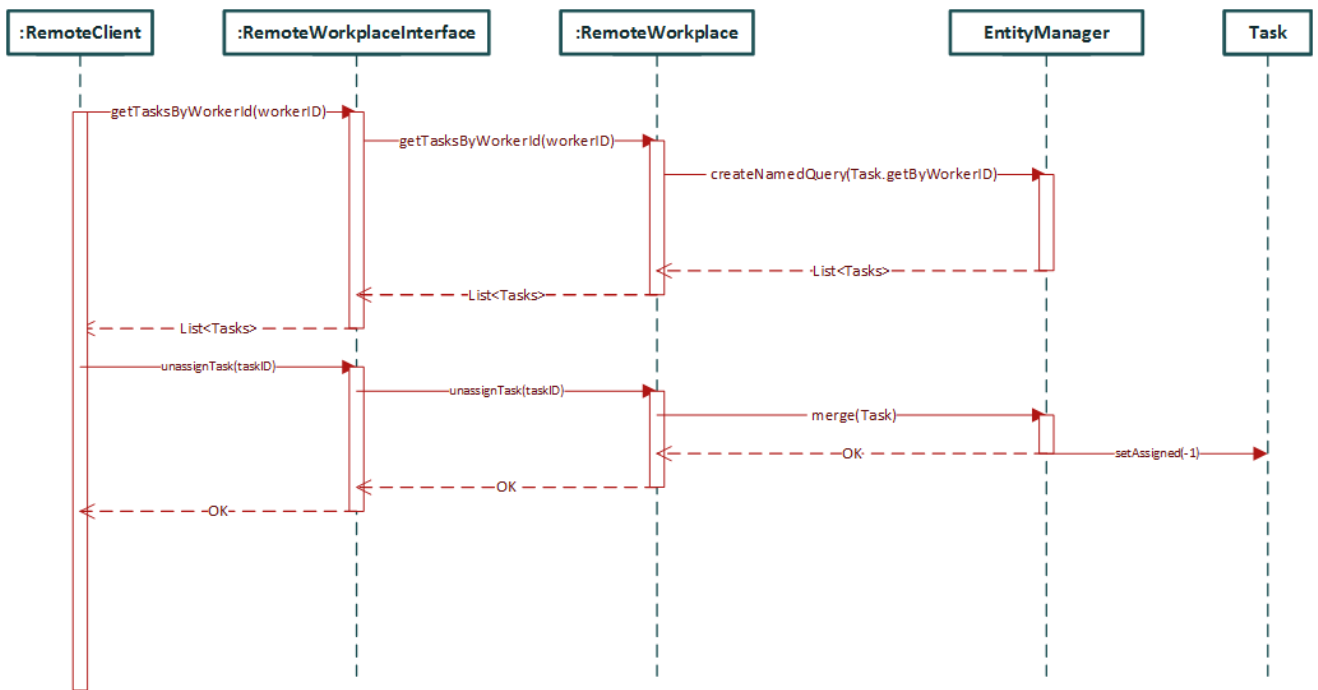


Рис. 4: Снятие задачи

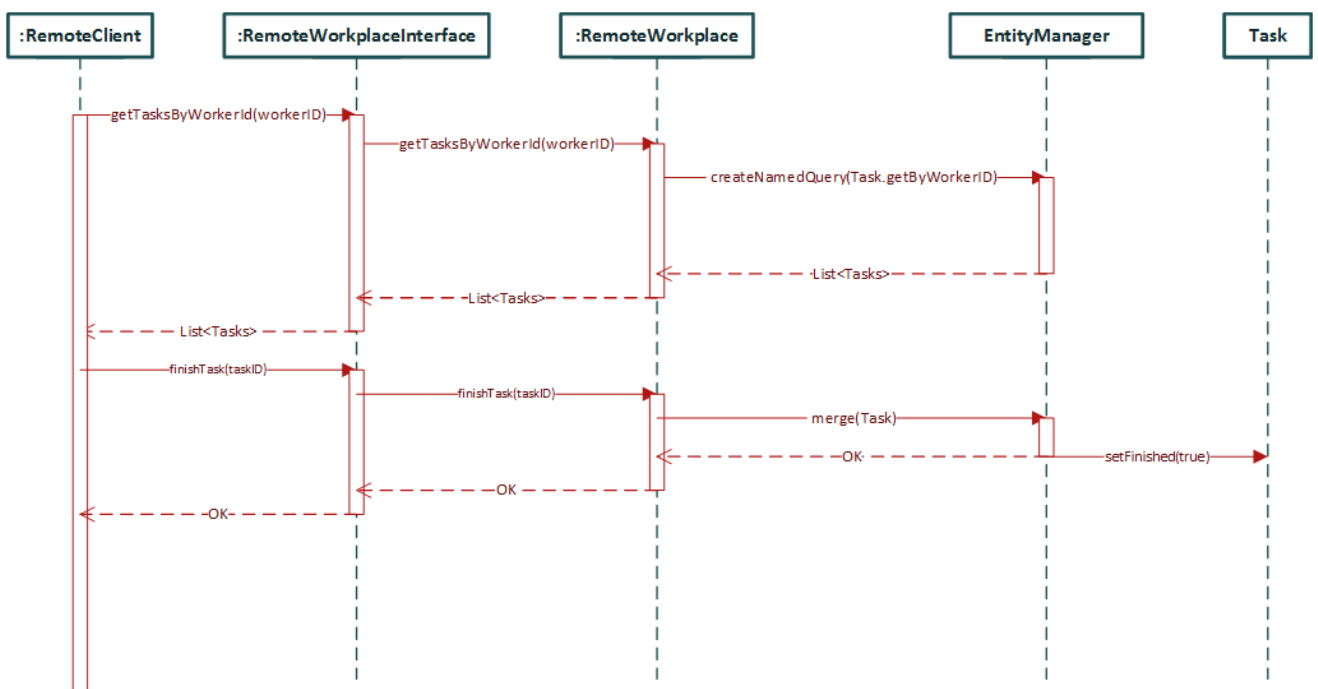


Рис. 5: Завершение задачи

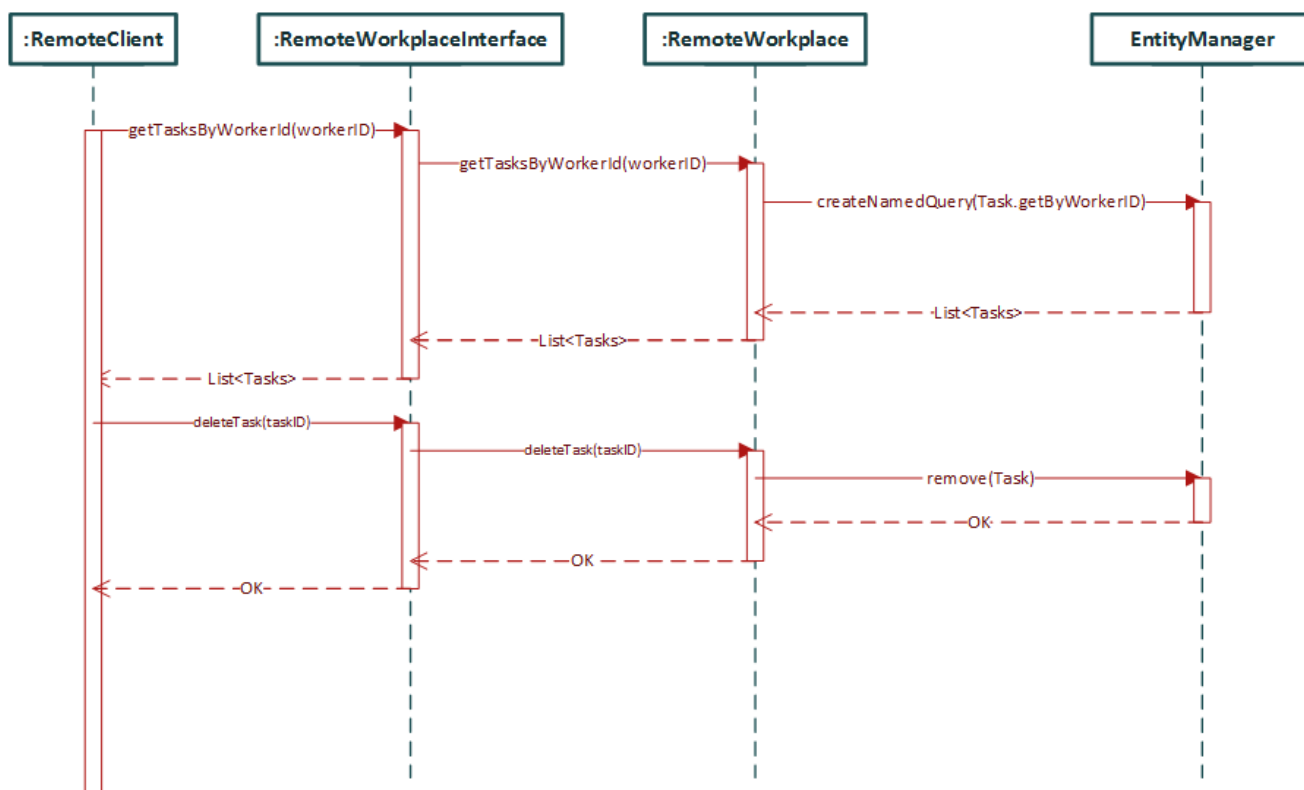


Рис. 6: Удаление задачи

3 Реализация

3.1 Слой хранения данных

В качестве СУБД для хранения данных использовалась JavaDB. В соответствии с моделью предметной области, были выделены основные сущности: «пользователь» и «задача». Были созданы таблицы, соответствующие данным сущностям.

3.2 Слой бизнес-логики

Для каждой из сущностей «пользователь» и «задача» были автоматически сгенерированы классы, соответствующие таблицам, созданным ранее в СУБД. Бизнес-логика расположена в пакете (*dcp.ivanov*), содержащий следующие классы:

- *Task.java* - класс, соответствующий сущности «задача»
- *Worker.java* - класс, соответствующий сущности «пользователь»
- *WorkplaceSessionBean.java* - реализация интерфейса взаимодействия удаленного пользователя с БД

3.3 Слой представления

Для разработки слоя представления была выбрана технология Java Server Pages. Покажем скриншоты всех имеющихся JSP страниц.

Login:

Password:

Рис. 7: Пример страницы авторизации *index.jsp*

Hello, manager Stas Rets

Avalanche, priority: 6; done

Blow up Death star, priority: 0

Kaboom!., priority: 0; done

Рис. 8: Главная страница менеджера *main.jsp*

Hello, worker Luke Skywalker

Aargh, priority: 12222; done

Finish

Shit!, priority: 35; done

Finish

Kill Vader, priority: 4; done

Finish

addTask

logout

Рис. 9: Главная страница работника *main.jsp*

Insert task
description here

Submit

Task successfully added

Return

Рис. 10: Добавление задачи *addTask.jsp*

3.4 Тестирование

Вариант тестирования	Ожидаемый результат	Фактический результат
<i>Менеджер</i>		
Войти с логином и паролем менеджера	Будет открыта главная страница менеджера	Открылась главная страница менеджера
Выбрать работника из списка	Будет показан список задач работника	Показан список задач работника

Ввести приоритет, выбрать работника и нажать на кнопку «Assign»	Задача появится в списке для работника	Задача появилась в списке для работника
Нажать на кнопку «Finish»	У задачи появится метка «done»	У задачи появилась метка «done»
Нажать на кнопку «Delete»	Задача исчезнет из всех списков	Задача исчезла
Нажать на кнопку «Unassign»	Задача вернется в список свободных	Задача вернулась в список свободных
Нажать на кнопку «AddTask», ввести описание задачи и затем «Return»	Ничего не изменится	Ничего не изменилось
Нажать на кнопку «AddTask», ввести описание задачи и затем «Submit»	Добавится новая задача	Задача добавилась
<i>Работник</i>		
Войти с логином и паролем работника	Будет открыта главная страница работника	Открылась главная страница работника
Нажать на кнопку «Finished» у задачи	Появится метка «done»	Появилась метка «done»

3.5 Инструкция системного администратора

Для развертывания приложения необходимы следующие системные компоненты:

1. СУБД JavaDB (поставляется с JDK)
2. Apache Ant
3. Сервер Glassfish версии 4.0

Настройка СУБД:

1. Запустить сервер СУБД через NetBeans, вкладка Services (порт 1527)
2. Создать схему БД
3. Создать таблицы Tasks и Workers в соответствии с диаграммой классов
4. Добавить необходимых пользователей

Затем, открыв проект через NetBeans, можно нажать кнопку Deploy, а после успешного окончания - Run для запуска приложения

3.6 Инструкция пользователя

При запуске приложения пользователь попадает на страницу авторизации, с которой он может перейти на страницу регистрации. В зависимости от типа пользователя, соответствующего паре логин-пароль, главная страница выглядит различно.

Менеджер

Главная страница менеджера представлена на рисунке 8. Изначально показаны свободные задачи. Для просмотра задач конкретного работника его требуется выбрать из списка вверху страницы и нажать кнопку «Choose». Для задач указан их приоритет и выполнены ли они (done).

Чтобы назначить работника на свободную задачу, нужно выбрать его из списка рядом с требуемой задачей, ввести целочисленный приоритет и нажать кнопку «Assign». Обратно, для снятия работника с задачи требуется нажать кнопку «Unassign» в списке задач этого работника. Свободные задачи можно также удалять нажатием на кнопку «Delete». Чтобы отметить задачу, как выполненную, достаточно нажатия на кнопку «Finish». Удостовериться в успехе можно, снова взглянув на это задание: около него будет написано «done».

Работник

Главная страница работника представлена на рисунке 9. Выводится список задач, назначенных этому работнику, с приоритетами и метками о выполнении. Отметить задачу, как выполненную, работник может нажатием кнопки «Finish» рядом с задачей, и она будет в списке с пометкой «done».

Оба пользователя могут создавать задачи нажатием кнопки «addTask». Соответствующее окно показано на рисунке 10. Можно ввести описание задачи и добавить её нажатием кнопки «Submit». Система ответит сообщением об успехе. Вернуться обратно можно нажатием на кнопку «Return».

4 Выводы

В данной курсовой работе получены начальные навыки работы с технологией EJB и сервером приложений Glassfish, а также разработки WEB приложений при помощи Java Server Pages и взаимодействия с базой данных JavaDB. Были достигнуты следующие задачи:

- Разработана схема простой базы данных в JavaDB.
- В соответствии с поставленным заданием разработан проект при помощи EJB
- Разработан прототип WEB-приложения на основе jsp-страниц
- Получен опыт взаимодействия с сервером приложений Glassfish

Данный проект может быть улучшен путем добавления дополнительного функционала, а также разработки более продвинутого пользовательского интерфейса.