Data Structures & Algorithms FInal Project - Retake

Nnadozie Ebere.

USSD Movie Booking System For MarketSquare Cinemas Nigeria.

- **Abstract**

  MarketSquare recently opened its cinema doors to my city of birth Aba for city dwellers to come with their friends and family to watch movies. However, they were quick to implement an app-dependent movie booking system that customers don't use. MarketSquare is looking for a more simplified and inclusive way for customers to make movie reservations as an augmentation to their existing application. This added system should be efficient and quick in response.

- **Problem Analysis**

  Most of the citizens of the city of Aba are unable to effectively book movies on the MarketSquare's mobile app platform because the network in the city is very unreliable. Worse still,, data is costly, which is quite discouraging for the customers. MarketSquare needs a system that is efficient and requires no form of internet connection to work. This system is expected to collect the users' input just like the existing app and help them reserve their favourite movies and seats before getting to the cinema.

- **Problem Interpretation**
  - **Data Stored**:
    - Customer name - The system requires a simple process to collect and store customers name.
    - Movies - The system requires a simple process to collect and store customers name stores available movies in an array and allows the customer to choose any of choice.
    - The number of seats selected- The system requires a simple process to collect and store the number of seats each customer has selected.

      Afterwards, a unique identifier is required to identify customers when they get to the cinema uniquely.
  - Constraints: The time complexity for going the process of finding a movie of choice should be O(1)

- **Problem Solution**

  After a thorough analysis of the case, I recommend a USSD feature that allows everyone to book movies. This solution will allow customers to use a USSd code on their mobile devices to directly book a movie instantly without relying

on messy internet or expensive data. Users will simply enter their name, see available movies, choose one and reserve a seat. After that, they will get a unique ticket number that will let them make payments at the cinema.

- **Solution Implementation**
  For the implementation of this project, I decided to implement a movie class and use methods to achieve the desired outcome of the system.

  Thereafter, I created an array to stories the movies that are available for viewing at the moment of booking

```
static String[] movielist = {"1. The Black Panther 2", "2. Money Heist S5", "3. Black Is King", "4. Queen Sono s2", "5. Blood & Water S2", "6. Merry Men 3"};
```

  Next, I created a function to get and store the name of the user

```
public static void getCustomer(){
  System.out.println("Welcome to Century Cinemas! Please input your name");
  name=scan.nextLine();
  System.out.println("Welcome " +name + "! Are ready to have a nice time?");
  selectMovie();
}
```

  In the same style, I created a function to get and store the movie choice of the customer

```
public static void selectMovie(){
  System.out.println("Kindly select a movie number");
  for (int i=0; i<movielist.length; i++) {
      System.out.println(movielist[i]);
  }
      number = scan.nextInt();
      System.out.println(name + ", you have selected " + movielist[number-1].substring(3,
              movielist[number-1].length()));
      getSeat();
}
```

  Finally, I created a method to collect and store the number of seats taken by the customer, thereafter, I created an algorithm to generate a unique ticket number for the customer which will facilitate payments at the cinema.

Conclusion

Problem Solving:

MarketSquare like other cinema houses is aiming for the most innovative ways of serving their customers. But by putting the customers' needs first, I created a swift and simple system that saves them the headache of a unreliable internet and expensive data prices, whilst guaranteeing them a memorable time at the cinema

Runtime: The complexity of fetching and adding to the array used here (no matter the size is O(1). Therefore, it is guaranteed that this system is fast and reliable.

User-Friendliness: This system is built to serve nearly every type of users and isn't device-specific. The USSD interface is quite easy to understand.

Code Repository: https://github.com/nnadozie0/RetakeSummative.git