# Постановка задачі

Розробити web-додаток для обліку і планування робочого часу на проектах.

1. **Короткі відомості**

У даному проекті були застосовані такі технології та фреймворки:

1. JavaServer Faces 2.2 (Facelets, Richfaces)
2. Spring
3. Сервер – Glassfish 4.1
4. БД – Oracle XE (MySQL для працюючого екземпляра на Java-хості Jelastic)
5. Репозиторій – Github
6. Демо-сервер – Jelastic
7. **Опис роботи програми**

У даній програмі існує 3 типи ролей, які можуть користуватись сервісами (consumer - замовник, admin – адміністратор сайту, employee - працівник).

Основні функції замовника:

1. При вході у систему замовника для нього передбачена реєстрація та вхід згідно існуючого логіну та паролю. Для зберігання паролю застосовується одностороннє кодування SHA-256.
2. У інтерфейсі передбачено можливість подання заявки на проект (button New Enquire) та перегляд усіх поданих раніше заявок згідно зі статусом які вони мають: Обробляється (Processing) та Прийнята (Accepted).
3. Можливість скачування звіту по проекту обраного замовлення у форматі PDF (Excel та перегляд діаграми Ганта у даній версії проекту неактивні).

Основні функції адміністратора:

1. При вході у систему адміністратора для нього висвічуються усі нові заявки замовників. При виборі однієї із них адмін опрацьовує її та створює проект на цій основі: обирає дати, назначає менеджера проекту та компанію, яка прийматиме замовлення.
2. Передбачена можливість створювати нового працівника та компанію.

Працівники ж поділяються відповідно до кваліфікації. Однак необхідно виділити менеджера проекту та всіх інших працівників (далі: працівник), незалежно від кваліфікації.

Основні функції менеджера проекту:

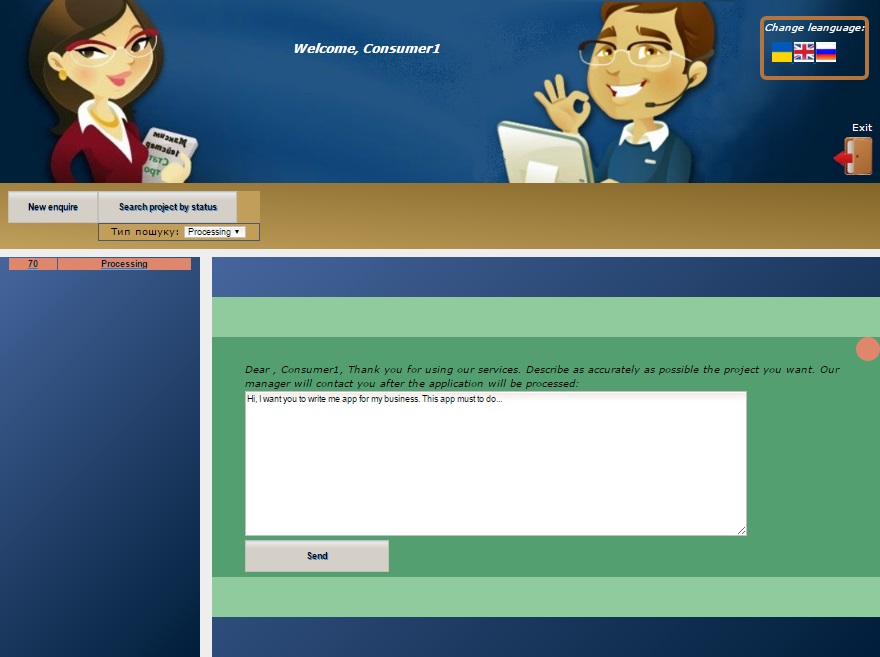
1. При вході у систему менеджера проекту для нього висвічуються усі нові проекти, на яких його призначили. При виборі однієї із них він переходить на сторінку керування, де може:
   1. Створення фаз проекту.
   2. Створення задач та встановлення приналежності до певної фази.
   3. Керування проектом. Встановлення для задач попередніх задач та працівників, які повинні їх виконувати.
   4. Керування задачами у ручному режимі(Menage Task (MM)). Можливість введення пізніх дат для задач (Пізній старт та пізнє завершення). На даний момент автоматичне введення не передбачено.
2. Передбачена можливість перегляду, квітування та видалення зі списку повідомлень від працівників про необхідність переносу дати завершення конкретної задачі.

Основні функції працівника:

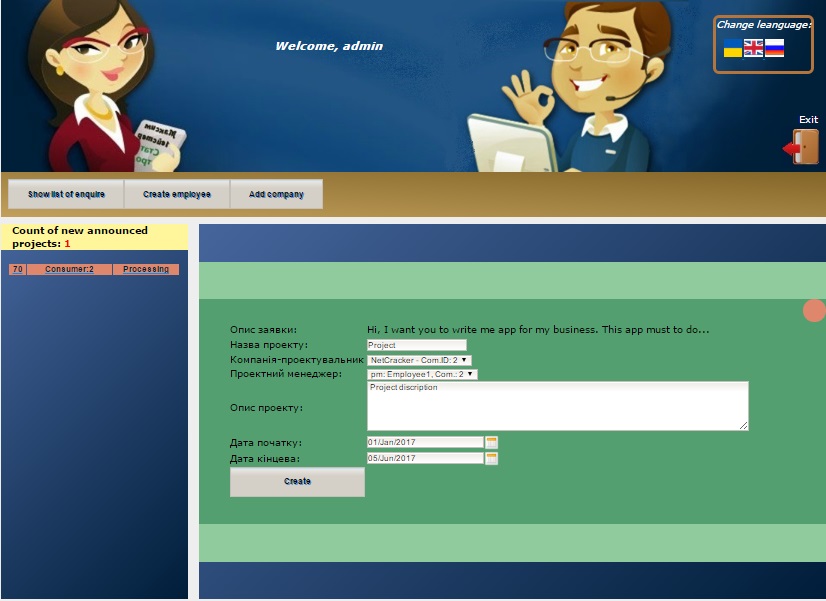
1. При вході у систему адміністратора для нього висвічуються усі нові задачі, однак можна переглядати усі задачі з його участю. При виборі однієї із них можна переглядати постановку та основний зміст, дати, статус та поточний прогрес виконання.
2. Передбачена можливість розпочати задачу, завершити виконання та змінити поточний прогрес виконання.
3. При необхідності змінити дату завершення задачі (перенесення строків задачі) є можливість подати заявку на її перенесення, де повідомлення зберігається у базу, а потім висвічується проектному менеджеру. А він у залежності від поточної ПІЗНЬОЇ дати завершення цієї задачі перегляне можливість переносу. Якщо перенесення неможливе, тоді у цьому випадку він повинен додати ще одного працівника на допомогу вирішення цієї задачі).

Приклади інтерфейсу залежно від ролі:

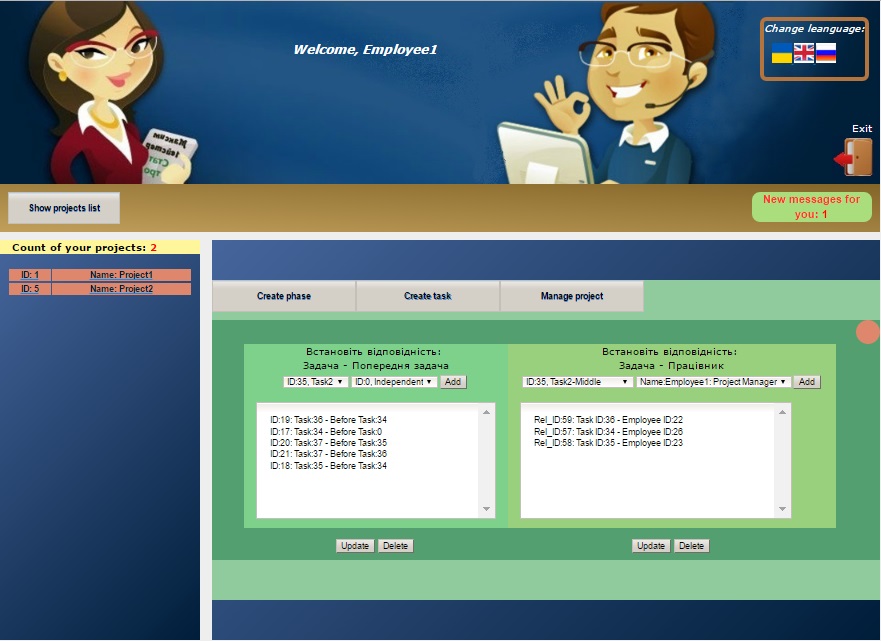
1. Замовник



1. Адміністратор



1. Менеджер проекту



1. Працівник



Додаткова інформація:

1. Структура проекту
   1. У папці з вихідними файлами знаходяться такі пакети:

* netcracker.project.web.beans – містяться класи із сутностями згідно таблиць в БД
* netcracker.project.web.controllers – допоміжні класи-контролери для робити із представленнями
* netcracker.project.web.dao – клас, для роботи із БД
* netcracker.project.web.enums – допоміжні класи
* netcracker.project.web.nls – конфігураційні файли для підтримки локалізації
* netcracker.project.web.operations – класи-контролери для різних ролей. Основне завдання - формування запитів у ДБ, відображення і збір даних із представлень
* netcracker.project.web.roles – міститься клас сутності User
* netcracker.project.web.servlets – сервлети для формування PDF, Excel звітів
  1. Робота із JSF виконана на основі шаблонного способу відображення представлень. Усі фрагменти та шаблони розташовані у папці «Web Pages/templates», основні сторінки для роботи розташовані: «Web Pages».
  2. Додаткові ресурси (css-файли, javascript-файли та картинки) містяться у папці «Web Pages/resources».

1. Підключення виконано за допомогою DataSource з використанням ConnectionPool на сервері-додатків Glassfish 4.1.
2. Недоліки та баги:

3.1) При запуску інстанса на демо-сервері версія серверу-додатків Glassfish 4.1.1, а у нього відомий баг – неможливість створити пул з’єднань(error 500), тому підключення до бази відбувається напряму через DriverManager. Тому існують певні баги із роботою програми. При натисненні кпопки на перегляд проектів, задач, заявок (у кожної ролі свої кпопки) у лівому меню нічого не відображається. Хоч на localhost усе працює і відображується коректно. У логах сервера жодних помилок немає. БД працює коректно.

Передивився логи браузера, там при натисненні кпопки ось така помилка:

Uncaught DOMException: Failed to read the 'selectionStart' property from 'HTMLInputElement': The input element's type ('submit') does not support selection.

at getTextSelection (chrome-extension://jgnebchahhepphmokjeohhoebakpfggp/data/content\_script/inject.js:443:30)

at getSelectedText (chrome-extension://jgnebchahhepphmokjeohhoebakpfggp/data/content\_script/inject.js:450:39)

at HTMLDocument.triggerTranslation (chrome-extension://jgnebchahhepphmokjeohhoebakpfggp/data/content\_script/inject.js:511:26)

Як я розумію це якісь проблеми з відображенням JSF-го тегу:

<ui:repeat>

<h:outputLink value="…" class="text"/>

</ui:repeat>

Конкретніше розібратись із помилкою не встиг.

3.2) У зв’язку із обмеженим часом виданим на проект і проблемами з локалізацією на localhost у БД дати проектів та задач зберігаються як поля VARCHAR2, а не DATE. Тому всі обрахунки із датами, виконуються на машині клієнта.

# Висновок

При розробці даного проекту було створено web-додаток для обліку та планування робочого часу на проектах. У ході виконання було використано такі технології JSF, Spring та сервер-додатків Glassfish. Перевагами JSF є красивий інтерфейс, широкі можливості масштабування сторінок та використання Expression Language і, як наслідок, відсутність прямого Java-коду на web-сторінках. Spring дає можливість підтягувати та впроваджувати наші класи, коли нам потрібно, без необхідності явної пров’язки до типу (лише інтерфейси). Це дає можливість реалізувати патерн Dependency injection та писати незалежний код. За допомогою Glassfish зручно налаштовувати різнотипні з’єднання до баз даних, можливість використання системи авторизації JAAS (у даному проекті не застосовується) та зручна навігація по сторінкам.

Виконав:

Каракой Андрій, КПІ, ТЕФ, 6 курс

Київ, 2017р

Посилання на вихідні файли проекту:

<https://github.com/Karakoi/PlanningApp>

Посилання на працюючий екземпляр проекту:

<http://env-9826035.mircloud.host/Karakoi/>

Скайп:

karakoy199405

Телефон:

(067) 11-40-801