WESTMINSTER
INTERNATIONAL UNIVERSITY IN TASHKENT
An Accredited Institution of the University of Westminster (UK)

Module name and code	Fundamentals of Programming, 4BUIS008C
CW weighting	60%
Lecturer setting the task with contact details and office hours	Vasiliy Kuznetsov vkuznetsov@wiut.uz
Submission deadline	Mar 30, 2017
Results date and type of feedback	Written Apr 13, 2017
The CW checks the following learning outcomes:	
3. have theoretical foundations of information systems and trends in application of information systems 4. understand and implement fundamental data structures using classes 5. understand and implement the concept of persistence in relation to objects 6. comprehend some of the classical algorithms for sorting and searching	

Task

CW Task

WeCanFixIt is a repair service agency operating in your city. Clients of WeCanFixIt send requests for particular types of workers (plumbers, painters, furniture makers) and the agency is searching through its records to find the particular people capable of doing the job. Moreover, according to current legislation, a person can work only particular number of hours a month and therefore the company should keep track of hours worked in the current period for all of its workers not to assign a person to do extra hours. The maximum number of hours to work per month is set to 180.

Currently the company keeps all workers' profiles in paper files and therefore searching for suitable workers is very time consuming. The company hires you to implement a simple personnel management system to facilitate its activities. The system should allow easy maintenance of electronic profiles of company personnel as well as keep track of hours worked each month by each employee.

You are required to design a database to contain all workers of WeCanFixIt. The database should have the following structure:

Table wo_worker – holds workers' information

wo_id	Artificial Id for the table (autonumber, primary key)
wo_name	Employee name
wo_skills	Comma separated list of skills the worker possesses (e.g. "painter, electrician")
wo_hours_worked	Total amount of hours worked in the current month
wo_rate	Hourly rate of the worker

Table jo_job – holds information on job requests from company clients

jo_id	Artificial Id for the table (autonumber, primary key)
jo_date	Date of the order
jo_work_type	Worker type required (only one type per job e.g. “painter” or “plumber”)
jo_amount	Number of workers required
jo_hours	Number of hours needed
jo_total	Total cost of the job

Based on a series of interviews with employees of the company you have identified the following requirements for the system to be developed:

1. Manage workers information
   - Display all workers of the company
   - Enable addition/editing of workers’ records
   - Enable deletion of workers’ records
   - Persist workers information in the database
2. Facilitate workers management
   - Allow sorting the workers by Name and Hours worked
   - Allow fast searching for a worker by Name
3. Manage job requests
   - Provide a form to register new job request (no need for edit and delete functionality)
   - Persist jobs in the database
4. Job request composition
   - As a part of job creation form a user should be able to generate a list of workers for the job. You should find workers suitable for the job and list them on the screen according to the following criteria:
     i. Employee should be capable (as stored in wo_skills column) to perform the work requested (as stored in jo_work_type column). Please note that a worker can possess many skills (stored as comma separated list) while a job request will contain only one type.
     ii. Employee should have enough hours left. As stated above, a worker can work a maximum of 180 hours a month. Therefore, an employee should be allowed for the job only if the sum of currently worked hours (wo_hours_worked column) and hours requested (jo_hours column) is less or equal than the maximum hours allowed per month.
     iii. Business rule states that all employees should be given equal opportunity to gain highest income and therefore you should queue the employees for jobs based on the amount of hours worked putting the employees with lowest number of hours worked first. For example, for a job requesting 3 plumbers for 15 hours the employees should be listed as following:

Name	Skills	Hours worked	Explanation
Jorah Mormont	plumber, painter	50	Listed first – lowest amount of hours worked
Cersei Lannister	plumber	70	Listed second – more hours worked
Theon Greyjoy	electrician, plumber	100	Listed third – more hours worked
Roose Bolton	plumber	115	Not listed – only three workers needed
Catelyn Stark	furniture maker	40	Not listed – not a plumber
Margaery Tyrell	plumber	170	Not listed – not enough hours left

- All workers selected for the job should be shown on the form. You should display name, skills, rate and current amount of hours worked.
- The form should also display the total cost of the job calculated as sum number of hours requested multiplied by the rate of workers listed for the job.
- The form should notify the user if there is not enough workers for the job requested (number of workers satisfying all the conditions is less than the number of workers requested).
- You are **NOT** required to persist (save to database) information on workers assigned to a job.
- Whenever users save a job, you should update the number of hours worked for all assigned employees (current number of hours worked + number of hours for the job just saved).

5. Start new period
    - Whenever new month starts, the number of hours worked for all employees should be reset to zero. You should provide such functionality in your application. It can be a main menu item or a button on one of the screens.

Overall the recommended structure of the application is the following:

- Parent form – should contain main menu and host all other forms
- All workers – list form to show all workers of the company. The form should contain controls for searching and sorting as well as buttons to manage workers (Add new, Edit and Delete). Start new period button should also be placed on this form.
- Add/Edit worker – details form to create new or modify existing workers.
- All jobs – list form to show all job requests. You should not show any controls for sorting or searching. Only Add new button is needed on this form.
- Add job – details form for adding new jobs. Job request composition controls should be placed on this form.

Important notes: you are allowed to use standard .Net mechanisms for sorting, searching as well as use Queue, Stack and other such classes. However in order to gain full marks you should provide your own implementation of at least one of the algorithms and your own implementation of a data structure. Be sure to use object-oriented approach as well as comment you code properly.

Even more important note: you must change the names of all columns to be of the following format:

{column name as it is stated in the task}_{your ID without leading zeroes},

so that if your ID is 00001111, “wo_id” becomes “wo_id_1111”, “wo_name” becomes “wo_name_1111” etc. Failure to do so will be considered as an attempt to plagiarise. However, the properties and names of your business classes should be ***normal*** English words e.g. “Worker”, not “wo_worker” and “WorkerName”, but not “wo_name_1111”, so they should not include your ID or prefixes like “wo_” and “jo_”.

Format

1. There is no need for any kind of written report for the coursework.
2. All you have to submit is your code (complete Visual Studio project folder including database file)
3. Use Harvard method of referencing if you use someone else’s code.

Assessment criteria

Component	Weight (marks)
General requirements	**15**
The program compiles	3
Comments present	5
Classes created for business logic and DAL	7
Required functionality	**65**
Display all workers registered	5
Enable addition/editing of workers' records	5
Enable deletion of workers' records	5
Persist workers information in the database	5
Allow sorting the workers by Name and Hours worked	10
Allow fast searching for a worker by Name	5
Enable creation of new job requests	5
Persist job information in the database	5
Job compilation	10
Update worker hours worked (after saving a job)	5
Resetting hours worked for all workers to zero	5
Advanced functionality	**20**
Custom sort/search	10
Custom data structure	10

Submission

Submit CD (hard copy) to the designated place to the WIUT Registrar's Office and an electronic format to WIUT Intranet.