MURDOCH UNIVERSITY

ICT373 Software Architectures
TNE, 2024
Assignment 2 (25%)

Due Date: 10/11/2024

All Students: Submit the Assignment via LMS by the due date.

Late penalty: 10% per day penalty for delayed submissions unless prior extension of deadline is obtained from the unit coordinator.

You should keep a copy of your work. Your submission must include a completed assignment cover sheet. An electronic copy of the assignment cover sheet is available at the unit LMS site.

This assignment consists of one question only.

Question [25 marks]

Design and implement (in JavaFX) a basic graphical user interface (GUI) program for recording information about a Magazine service. In addition, you will need to demonstrate persistence, i.e., store the data in file(s) and re-use them later.

This Assignment is an extended version of the work that you did in Assignment 1[1]. I.e., Each week, the main part of the magazine is made available to each customer along with some extra supplements as chosen by the customer.

- A customer has a name, an address, an email address and a list of supplements which they are interested in. An address has a street number, street name, suburb and postcode.
- A paying customer has a payment method and a list of associate customers whom they also pay for.
- An associate customer is a customer whose subscription is paid for by a specified paying customer.
- A payment method could be by a specified credit card or direct debit from a specified bank account.
- A supplement has a name and a weekly cost.
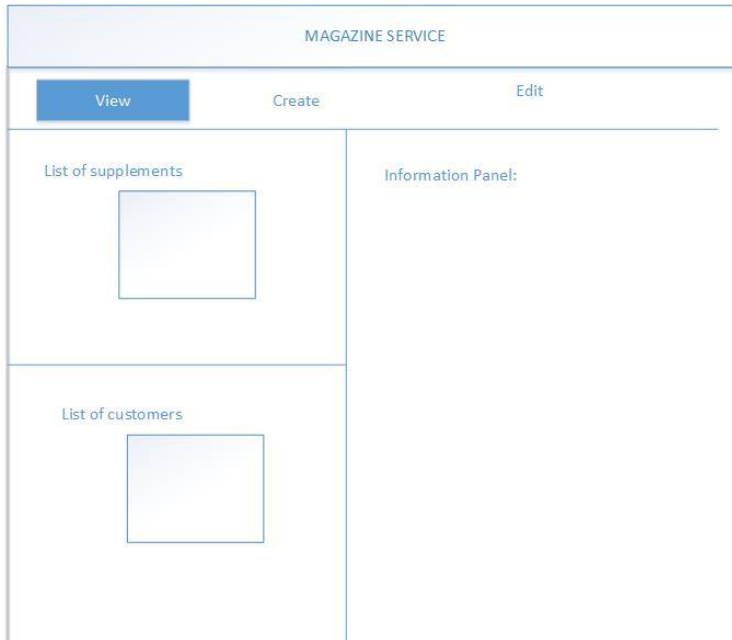- The magazine also has a weekly cost for the main part.

Home screen:
The program can be started in create mode, edit mode or viewing mode.

---

[1] You may need to use some information from Assignment 1.

Viewing mode:
i) The GUI shows the list of supplements and the list of customers.
ii) If a supplement is clicked on, relevant information e.g., name, cost, date, subscribed customers should be displayed on the information panel. A sample example is given below:

iii) If a customer is clicked on, relevant information e.g., name, address, list of subscriptions, status (paying or associate), the list of associate customers (if paying customer) should be displayed on the information panel.
Please also show the billing history of a paying customer. Attempt to use multithreading for this.

Edit mode:
Should allow to edit existing information about the supplement and customers. It includes: correction of information, adding a customer (either paying or associate), deleting a customer, adding or deleting a supplement. In case of adding an associate customer, the paying customer can be selected from a dropdown list of existing paying customers.

Create mode:
In this mode you start by adding a Magazine. Then you can add supplements and customers.

The software should allow a user to conveniently enter the data and have it stored for subsequent editing and viewing.

Set up a demonstration Magazine service, which you submit and give clear instructions in the User Guide for use.

Please note, you need to use the major items such as serialization, JavaFX components, FileChooser, and Exception handling (that we have covered in the unit),

You may consider using TreeView, TreeItem of JavaFX. (Optional)

Required Documentation for the Question

Please remember to submit the Java source code and executable version of your program (i.e. the whole NetBeans project). NetBeans is our preferred IDE.

For internal documentation (ie in the source code) we require:

- a beginning comment clearly stating title, author, date, file name, purpose and any assumptions or conditions on the form of input and expected output;

- javadoc and other comments giving useful low-level documentation and describing each component;

- well-formatted readable code with meaningful identifier names and blank lines between components (like methods and classes).

Required External Documentation (One file)

- **Coversheet:** A signed Assignment cover sheet.

- **Title**: a paragraph clearly stating title, author, date, file names, and one-line statement of purpose.

- **Requirements/Specification**: a paragraph giving a more detailed account of what the program is supposed to do. State any assumptions or conditions on the form of input and expected output.

- **User Guide**: instructions on how to compile, run and use the program.

- **Structure/Design**: Outline the design of your program: describe why you chose one approach rather than other possible approaches. Give a written description. Describe how the classes relate to each other. Describe each class: its purpose, list of methods. Use diagrams, especially UML and STDs. Include a complete description of the design. (Please also include pseudo codes for three of your interesting or complicated algorithms.)

- **Limitations:** Describe program shortfalls (if any), e.g., the features asked for but not implemented, the situations it cannot handle etc.

- **Testing**: describe your testing strategy (the more systematic, the better) and any errors noticed. Provide a copy of all your results of testing in a document saved in Word format. Note that a copy of the sample data and results from a test run of the program is required (copy from the program output window or the GUI windows and paste to the Word file). Your submitted test results should demonstrate a thorough testing of the program.

Source code: Save all your java source code files in appropriate names and directory structure. You must include them in the submission.

Note that all of the above external documentation must be included in a file saved in pdf format.

The external documentation together with all relevant files must be compressed in a .zip file before submitting. Make sure that all necessary files are submitted so that the program can be viewed, compiled and run successfully. Note that the whole NetBeans project folder can be zipped.

Warning: Do not use visual tools e.g., bean builders or any type of code generators for your GUI design. You need to write code. [Heavy penalties apply]. However, FXML is allowed.

Remember to complete and sign the Assignment Cover Sheet and submit it with your work.