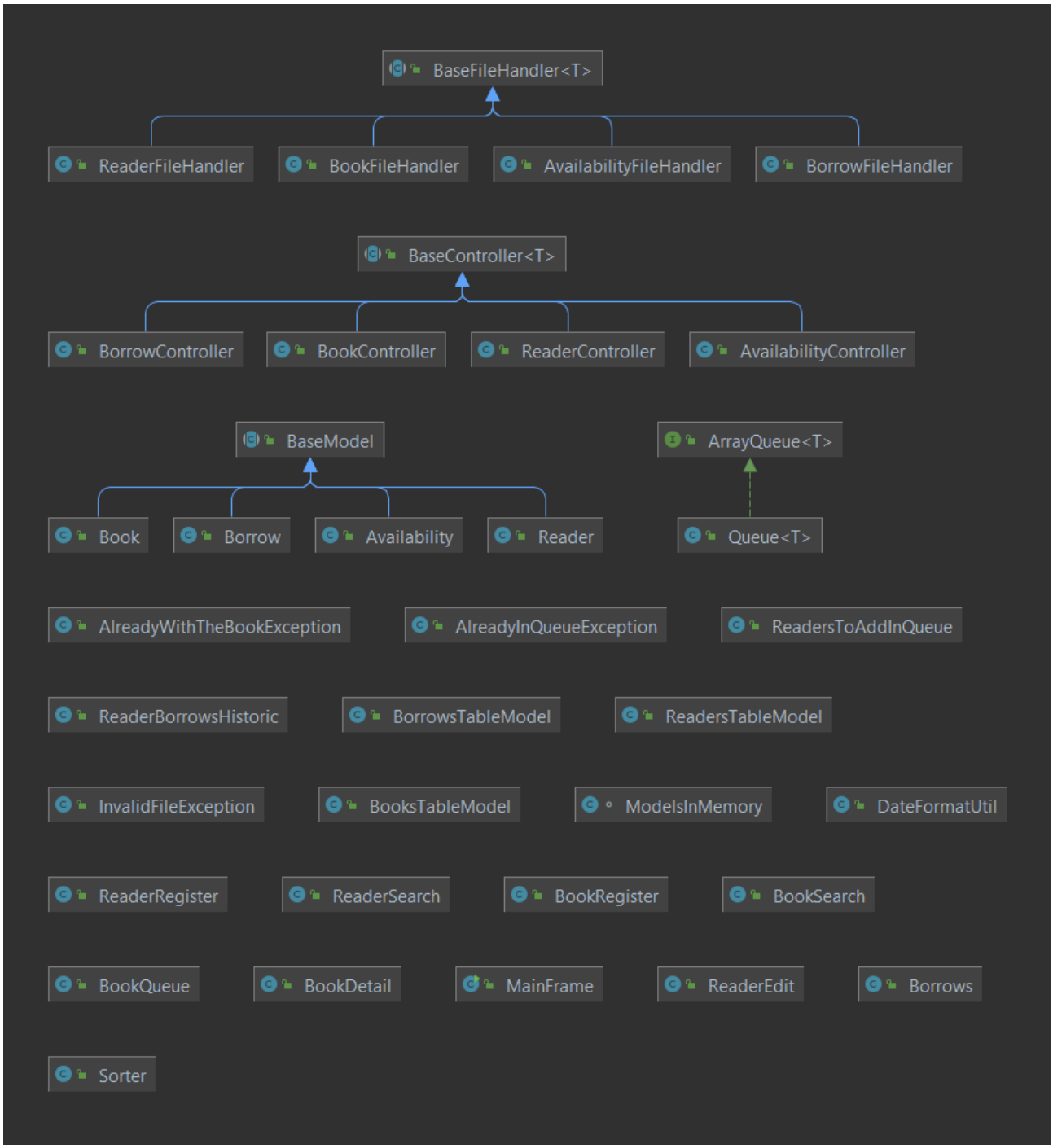
cct | College Dublin
Computing • IT • Business

Library System

Francisco Leite Meireles da Fonseca
2019300
BSc in Information Technology
3rd Year
Data Structure & Algorithms
Lecturer – Aldana Louzan

Table of Contents

Class Diagram:



Challenges and Design Decisions

Of the challenges I faced implementing the queue was the one that took me more time to do, after various attempts I decided that the best approach here would be put the queue not to the book but to an Availability Object that would hold the queue, and tell the system whether the book is available or not. It has the same Id as the book so it is easy to find the availability using the book's ID itself: the queue have 2 readers Id in the example below:

Book_Id	Amount_Available	queue
140a3d7e-a417-4c2b-96fe-32f9a2732693	0	9b9b0f50-b67b-4b74-82a2-fd658fde4940;2321748a-a850-466b-bac6-6a31968625de

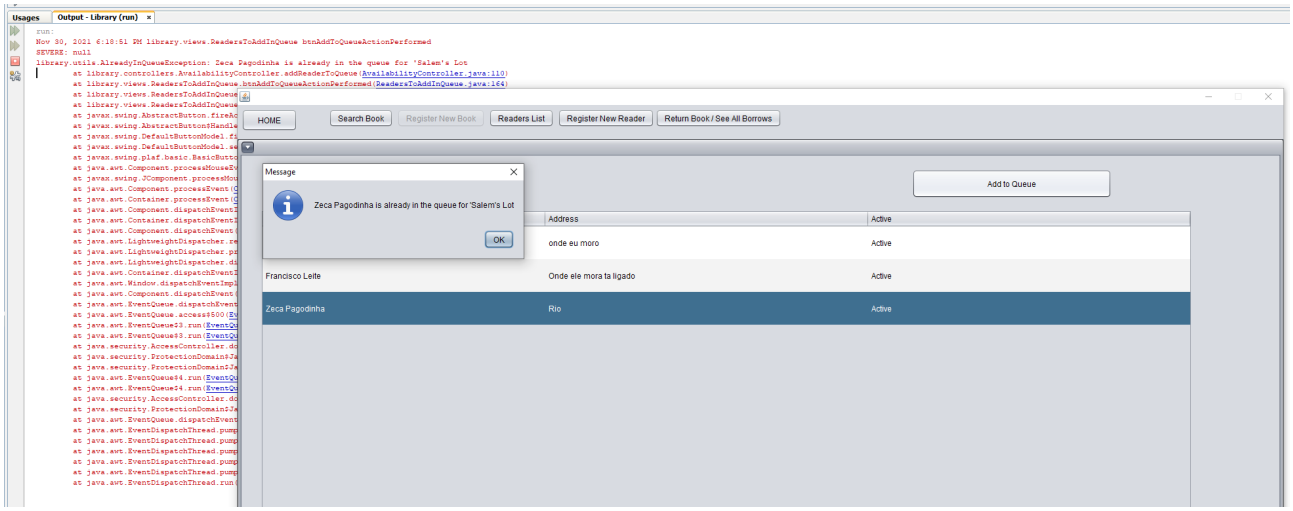
When I started the CA I didn't started with the base controller and the base file handler so after noticing I was repeating the code I ended up starting it all over again but now with proper logic, but this led me to other problem for example this method read file :

```
public T[] readFile() throws IOException, InvalidFileException{
    String fileName = getFileName();
    // if the file doesn't exist will create it based on the file name passed by the respective file handler
    if(!new File(fileName).exists()){
        writeFile(null);
    }
    BufferedReader reader = new BufferedReader(new InputStreamReader(new FileInputStream(fileName),"UTF-8"));
    String header = reader.readLine();
    validatedFile(header);
    // lemos os dados de determinado tipo T
     ArrayList<T> data = new ArrayList<>();
    String line = null;
     while ((line = reader.readLine()) != null) {
         //will get one element at a time because each File handle will split the line differently and
         //and it was not possible to convert an arrayList of T into an array(it needs the type) so the solution was to get one object
         //and create a "caster" to transform the array list into an array of the desired object
        T elem = loadData(line);
        data.add(elem);
     }
    // fechamos os streams
    reader.close();
    return castToArray(data);
}
```

This method in the Base File Handler is responsible for reading the data and returning an array of determined object, and it needs to be ArrayList because I wouldn't know how many lines there would be in the CSV File, but Java doesn't allow us to use the toArray() method with ArrayLists of Generics, and because I like to use table models to display the information for the user, working with array lists doesn't work great, so after a few hours of thinking I decided that the loadData() method implemented by each file handler would only return one object instead of the whole array list and then I created the abstract method castToArray() so each specific file handler would cast the ArrayList into an Array eg:

```
/**
 * cast an Array List and returns it as an Array
 * @param data
 * @return
 */
@Override
protected Borrow[] castToArray(ArrayList<Borrow> data) {
    return data.toArray(new Borrow[data.size()]);
}
```

I also experimented using my own exceptions so for example, when you try to add a read to a queue it is already in, it will throw an exception, it will display a message to the user and throw an exception, but the program will keep run smoothly despite the errors in the console:



I created a button and a view to register new books to the system, but I decided not to make the option available for this prototype, it is very easy to implement but taking it from my job as a data admin, even making some serious validations it is possible for the end user make mistakes with genres or ending up adding a book that already exists so it would have to be discussed with the Library in this case if they would like this to be a feature that only some system admin would be authorized to add a new book or anyone could, any way with the base controllers and file handlers it would be very easy to implement. So the button and the view and controllers are there but not activate for this prototype.

Java Doc is Available in the Dist Folder of the project