The whole program is consist of four classes named Booking, Continent, Country, and Destination. Booking Class has a main method and a switch case with four cases, each of which has a loop that allows the user to perform actions until they choose to go back to the main menu. In the first case, the user can add information about a country, including its name, spoken language, and whether it is open to tourists. The user is also prompted to choose a continent to which the country should be added using the method addCountriesToContinent() which is created in the Continent class with an argument to choose a continent along with the Country class properties. In the second case, the user can add information about a destination, including its name, capacity, latitude, and longitude. The user is also prompted to choose a country and a continent in which the destination is located, using the method displayCountriesOfSelectedContinentandAddandAddDestination() by taking all fields of the continent and a variable to select continent as arguments. in this third case, user is prompted to choose a number as an option to see the statistics of the entire system. And the last case is for closing the program. In the Continent class, I created a final ArrayList of ArrayList to store all the default continents by using a function called continents() which returns an ArrayList of Country type ArrayList. In order to add countries in the continents I created a method named addCountriesToContinents() taking arguments name of the country, the language of the country, flag for showing is it open to tourists or not, and also an input from the user to choose a continent that user wants to add country in. To add a Destination in a user-selected country I created an ArrayList of Destination in the Country class which stores all the destination objects created by a user in runtime for a specified country using a method addDestination() taking arguments name of the destination, capacity, latitude, and longitude. But To select a country to add a destination to it first I created a method called printCountries() To print all the countries in a specific continent. Then again I created a function AddDestinationToDesiredCountry() which adds destinations to the desired country including arguments name of the destination, capacity, latitude, and longitude along with the index to select a continent and also an index to select country as an argument. In the last I created a function named displayCountriesOfSelectedContinentandAddDestination() to check if a continent selected by the user is empty or not, if not then first call the printCountries() method to print all the existing countries in a user selected continent using seven switch cases for seven continents and calling the addDestinationToDesired() country method to add destinations according to the user selected continent and country using their index, taking arguments inputChoice,userInputForContinent, name of the destination, capacity, longitude, and latitude. To view the statistics, first I created three methods in the country class as per the coursework description which gives the average capacity of a country, methods which print all the destination which has a capacity larger than a given number, and a method which returns a destination object which has the highest capacity for a given number. Again, as per the coursework description, I created these three functions for the entire system not specific to a country including a method that gives a list of countries which has a capacity larger than a given number.

Overall, the program appears to be well-structured and easy to understand, with clear and concise variable names and organized code blocks. The program also handles input mismatch exceptions, which is important for ensuring the stability and reliability of the program. It also checks for the null pointer exception and index out-of-bound exception. It helps to prevent the program from crashing due to invalid input. Talking about improvement in terms of object-oriented programming, the program is not using all the components of OOP like composition, interface, abstraction, etc. It could be improved using all the pillars of OOP. One improvement could be adding more validation to input to get rid-off occasional errors and inconsistencies in the data. To extend the Program we can allow the user to search for specific countries or destinations by it is named, rather than displaying all the information of the continent. One can implement the ability to delete and edit countries and destinations from the continent information. Also, One possibility could be adding more information about destination and country , like any special attraction in a the destination, the currency of a country etc. it would help users to get more in depth information about a place.