King's College London

University of London

6CCS3PRJ Final Year Project
Task Allocation System

Third Progress Report

Author: Hussain Miah

Supervisor: Dr Tomasz Radzik

Student ID: 1114801

Date: 10th November 2014

Progress

The bulk of the project work as been completed. Where there is a fully functioning database system, providing the query and manipulation functionalities.

I have implemented a **Task Allocation** file. This is where the user will be able to save the suggested allocation to a file before they have it committed to the database. The reason why I chose to make this file, is to allow the user to compare two different allocation methods. Giving the user this functionality allows them to fine tune the matching process. Where they can choose different matching algorithms or decide they would like the application to check if the employee is still assigned to a task, and if the new task will fit in with the employee's schedule. The next step would be to implement a new compare view to present the two suggested allocations to the user.

I have implemented a **Greedy** and the **Hopcroft-Karp** algorithm as my matching algorithms.

Greedy

The greedy algorithm that I have implemented, assigns the first employee that has the correct skills for the task. After this task has been assigned to the user the algorithm will move onto the next employee and remove the assigned task from the set.

Hopcroft-Karp

This is the algorithm that I detailed in my Background and Specification Report. After further investigation and implementing this method I noticed that the algorithm does not backtrack to find new allocations and in some occasions will not lead to the maximum matching for the graph, for example if the algorithm is given the following bipartite graph:
E1 —> T1
E1 —> T2
E2 —> T1

The algorithm will assign E1 to T1 and then move onto E2, once that has happened the algorithm will see that the we have already visited T1 and move onto the next node, which in this case will be the sink (not shown for simplicity) and the algorithm will terminate, which does not lead to the maximum matching for the graph.

Advanced Options

There are number of advanced options the user will be able to specify when they are choosing which matching algorithm they would like to use. The advanced options are:

- Check if employees are assigned to tasks

  This option will check if employees are assigned to a task within the database, the algorithm will then check to see if the start and end dates of the assigned task and the new task clash or not.
- Order of the Employees
  - Name Alphabetical
  - Name Reverse Alphabetical
  - Cost Ascending
  - Cost Descending
- Use Heuristic, this option can only be used with the greedy matching algorithm