Software development 3



|  |  |
| --- | --- |
| Due date: | 8 June 2015 |
| Value: | 30 % |

Objectives

**The course objectives met by this assignment include:**

• Course specifications are available online from the USQ website <http://www.usq.edu.au/course/specification/>. Always check the website for the latest version.

**Please read carefully:**

• This assignment must be your own work.

• It is acceptable to discuss course content with others to improve your understanding and clarify requirements, but **solutions to assignment questions must be done on your own**. Make sure that you do not discuss your solutions or show your solution to anyone to protect yourself.

• **You must not copy from anyone**, including tutors and fellow students (including StudyDesk), **nor allow others to copy your work**.

• **Assignments that do not adhere to this requirement will be deemed to be the result of academic misconduct.**

• **Submit any work you have done by the due date, even if it is incomplete or does not work. Our assignment feedback is constructive so we will provide you with suggestions so that you can do better next time.** Submitting incomplete work is better (you get partial marks but more so also feedback) than not submitting at all. For this reason, the percentage of this assignment is low to encourage you to present your work to make sure you are on the right track.

This assignment specification provides you with [How you need to write your business application](#_How_you_need_1) and [The program specifications for your business application](#_The_program_specifications).

# How you need to write your business application

You are required to write a new business application/solution in accordance to the business requirements outlined in [“The program specifications for your business application”.](#_The_program_specifications)

## Business and GUI layer

This business solution will be written with a clear separation between the GUI (Graphical User Interface) and business layer class definitions. The GUI layer classes will be solely responsible for the interaction with the end user, while the business layer classes will contain the relevant data and processing required on that data.

Both layers require class definitions to be written in accordance with the course applying all the object-oriented principles that this course covers using the guidelines of module 1 and 2 under the “Link to course contents” on the study desk and as per the experience you have gained from the weekly exercises also posted on the study desk.

## Business layer class definitions

In your software development 2 you already provided the business layer classes for which you also received feedback. As you progress through the rest of the course material your skills of implementing the course material matured and this will be reflected in your class definitions. In other words, your business layer class definitions for software development 3 would have changed as a result of your learning progress and applying the feedback from the software development 2.

## GUI layer class definitions

In your software development 2 you also supplemented the GUI layer classes with one or more class definitions to communicate with the console. For this assignment, you need to replace the console with the relevant new set of GUI class definitions. A clear separation of these two layers will make this process easier, as the communication process should be similar - calling a different object’s method. For example, instead of calling an object of class type Console you would be calling an object of class type GUI.

## Class definitions

You must apply the instructions for the class definitions provided in the software development 2 specification (Part B, heading “Class definitions”) to the design of your class definitions for this software development 3.

## Scope of the business solution

While it is admirable to always want to cater for everything in a new application, organisations are limited by time and money, and it is essential to listen to the business needs as outlined in the specification and develop a solution that will address these needs. You are specifically requested not to add any features or functionalities that are not described here.

As a third year student you already have the skills to follow software development specifications. Part of providing the business solution is the inherent skill of carefully reading, re-reading and understanding the specification to then translate into the correct solution. This also includes, that some of the information is not specifically stated but is inherently required in the solution (such as reports). You need to be pro-active in identifying this and obtaining the information so that you can cater for it appropriately in your solution.

Therefore, you need to ensure that the functionality you are going to provide is a) specified directly or b) an inherent requirement to satisfy one of the directly specified requirements.

When designing and developing your business solution, you need to incorporate prior assignment feedback, and implement the solutions in accordance to the material of this course, which includes the modules in the link to the course content, lecture slides, and weekly exercises. The textbook is your reference to the theories used in this course and contains examples of implementing object-oriented principles, which are overridden by the CIS3001 course material (particular module 1 and 2 of the course content).

## Please note that

• You must not copy any source code from the text book. You must not use any of the source code provided to you in the weekly exercises posted on the study desk, or nor copy any source code from the internet or any other source. If you have any questions regarding this assignment, please contact your course leader per email.

• You must write your own source code to show that you understand the course material and show that you can apply it.

• If you have any issues, such as bugs, incomplete features or processing by the time that you are submitting your assignment, it would be beneficial for you to list them with a short explanation in a separate word document.

## Style and coding conventions

As this is a third year course, your source code must be written clearly, logically and concisely at a high level of proficiency for a third year programming course. It is expected that you review your source code to reduce duplication, use smaller methods that have one simple task to perform. Please ensure that your source code is correctly indented.

In regards to naming conventions, please refer to the programming standards published on the study desk for this course.

These are simple steps to ensure that you are not losing marks.

## Object oriented design and implementation

You must implement the appropriate object oriented methodologies by using the coding conventions in this course. The study book (modules 1-12 found on the study desk under “Link to course content”), lecture slides, and weekly exercises posted on the study desk provide you with guidelines.

For your software development 3 you are required to implement object-oriented principles, specifically as outlined in all the CIS3001 course material.

## File handling

Your solution needs to cater for the following implementation.

|  |  |
| --- | --- |
| **Type** | You are only allowed to use **text files**. Do not use any other file type. |
| **Location** | You are required to use **JFileChooser** in your solution to obtain the directory from the person running your application. |
| **Layout** | You need to determine how many text files your solutions needs and what data should be stored in these text files. You need to name the text files accordingly using a meaningful name that reflects the contents. You need to provide documentation about this at time of submission. |
| **Reading** | When the applications starts – all files must be read and the data be retained in memory (objects in collections).  The data from the text files is to be validated in the same manner that data entry is validated.  You need to create the appropriate objects that contain and process the data and store these objects in the appropriate collections in the same way as you would for data entry.  You must **only use the Scanner class** – NOT the BufferedReader class. |
| **Saving** | When the application terminates, all data stored in memory must be written to file.  You must not write to file at any other time but application termination. |

## Collection(s)

You are requested to use an appropriate collection to store your objects. **You must use only Maps and/or Lists for this purpose**. Arrays and Vectors are not permitted for this solution.

At the **start of the application, the data is to be read from file** into the relevant collections. Any change to the data during data entry must only occur in memory to the relevant collections. The contents of the collection(s) are then to be **written to file only when the application terminates**.

You need to show your understanding of Maps and Lists by selecting the appropriate type depending on the content, efficiency of use, ease of maintenance, and performance.

Vectors and arrays are not permitted for the storage of objects.

## Reports

**Reports screens**

There should be an option on the main screen where the user selects 'Reports'. This must lead to the opening of another window, which then lists all the report options. Depending on the requirements for the requested reports in the specification, you then need to design accordingly obtaining relevant input from the user.

The bottom of the each report should have two buttons, one to close the report screen and one to save the report to a file. The save report button should then use **JFileChooser** where the user enters the path and file name for the report to be saved to. After the report has been saved a message of the success or failure of the operation must be displayed to the user. Closing the report should take the user back to the window that shows the list of reports, which also contains a close button. If this close button is clicked the user should be returned to be main screen.

**Formatting report output**

All reports must be correctly formatted, numeric columns aligned to the right, text columns aligned to the left, amounts correctly positioned and formatted. Data must NOT be truncated in the display, and be in the correct order as requested. Each report must also have a report heading that clearly communicates the meaning of the report.

The reports must be professional so that you would feel comfortable to show it to the CEO of the organisation. All names must be in appropriate case, all currency fields must have 2 decimal places (assume dollars as currency), all columns correctly left or right aligned depending on the data.

All reports must be displayed in a window first, with appropriate heading, and a button to allow saving the report to a text file must be provided.

## Data validation

**Exception handling**

You must implement Exception handling as specified in the course material and where appropriate.

**Data entry and reading from file**

Data entered by a user or data read from a file must be appropriately and professionally validated at the business layer not the GUI layer. This will required the object containing the data to execute its validation methods.

**Error messages**

You must provide user friendly error messages that users are enabled to resolve issues where they can and clearly communicate the problem to them, as well as providing guidelines to resolving these problems. For this purpose you are required to use dialog boxes.

As the developer you would know which issues cannot be resolved by the person running your application, and therefore you might then provide advice who the user should be contacting and what information they need to provide to that person. In this situation your error message must contain the details of that advice.

## Data Entry

**JFrames**

All windows must be classes that inherit from JFrame. You must NOT use dialog box for data entry.

You must apply appropriate layout managers to position your components. It is essential that you use at least one **JComboBox** and at least one scrollable **JList**. You must use labels, text fields, and buttons.

Your design must be user friendly and well balanced, with positioning of buttons in a consistent manner on all frames. Your labels must communicate well what the data is to be entered and should provide examples where appropriate.

The main screen contains a number of SWING buttons which will launch other screens required for data entry and reports. As new screens are opened the launching screen will be inaccessible to the user.

The business application should consist of multiple sub screens (frames) which are to be opened and closed appropriately. The main screen is to remain on the screen at all times.

**JDialog**

Use dialog boxes only to communicate error messages, do not use it for data entry.

## Starting the business application

As described in more details above, when the business application is started the user must first be prompted to select the appropriate file directory using **JFileChooser**. This directory is then used to read all the relevant files. All file content must be validated and stored in the appropriate objects which are then added to the appropriate collections (see above for more details) so that the data read from file and all the data entered while the application is running is stored in memory – an only written back to file when the application is terminated.

After the files are read and all data is stored in memory, the business application is then going to display a main window which lists the options a user can choose from. Each option is represented by a button, and all buttons need to centre in the middle of the window.

The specifications for software development 2 provided you with instructions on the type of classes that you should have and details about the class definitions. You need to ensure that you re-read this particular section of the software development 2 specifications.

## Terminating the business application

The only way the user can end the business application is by clicking on the Window close button top right corner of the window or selecting the "Close Application" action. When this occurs, all data currently stored in the collections must be written to the relevant files. The files directories details are recorded when the business application was opened and information was read into the business application. A message must be displayed to the user "Files have been updated". This is the only time that the business application will write to the data files.

# The program specifications for your business application

## Business background

You are required to write a business solution for company XYZ. This company requires an application that keeps track of the software they have purchased, the licence details and updates for the software, and the employees who are using the software and corresponding software licence.

Employee details would include the usual name, address, and other contact details (phone, email). An employee’s computer can have more than one software installed, and have multiple licences applied depending on the number of years the employees has been employed.

Software details would include the company name that sells the software, the name of the software, address details of the company (to obtain licence updates), the version number, any hotfixes, upgrades, etc.

Licence details would include the licence number, expiry date, which version the licence is associated with, type of licence (site licence, desktop, etc.). You may need to investigate the types of licences of the relevant software.

The company would be using the typical base software such as operating system, word processing, spreadsheet, email, financial planner. In your design you need to cater for at least applications, and design your solution so that the company can add other software they will be purchasing in the future. The design, while custom build, must be flexible to cater for any software and any licence type.

The application will be specifically custom build and only address the needs out lined in this specification, including the requirements of the reports.

Using the information provided to you under the heading [How you need to write your business application](#_How_you_need), your business solution must cater for the data entry of the program and course details by writing the relevant GUI classes.

Any data already on file as a result of you running your application will be available to the user to select from a list. The user should be able to add a course or delete one from the list.

To test your data entry page you should start with an empty file, and then enter the data to test your design and validation, then save the result to file at termination of the application.

## Reports

The reporting requirements below will determine which other data you need to capture in your data entry. You should find all the answers in the relevant course specs and links provided above.

* Report on all employees listing the software they are using sorting by employee by software
  + List the licence number, licence type, licence expiry date for the software for this employee and list with oldest licence first
* Cater for a dropdown list of all software, then allow user to select a software
  + List all the software details, such as name, type of software, upgrades, hotfixes, including the list of licence number and expiry date that the organisation paid for over the past years, and the type of licence (desktop, site licenced, etc.)
* Cater for a dropdown list of all employees, then allow the user to select one employer
  + List all the software that this employee is using showing the licence expiry date
* Report on all licences that are due within the next two months.
* Report on software that is allocated to employees who are no longer employed by the organisation.

Your solution must be written in accordance to the course material and you will find further details under [How you need to write your business application](#_How_you_need), including the instructions for “Class definitions” from the specification of the software development 2.

Once you have completed your development and testing of your business application, you need to provide a 10 minute power point presentation that clearly shows and demonstrates your working application as you run it, using screen capture and voice over.

# Submission requirements

Create one zip file (studentnumber\_studentname\_coursenumber.zip) that needs to contain the following.

Submit your zip file to the assignment submission link on the study desk.

Your zip file must contain:

* All .java source code files of your business application (you can use the src folder in the NetBeans directory) so that the marker can run your application.
* All text files and report files.
* Word document with appropriate headings and formatted professionally that contains:
  + Print screens of all your error messages displayed to the user
  + Print screens of the files before you run the application containing some but not all of the programs, majors and courses
  + Print screens of the files after you added the rest of the programs, majors and courses
  + Description of what data is stored in which text file including the name of the relevant text file
  + Print screens of the data entry while entering data, menus, reports, etc.
  + Print screens of all saved reports
  + List of bugs or missing features and your suggestion to fix them. Volunteering this information will provide us with more information on your design
  + Any assumptions you have made.
* Power point slides with voice over.

