## Complete the Tasks

Make sure you take notes as you code while completing the tasks. These notes will help you to answer the questionnaire once you have completed all the tasks.

## Warm Up Scenario:

1) Add a new `Feature Mapping` file within the `Item` package named `.feature-to-folder`.

    - Ensure that the feature *`Data`* is defined in the Feature Model using the `Feature Model View window` found at the bottom left.
    - Map the feature *`Data`* into the new `.feature-to-folder` file to create its mapping.

By completing these steps, you've successfully mapped the feature *`Data`* to the `Item` package.

2) Add a new `Feature Mapping` file within the `IO` package named `.feature-to-file`.

    - Ensure that the feature `Membership` is defined in the Feature Model using the `Feature Model View window` found at the bottom left.
    - Map the feature `Membership` to the file `Controller.java` into the new `.feature-to-file` file to create its mapping.

By completing these steps, you've successfully mapped the feature *`Data`* to the file within the `Item` package.

3) Navigate all the usages of the feature *`Search`* using the Feature Model.

    - To ensure exactly where the features have been annotated in the source code. Use the `Feature Model View window` at the bottom left, by right-clicking on the feature *`Search`* and selecting the `Find Usages`.

By completing these steps, you've successfully found the feature *`Search`* used in all places `6 results`.

4) Adding a feature to code mapping.

    - The feature *`Review`* is implemented in the `Rentable()` class. Your task is to identify the corresponding code fragment and annotate it with the feature *`Review`*.

      *Hint: You will need a begin and end annotation for the feature annotation.*

## Scenario 1: Task*: Importing Data from Text Files*

Task: Implementing ImportData Feature

Objective:

Your task is to add a new feature called `DataLogging` to the DARTPlus system, specifically implementing the `ImportData` functionality. This feature will enable the system to read data from the text file `dartData.txt` containing information about employees, songs, games, and customers, and register them accordingly. Please note that employees and customers have dedicated array lists (employees and customers), whereas rentable items (songs and games) can be added to the same array list (i.e., items).

Integration Steps:

1. Feature Addition:

- Open the feature model of the DARTPlus system.

- Add a new feature `DataLogging` to the `feature model`.

- Create two sub-features under `DataLogging` named `ImportData` and `ExportData`.

2. Code Implementation and Annotation:

- Navigate to the class `Manager` in the directory `Persons` in the DARTPlus source code.

- Implement a method named `readFile()` within the `Manager` class.

- Design the method to read data from a txt file (called `dartData.txt`) and parse each line to extract relevant information about employees, songs, games, and customers.

Note: The `dartData.txt` shown in the end of the Task

-Register the Employees in array list **`employees`**, Songs and Games in **`Items`**, and Customers in the **`customers`**.

**Help:**

For file reading, you can use BufferedReader:

```
File dartData = new File("dartData.txt");
FileReader fr = new FileReader(dartData);
BufferedReader br = new BufferedReader(fr);
String line;
while((line = br.readLine()) != null) {
   String[] dartInfo = line.split(";");// parse data
//register employees, songs, and games based on the first word of
the line (employee, game, song)
//For registering, use methods this.registerEmployee(),
items.add(game or song), and customers.add()
}
```

- You are supposed to **annotate** the implemented code with embedded feature annotations.

3. Menu Option Addition:

- Navigate the Controller class.

- Add a menu option for data importing in the manager's menu options within the Controller Class. For that, go to the `managerLoggedIn()` method in the `Controller` class, and add a print statement adding a new option (`System.out.println("b. Import data from text file");`)

- You are supposed to **annotate** the implemented code or Menu Option with embedded feature annotations.

- Lastly, put a new case in the body of the `managerLoggedIn()` method that calls the `readFile method()`.

Expected Outcome: By completing this task, you will enable the DARTPlus system to import data from the text file, enhancing its functionality and data management capabilities. Remember to annotate the code appropriately to maintain consistency between the feature model and codebase.

A sample of `dartData.txt` would look like this:

```
Employee;Dwight Schrute;1989;Abc Street;2000;2500
Song;Waka Waka;Shakira;2;2020
Game;Mario Bros;Platform;3;2018
Customer;Creed Bratton;abc123
Employee;Michael Scott; 1990;Def Street;2100;2700
Song;Shape of You;Ed Sheeran;4; 2017
Game;The Legend of Zelda;Adventure;5;2017
Customer;Angela Martin;password123
Employee;Pam Beesly;1992;Ghi Street;2200;2800
Song;Despacito;Luis Fonsi; 6;2017
Game;Red Dead Redemption 2;Action-Adventure;7;2018
Employee;Jim Halpert;1995;Jkl Street;2300;2900
Customer;Kevin Malone;theoffice
Song;Old Town Road;Lil Nas X;8;2019
Game;The Witcher 3: Wild Hunt;RPG;9;2015
Employee;Stanley Hudson;1998;Mno Street;2400;3000
Song;Uptown Funk;Mark Ronson;10;2014
Customer;Kelly Kapoor;qwerty
Game;Overwatch;First-Person Shooter;11;2016
```

# Scenario 2: Task: *Allowing Customers to also Rent Movies*

**Objective:**

Clone the `Song` class to create a `Movie` class in the DARTPlus system. Add a `feature-to-file mapping` for `RentItem` to `Movie.java` and implement the functionality of renting movies. Details written below.

**Tasks:**

1. Clone `Song` Class:

- Locate the Song `class` in the DARTPlus codebase.

- Clone the `Song` class to create a new class called `Movie` (in the same folder, i.e., `Items`).

- Within the Movie class, change what's necessary to allow movie rentals (e.g., replace `artist` by `director`).

2. **Map** the cloned file to `RentItem`.

3. Implement Movie Renting Functionality:

- Add a menu option for renting Item (movie) in the `customerLoggedIn` options within the `Controller` Class.

- Ensure that the implementation aligns with the existing rental functionality for `Games` and `Songs`. In the `feature model` view, right-click on the `RentItem` feature, click `Find Usages`, and update the existing feature implementation to accommodate movie rentals.

Expected Outcome:

- Feature-to-file mapping updated to include Movie.java for the RentItem feature.
- Implementation of movie rental functionality within the Movie class, allowing users to rent movies.