Lab 5: Bilingual App (GUI & File Streams)

Object Oriented Programming 1 – Fred Stiebler

Program Description

In this lab you will **write a bilingual Java GUI App** where the user can choose between displaying the app in **English** or **French**.

⚠ **All strings in the app must be loaded from external language files using the scanner**.

Language Files

You will **create 2 language files**, one named `English.lang`, and another named `French.lang`.
The content of these files must be as shown in the table below:

English.lang	French.lang
`Bilingual App`	`App bilingue`
`Welcome to Durham College!`	`Bienvenue à Durham College !`
`Enter your name:`	`Entrez votre nom :`
`Enter your program name:`	`Entrez le nom de votre programme :`
`Register Student`	`Inscrire un étudiant`
`Please answer all questions!`	`Veuillez répondre à toutes les questions !`
`Warning!`	`Avertissement !`
`Registration Complete!`	`Inscription terminée !`
`Student registered in file:`	`Étudiant inscrit dans le fichier :`

Properly Displaying French Accents

Java 17 and older do not use UTF-8 encoding by default, which makes it **not possible to display accented letters** e.g. the **ç** in Français. To **fix this problem** we will have to manually **set the encoding to UTF-8 when creating a Scanner object to read text from files**: `new Scanner(file, "UTF-8");`

Language Selection Screen

Allows the user to choose their language, it's the **only screen that has** "string literals" in both English and French, because **we do not know yet what language the user speaks**.

1. Set the title and ask the user to choose a language
2. Display the language options (English or Français)

   ⚠ If the user ☒ closes the window without choosing a language, then it will **default to English**.

Student Registration Screen

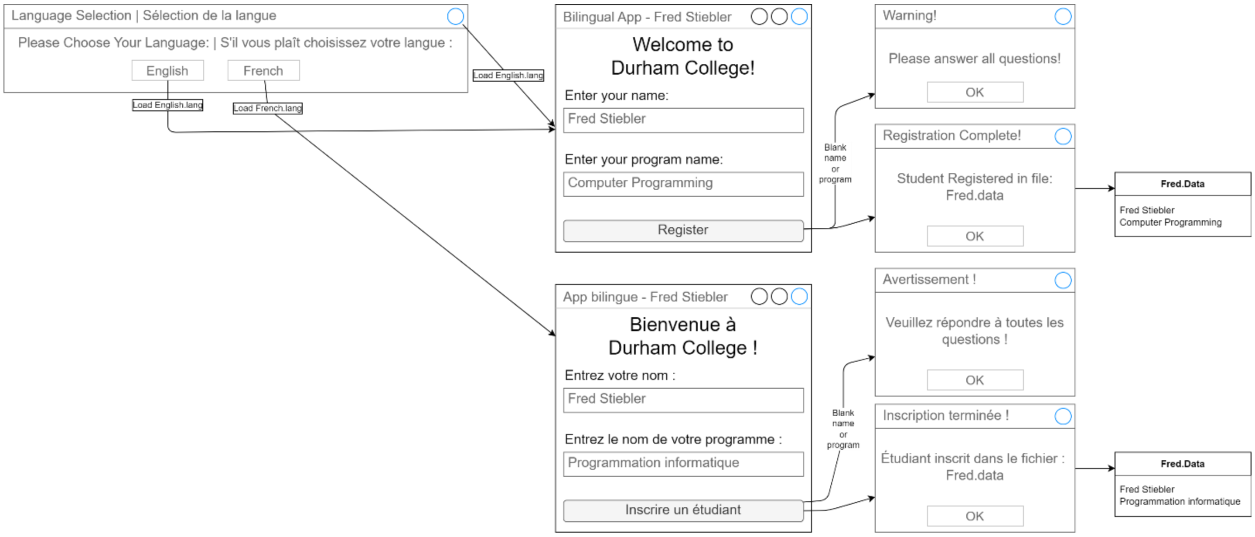
3. Open the selected **language file**
4. Prompt for the student's name and what program the student is enrolled
5. Save the student's data in a **new data-file**
   - Data-file **must be named in this format**: Name.data

     E.g. A student **named Fred** will be saved in a **file named** Fred.data

     ⚠ **Optional**: Extract just the first name: Fred Stiebler → Fred.data
   - **Replace any existing data-file** with the same name
   - Write the student's name and program in the data-file
6. **Pop-up** telling the user the registration is complete, and their data was saved in Name.data
7. **Clear the form** so that the next student can be registered

Mock-up

Try to match this as close as possible:



Language Selection | Sélection de la langue

Please Choose Your Language: | S'il vous plaît choisissez votre langue :

English French

Bilingual App - Fred Stiebler

Welcome to Durham College!

Enter your name: Fred Stiebler

Enter your program name: Computer Programming

Register

Warning!

Please answer all questions!

OK

Registration Complete!

Student Registered in file: Fred.data

OK

Fred.Data

Fred Stiebler
Computer Programming

App bilingue - Fred Stiebler

Bienvenue à Durham College !

Entrez votre nom : Fred Stiebler

Entrez le nom de votre programme : Programmation informatique

Inscrire un étudiant

Avertissement !

Veuillez répondre à toutes les questions !

OK

Inscription terminée !

Étudiant inscrit dans le fichier : Fred.data

OK

Fred.Data

Fred Stiebler
Programmation informatique

Style Guide and Documentation

To be eligible for full marks on this or any lab in this course your application must conform to the requirements as outlined above and the course Style Guide, in this case making sure to include:

- Your code follows our Java style guide.
- **Appropriately declared data types for all possible variables and constants.**
- Appropriate and complete **program documentation** (code comments).

⭐ Bonus Marks Opportunity (up to 5%)

Update your GUI to:

- Ask the user to enter their **student number** in a **spinner**.
  - This number will be used as **file name instead of their name**
  - It's a **unique ID**, as less likely to cause problems when saving student records.
- Add a **Load button** beside the **spinner**.
  - **When clicked**, it will check the entered student number and **try to load the student records** from a **file named** studentNumber.data
  - If that file exists, then **automatically populate all remaining student's information**.
  - Otherwise, just **show a pop-up** saying that there are **no records of a student with that number**.