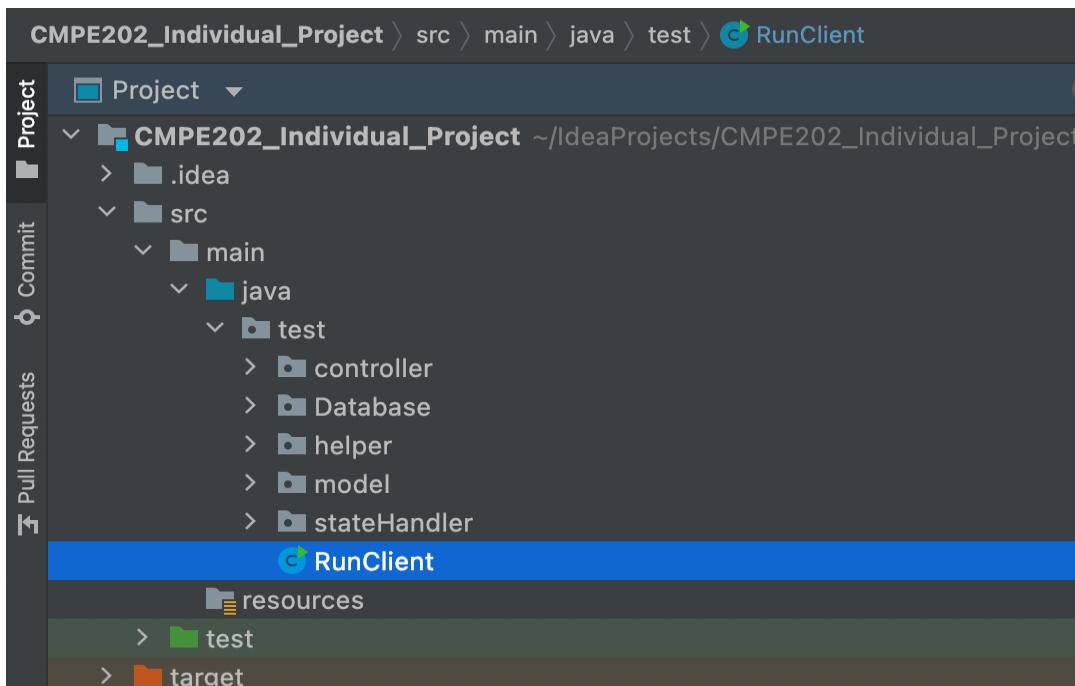


INSTRUCTIONS TO RUN THE APPLICATION:

Needs Java 18 installed in the system that is running the program.

INSTRUCTIONS TO RUN THE MAVEN APPLICATION:

- File Structure:

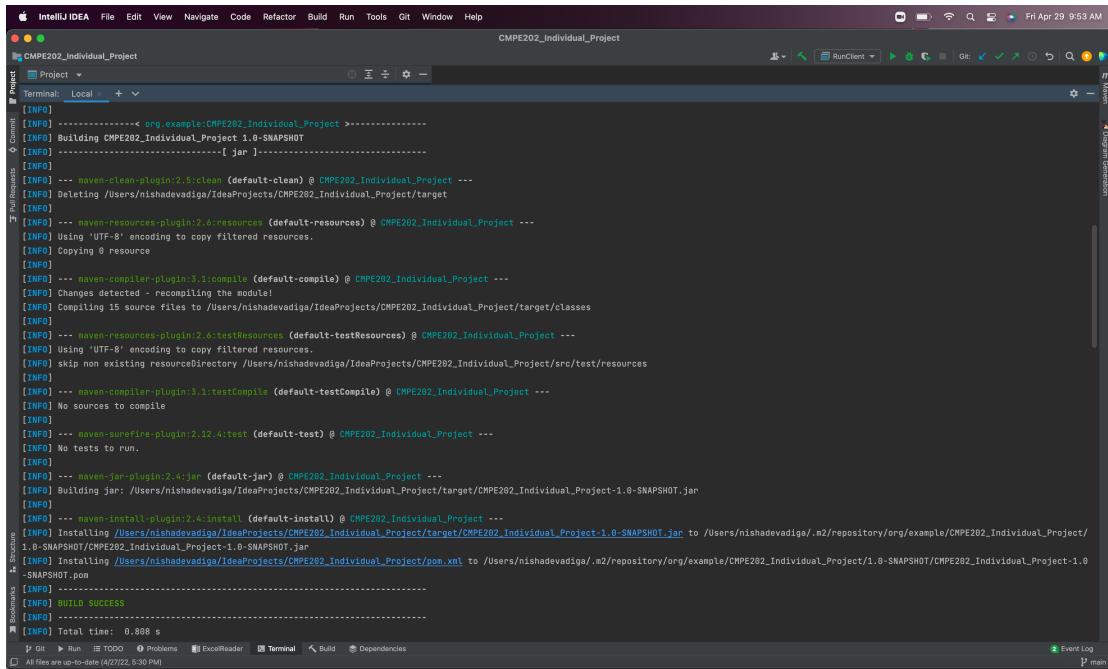


- Open Command Prompt in the main directory - **Go to CMPE202 Individual Project.**
Run below commands in the sequence to install the maven project –

```
mvn compile
```

```
mvn clean install
```

```
(base) nishadevadiga@Nishas-MacBook-Pro CMPE202_Individual_Project % clear
(base) nishadevadiga@Nishas-MacBook-Pro CMPE202_Individual_Project % mvn -version
Apache Maven 3.8.3 ((599493414f8046de2324203b758ddcf9b5e388aa0)
Maven home: /opt/homebrew/Cellar/maven/3.8.3/libexec
Java version: 18.0.1, vendor: Homebrew, runtime: /opt/homebrew/Cellar/openjdk/18.0.1/libexec/openjdk.jdk/Contents/Home
Default locale: en_US, platform encoding: UTF-8
OS name: "mac os x", version: "12.3.1", arch: "aarch64", family: "mac"
(base) nishadevadiga@Nishas-MacBook-Pro CMPE202_Individual_Project % mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] ----- org.example:CMPE202_Individual_Project >-----
[INFO] Building CMPE202_Individual_Project 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ CMPE202_Individual_Project ---
[INFO] Using UTF-8' encoding to copy filtered resources.
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ CMPE202_Individual_Project ---
[INFO] Nothing to compile - all classes are up to date
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.287 s
[INFO] Finished at: 2022-04-29T09:41:38-07:00
[INFO] -----
(base) nishadevadiga@Nishas-MacBook-Pro CMPE202_Individual_Project % mvn clean install
[INFO] Scanning for projects...
[INFO]
[INFO] ----- org.example:CMPE202_Individual_Project >-----
```



- Run the following command in the command prompt:

```
mvn exec:java -Dexec.mainClass=test.RunClient -Dexec.args=" <Path to Flight details to populate DB>"
```

For Example:

- ### 1. For Input File 1: -

```
mvn exec:java -Dexec.mainClass=test.RunClient -Dexec.args="InputFile.csv flights.csv"
```

- ## 2. For Input File 2: -

```
mvn exec:java -Dexec.mainClass=test.RunClient -Dexec.args="InputFile1.csv flights.csv"
```

3. For Input File 3: -

```
mvn exec:java -Dexec.mainClass=test.RunClient -Dexec.args="InputFile2.csv flights.csv"
```

4. For Input File 4: -

```
mvn exec:java -Dexec.mainClass=test.RunClient -Dexec.args="InputFile3.csv flights.csv"
```

5. For Input File 5: -

```
mvn exec:java -Dexec.mainClass=test.RunClient -Dexec.args="InputFile4.csv flights.csv"
```

- Output files will be created in the directory as the files provided as arguments to run the command.

PRIMARY PROBLEM:

We will build a flight booking application which maintains an internal, static database (inventory of flight details) created using HashMap. The data will contain the following tables and fields:

Table 1: Flights

- **Category** (Economy, Premium Economy, Business)
- **Flight number**
- The available **Seats** of each category
- **Price** of each seat
- **Arrival City**
- **Departure City**

Input CSV file contains booking details including booking name, flight number, seat category, number of seats, and the payment card number. Here we will need to perform few validations on seats, flight, and card details.

The primary problem is to decouple a request's sender and receiver when handling the validation requests. Multiple objects, determined at runtime, are candidates to handle a request. Here specifying handlers explicitly in the code could lead to non-reusability. To overcome this problem, we opted for Chain of Responsibility design pattern that solves the problem by designing flexible and reusable object-oriented software. That is, objects are easier to implement, change, test, and reuse.

SECONDARY PROBLEMS:

The secondary problem is related to utilization of heap memory by allowing to create the copies of objects for the database. In this case, single database object is good enough to perform the operations in the project for various operations. To solve this problem, we use Singleton Design Pattern that restricts the instantiation of class to only one object.

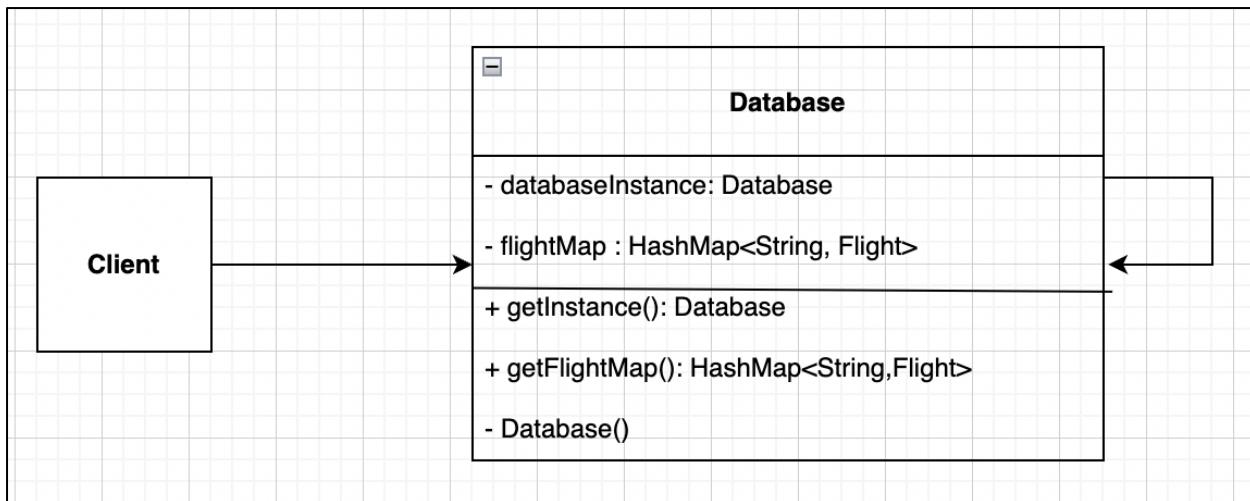
The next secondary problem is tight coupling and poor level of abstraction where complexities of code and internal implementation is visible to user. To avoid this, we need to perform encapsulation by implementing Factory Design pattern.

DESIGN PATTERN(S):

- Singleton
- Factory
- Chains of Responsibility

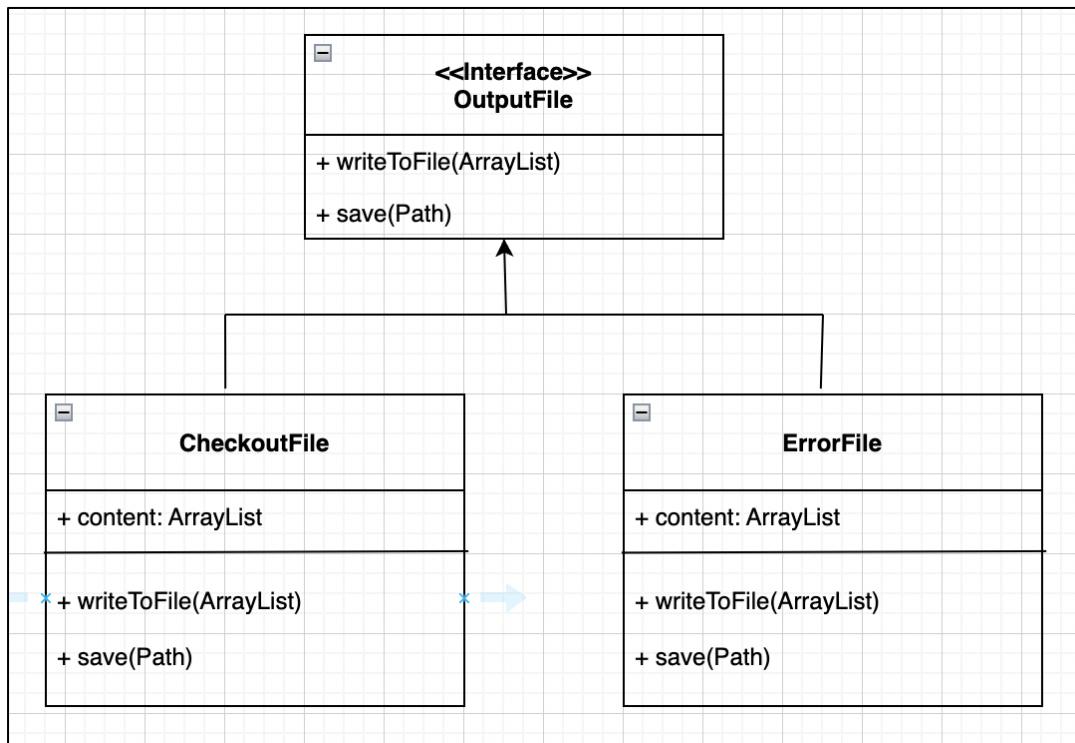
SINGLETON:

- It is creational design pattern that lets you ensure that a class has only one instance, while providing the global access point to this instance.
- This pattern is used to build the database for the application. Only the Database class is used as part of this pattern.
- Whenever an instance of the database is needed, the getInstance() method is called so that only one instance of the object can be reused.



FACTORY:

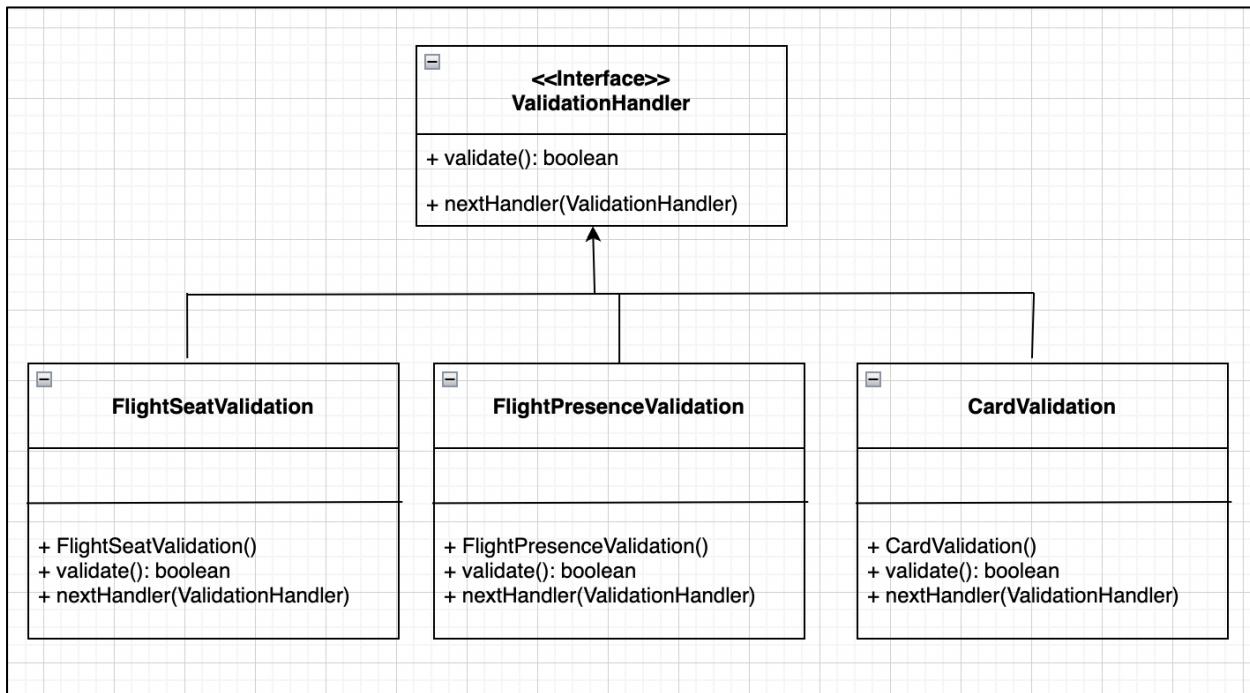
- It is a creational design pattern that provides an interface for creating objects in a superclass but allows subclasses to alter the type of objects that will be created.
- We have used this pattern to output when an order is processed. The following files are as part of this pattern implementation:
 - Interface OutputFile
 - Class CheckoutFile
 - Class ErrorFile
- When the respective concrete handler figures out the output should be a checkout file or error file, we will create an object of OutputFile. The FileReader class is used to read and create files in the application.



CHAINS OF RESPONSIBILITY:

- It is a behavioral design pattern that lets you pass requests along a chain of handlers. Upon receiving a request, each handler decided either to process the request or to pass it to the next handler in the chain.
- We have used this design pattern to determine if the order can be processed correctly or not. The following files are used in this pattern:
 - Interface ValidationHandler
 - Class FlightPresenceValidation
 - Class FlightSeatValidation.
 - Class CardValidation

- Before we verify the order can be processed or not, ValidationHandler object is created and will be passed on from one Handler to another using the same.



CONSEQUENCES OF USING THESE PATTERN(S):

Singleton

The Singleton pattern solves below two problems at the same time but by violating the Single Responsibility Principle:

1. Ensures that a class has just a single instance – Database Class Instance is instantiated only once to avoid creating a new one.
2. Provides a global access point to that instance – Just like a global variable, the Singleton pattern lets you access Database object from anywhere in the program. However, it also protects that instance from being overwritten by other code.

Chain of Responsibility

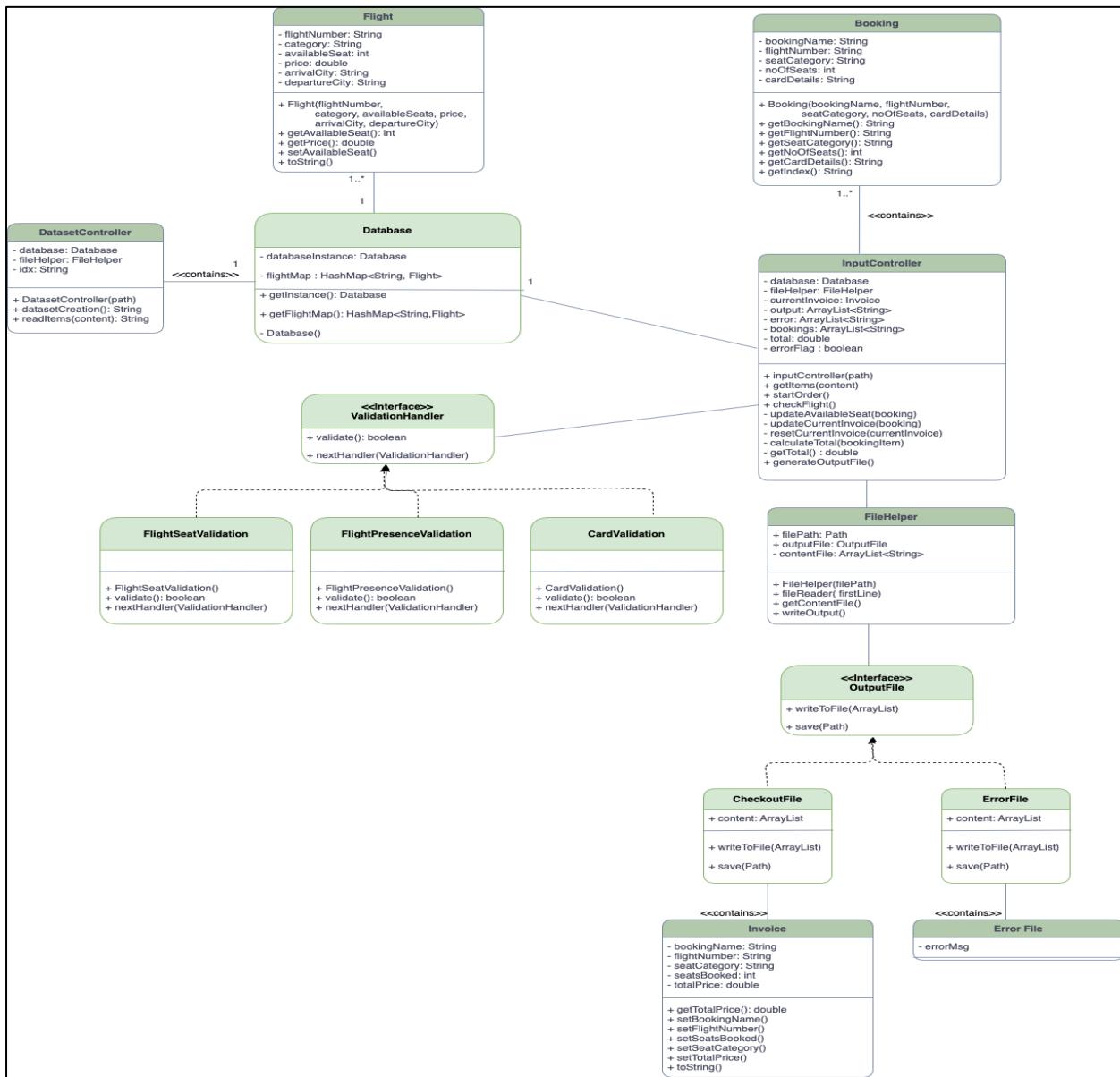
Using Chain of Responsibility, two important principles of SOLID can be achieved.

1. SINGLE RESPONSIBILITY PRINCIPLE - You can decouple classes that invoke operations from classes that perform operations.
2. OPEN/CLOSED PRINCIPLE - You can introduce new handlers into the app without breaking the existing client code.

Factory

1. Provides hooks for subclasses. Creating objects inside a class with a factory method is always more flexible than creating an object directly.
2. Factory Method gives subclasses a hook for providing an extended version of an object.

CLASS DIAGRAM:



JUNIT TEST CASES SHOULD BE WRITTEN FOR ALL THE DESIGN PATTERN CLASSES/METHODS. SCREENSHOTS OF TEST EXECUTION AND RESULT SHOULD BE INCLUDED:

Flight DB:

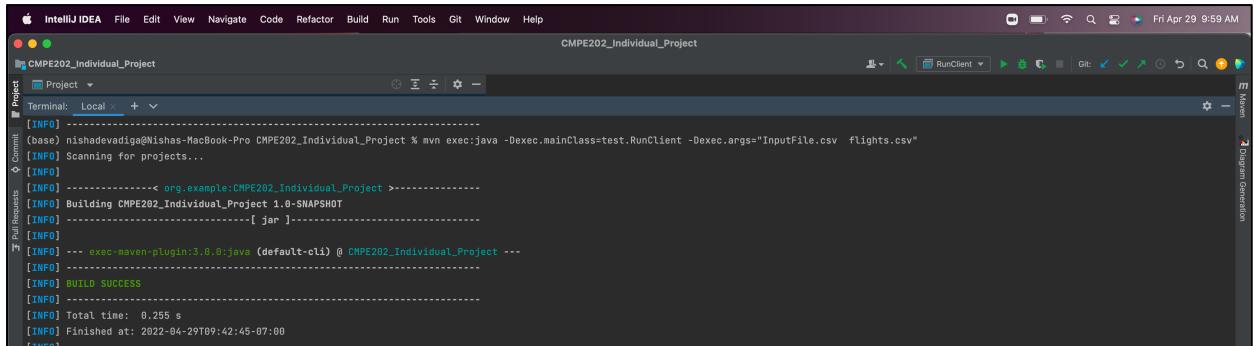
A	B	C	D	E	F
Category(Economy Premium Economy Business)	FlightNumber	AvailableSeats	Price	Arrival	Departure
Economy	SJ456	5	250	Seattle	San Jose
Premium Economy	BY110	5	500	San Francisco	New York
Business	BY110	5	2000	San Francisco	New York
Economy	CA453	5	300	Seattle	San Jose
Business	CA453	5	1500	Seattle	San Jose

Test Case 1 – Non- Existent Flight in the input file.

- Input File:

A	B	C	D	E
BookingName	flightNumber	seatCategory	numberOfSeats	paymentCardNumber
Sam	SJ456	Economy	4	5410000000000000000
Richard	BY110	Premium Economy	2	3410000000000000000
Anna	SJ456	Economy	1	4120000000000000000
John	CA111	Economy	1	5410000000000000000
Sierra	CA453	Business	1	5410000000000000000

- Command:

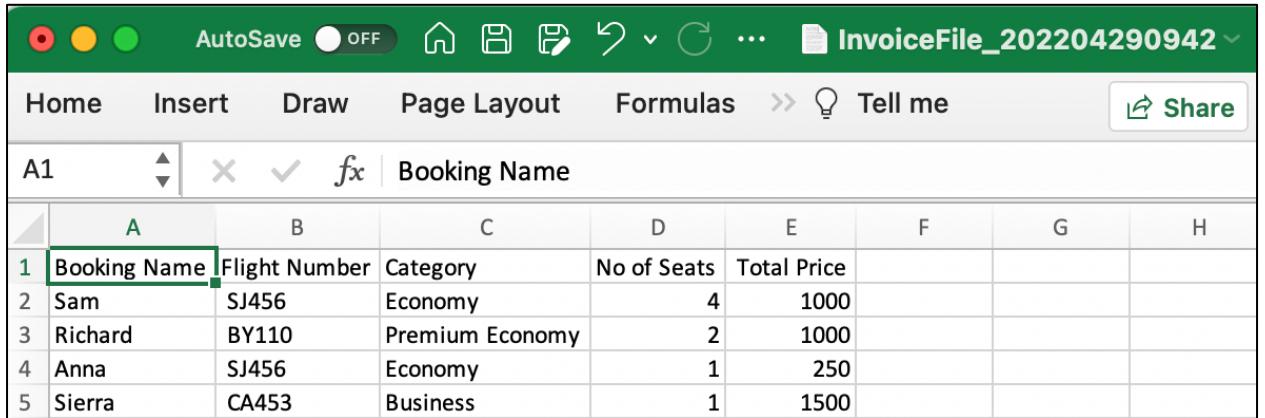


```

IntelliJ IDEA 2021.3.1
File Edit View Navigate Code Refactor Build Run Tools Git Window Help
CMPE202_Individual_Project
Project Terminal Local + -
[INFO] ...
(base) nishadevadiga@Nisha-MacBook-Pro CMPE202_Individual_Project % mvn exec:java -Dexec.mainClass=test.RunClient -Dexec.args="InputFile.csv flights.csv"
[INFO] Scanning for projects...
[INFO] ...
< org.example:CMPE202_Individual_Project >
[INFO] Building CMPE202_Individual_Project 1.0-SNAPSHOT
[INFO] ...
[INFO] ... exec-maven-plugin:3.6.0:java (default-cli) @ CMPE202_Individual_Project ...
[INFO] ...
[INFO] BUILD SUCCESS
[INFO] ...
[INFO] Total time: 0.295 s
[INFO] Finished at: 2022-04-29T09:42:45-07:00
[INFO]

```

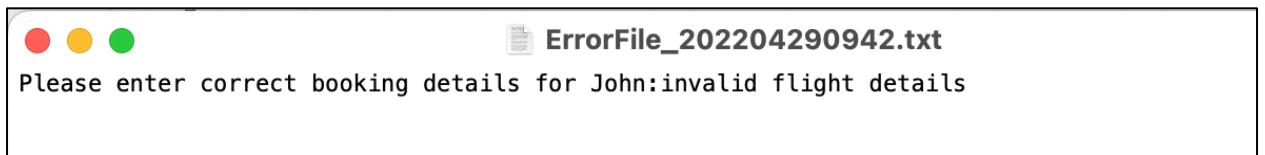
- Output File:



A screenshot of Microsoft Excel showing a table of flight bookings. The table has columns for Booking Name, Flight Number, Category, No of Seats, and Total Price. The data is as follows:

	A	B	C	D	E	F	G	H
1	Booking Name	Flight Number	Category	No of Seats	Total Price			
2	Sam	SJ456	Economy	4	1000			
3	Richard	BY110	Premium Economy	2	1000			
4	Anna	SJ456	Economy	1	250			
5	Sierra	CA453	Business	1	1500			

- Error File:



Test case 2 – Seat Unavailable for Flight SJ456 - Economy

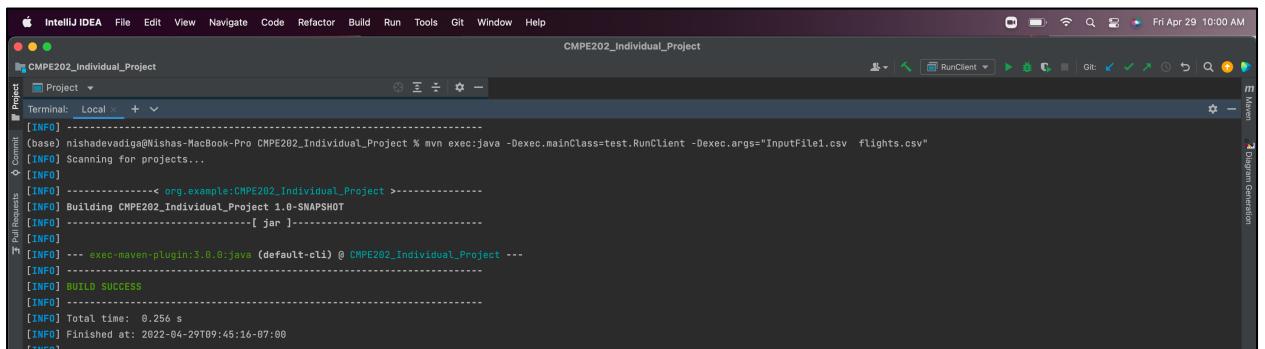
- Input File:



A screenshot of Microsoft Excel showing a table of flight bookings. The table has columns for BookingName, flightNumber, seatCategory, numberofSeats, and paymentCardNumber. The data is as follows:

	A	B	C	D	E
	BookingName	flightNumber	seatCategory	numberofSeats	paymentCardNumber
	Sam	SJ456	Economy	4	5410000000000000000
	Richard	BY110	Premium Economy	2	3410000000000000000
	Anna	SJ456	Economy	3	4120000000000000000
	John	CA453	Economy	1	5410000000000000000
	Sierra	CA453	Business	1	5410000000000000000

- Command:



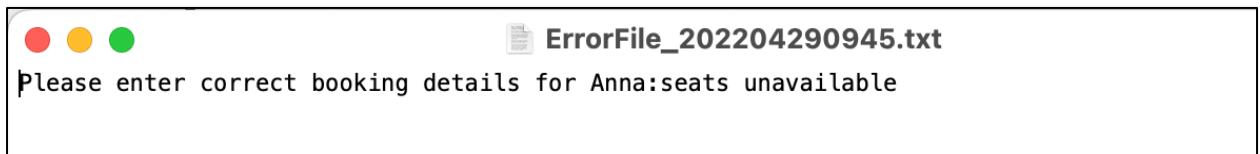
A screenshot of the IntelliJ IDEA terminal window. The output shows the execution of a Maven command to run a Java application named RunClient. The command was run with arguments "InputFile1.csv flights.csv". The output indicates a successful build and execution.

```
IntelliJ IDEA 2021.2.3
File Edit View Navigate Code Refactor Build Run Tools Git Window Help
CMPE202_Individual_Project
Project Terminal Local Fri Apr 29 10:00:00
[INFO] ...
[INFO] (base) nishadevadiga@nishas-MacBook-Pro CMPE202_Individual_Project % mvn exec:java -Dexec.mainClass=test.RunClient -Dexec.args="InputFile1.csv flights.csv"
[INFO] Commit: [REDACTED]
[INFO] Scanning for projects...
[INFO] [INFO] -----> org.example:CMPE202_Individual_Project <-----[REDACTED]
[INFO] Building CMPE202_Individual_Project 1.0-SNAPSHOT
[INFO] [INFO] -----[ jar ]-----[REDACTED]
[INFO] [INFO] --- exec-maven-plugin:3.0.0:java (default-cli) @ CMPE202_Individual_Project ---
[INFO] [INFO] BUILD SUCCESS
[INFO] [INFO] -----
[INFO] Total time: 0.256 s
[INFO] Finished at: 2022-04-29T09:45:16-07:00
[INFO]
```

- Output File:

	A	B	C	D	E	F	G	H	I
1	Booking Name	Flight Numbr	Category	No of Seats	Total Price				
2	Sam	SJ456	Economy	4	1000				
3	Richard	BY110	Premium Econ	2	1000				
4	John	CA453	Economy	1	300				
5	Sierra	CA453	Business	1	1500				
6									

- Error File:

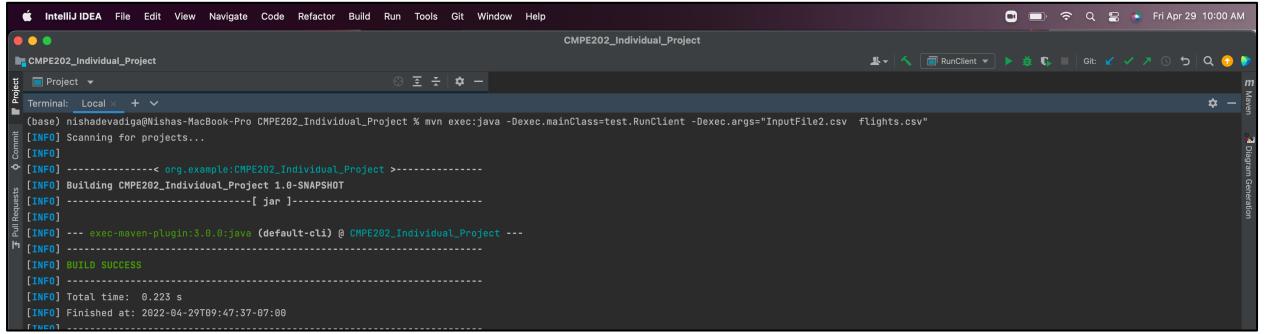


Test Case 3 – Flight No present but Category not Present in Flight Database. For example - CA453 & Premium Economy.

- Input File:

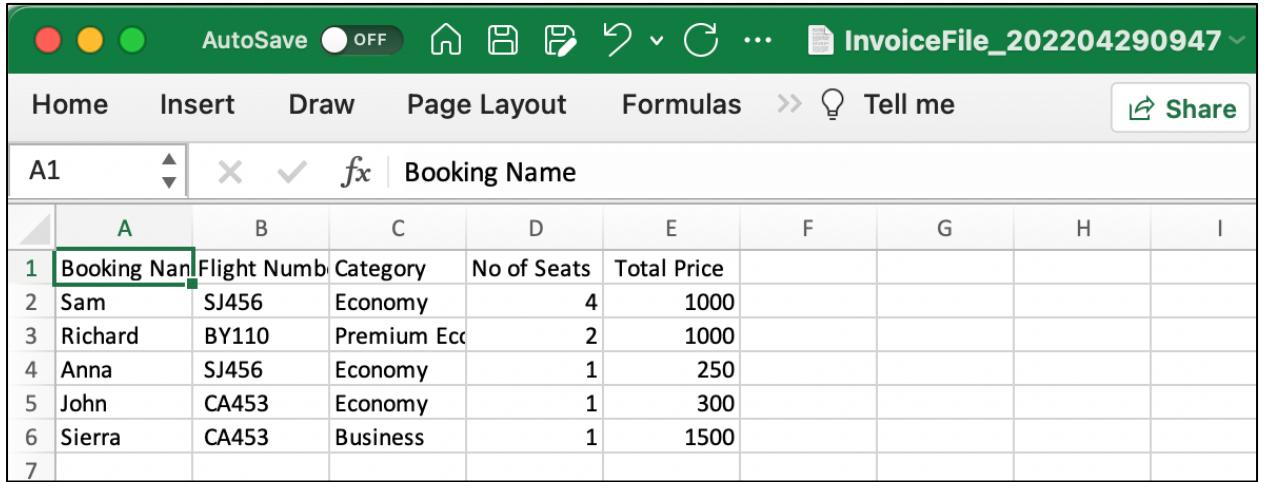
A	B	C	D	E
BookingName	flightNumber	seatCategory	numberOfSeats	paymentCardNumber
Sam	SJ456	Economy	4	5410000000000000
Richard	BY110	Premium Economy	2	3410000000000000
Anna	SJ456	Economy	1	4120000000000000
John	CA453	Economy	1	5410000000000000
Sierra	CA453	Business	1	5410000000000000
Nisha	CA453	Premium Economy	1	5410000000000000

- Command:



```
(base) nishadevadiga@Nisha-OptiPlex-5090:~/Desktop/CMPE202_Individual_Project % mvn exec:java -Dexec.mainClass=test.RunClient -Dexec.args="InputFile2.csv flights.csv"
[INFO] Scanning for projects...
[INFO] ...
[INFO] -----< org.example:CMPE202_Individual_Project >-----
[INFO] Building CMPE202_Individual_Project 1.0-SNAPSHOT
[INFO] ...
[INFO] --- exec-maven-plugin:3.0.0:java (default-cli) @ CMPE202_Individual_Project ---
[INFO] ...
[INFO] BUILD SUCCESS
[INFO] ...
[INFO] Total time: 0.223 s
[INFO] Finished at: 2022-04-29T09:47:37+07:00
[INFO]
```

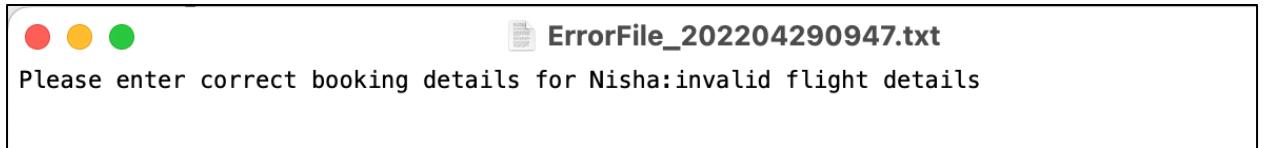
- Output File:



InvoiceFile_202204290947

	A	B	C	D	E	F	G	H	I
1	Booking Name	Flight Num	Category	No of Seats	Total Price				
2	Sam	SJ456	Economy	4	1000				
3	Richard	BY110	Premium Econ	2	1000				
4	Anna	SJ456	Economy	1	250				
5	John	CA453	Economy	1	300				
6	Sierra	CA453	Business	1	1500				
7									

- Error File:



ErrorFile_202204290947.txt

Please enter correct booking details for Nisha:invalid flight details

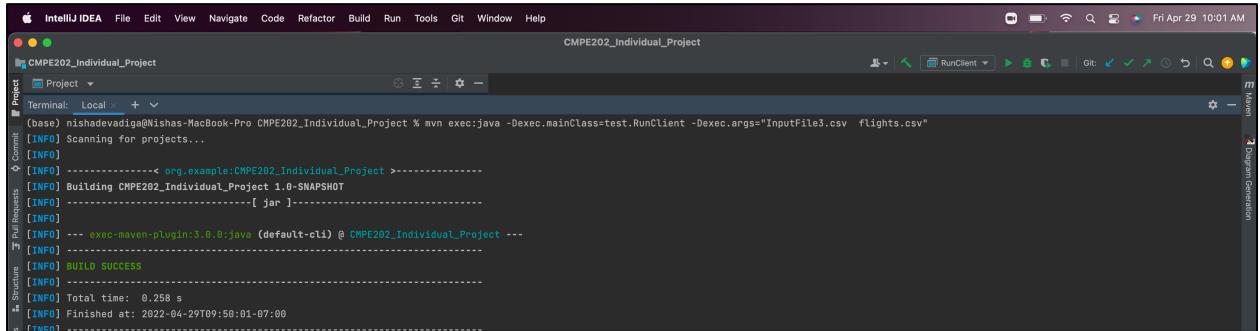
Test case 4: All Validations Passed

- Input File:



A	B	C	D	E
BookingName	flightNumber	seatCategory	numberOfSeats	paymentCardNumber
Sam	SJ456	Economy	4	5410000000000000
Richard	BY110	Premium Economy	2	3410000000000000
Anna	SJ456	Economy	1	4120000000000000
John	CA453	Economy	1	5410000000000000
Sierra	CA453	Business	1	5410000000000000

- Command:

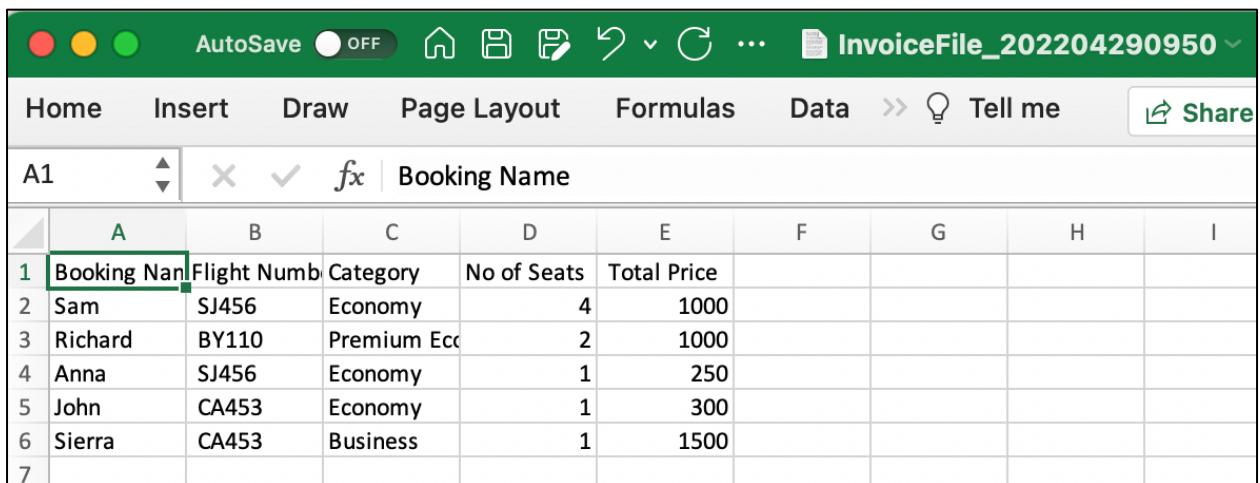


```

IntelliJ IDEA 2021.3.1
File Edit View Navigate Code Refactor Build Run Tools Git Window Help
CMPE202_Individual_Project
Project Local + -
Terminal Local + -
(base) nishadevadiga@nisha-MacBook-Pro CMPE202_Individual_Project % mvn exec:java -Dexec.mainClass=test.RunClient -Dexec.args="InputFile3.csv flights.csv"
[INFO] Scanning for projects...
[INFO] ------------------------------------------------------------------------
[INFO] Building CMPE202_Individual_Project 1.0-SNAPSHOT
[INFO] [jar]
[INFO] ------------------------------------------------------------------------
[INFO] --- exec-maven-plugin:3.0.0:java (default-cli) @ CMPE202_Individual_Project ---
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 0.258 s
[INFO] Finished at: 2022-04-29T09:50:01+07:00
[INFO] 

```

- Output File:



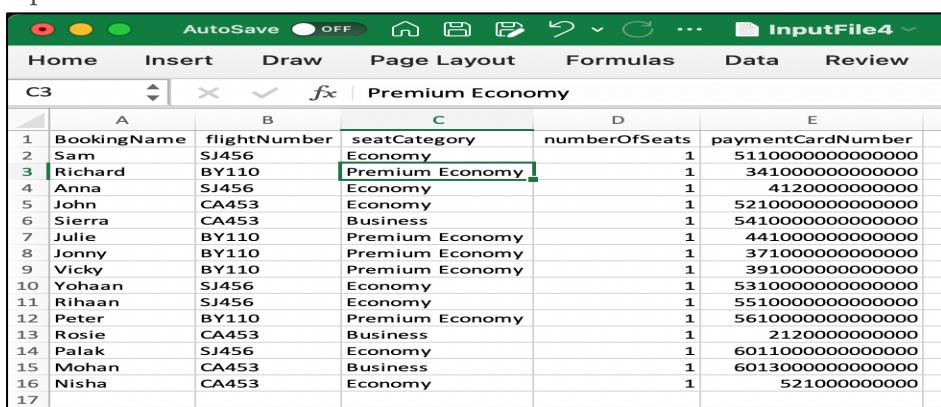
The screenshot shows a Microsoft Excel spreadsheet titled "InvoiceFile_202204290950". The data is organized into columns labeled A through I. Column A is "Booking Name", column B is "Flight Numbr", column C is "Category", column D is "No of Seats", and column E is "Total Price". The data rows are as follows:

	Booking Name	Flight Numbr	Category	No of Seats	Total Price
1	Sam	SJ456	Economy	4	1000
2	Richard	BY110	Premium Econ	2	1000
3	Anna	SJ456	Economy	1	250
4	John	CA453	Economy	1	300
5	Sierra	CA453	Business	1	1500
6					
7					

- Error File:
No error File Generated.

Test case 5: Invalid Card details

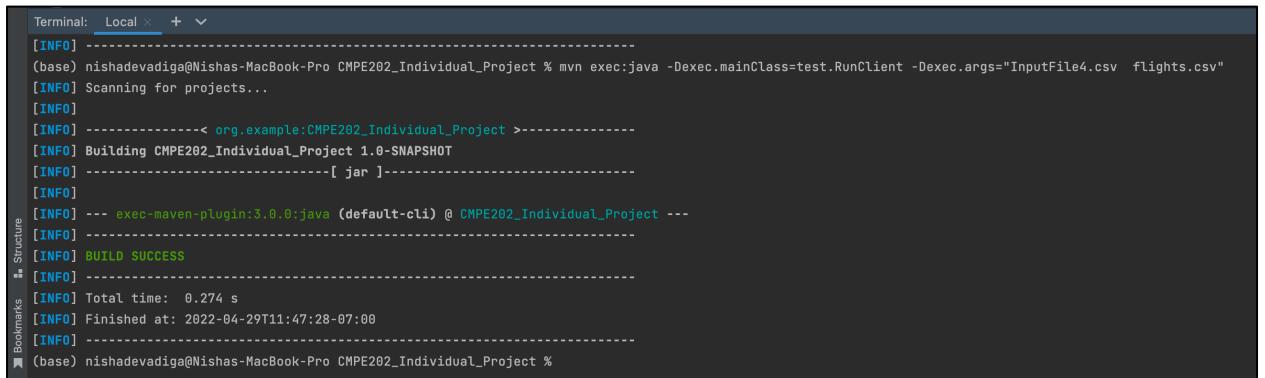
- Input File:



The screenshot shows a Microsoft Excel spreadsheet titled "InputFile4". The data is organized into columns labeled A through E. Column A is "BookingName", column B is "flightNumber", column C is "seatCategory", column D is "numberOfSeats", and column E is "paymentCardNumber". The data rows are as follows:

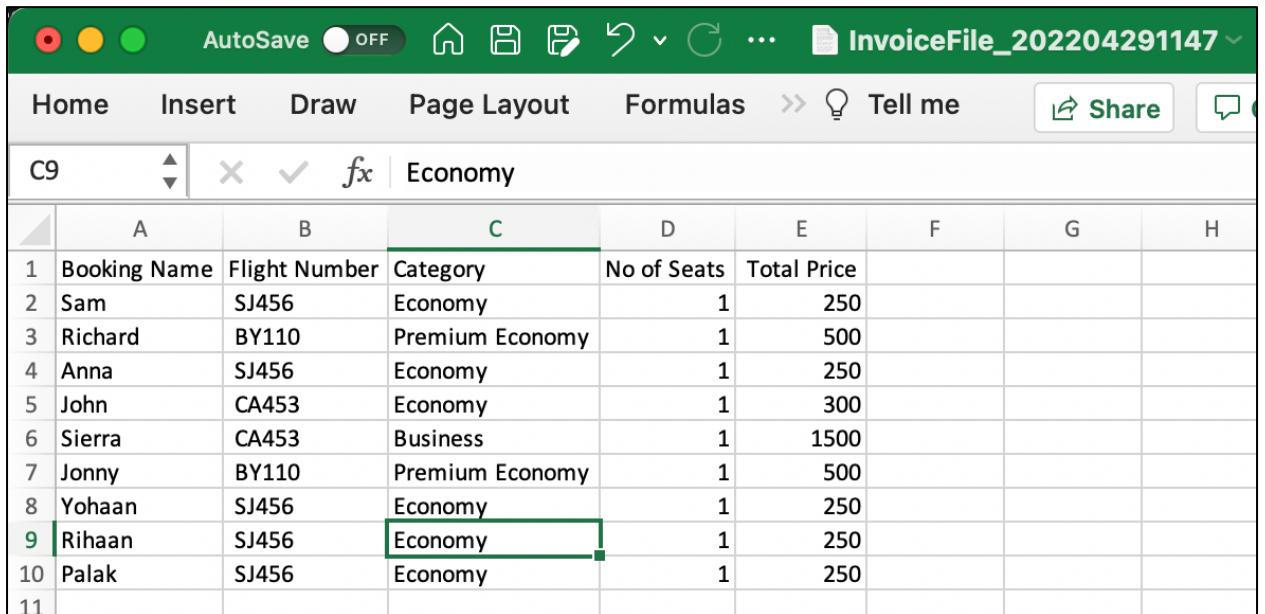
	BookingName	flightNumber	seatCategory	numberOfSeats	paymentCardNumber
1	Sam	SJ456	Economy	1	5110000000000000
2	Richard	BY110	Premium Economy	1	3410000000000000
3	Anna	SJ456	Economy	1	4120000000000000
4	John	CA453	Economy	1	5210000000000000
5	Sierra	CA453	Business	1	5410000000000000
6	Julie	BY110	Premium Economy	1	4410000000000000
7	Jonny	BY110	Premium Economy	1	3710000000000000
8	Vicky	BY110	Premium Economy	1	3910000000000000
9	Yohaan	SJ456	Economy	1	5310000000000000
10	Rihaan	SJ456	Economy	1	5510000000000000
11	Peter	BY110	Premium Economy	1	5610000000000000
12	Rosie	CA453	Business	1	2120000000000000
13	Palak	SJ456	Economy	1	6011000000000000
14	Mohan	CA453	Business	1	6013000000000000
15	Nisha	CA453	Economy	1	5210000000000000
16					
17					

- Command:



```
Terminal: Local + 
[INFO] -----
(base) nishadevadiga@Nishas-MacBook-Pro CMPE202_Individual_Project % mvn exec:java -Dexec.mainClass=test.RunClient -Dexec.args="InputFile4.csv flights.csv"
[INFO] Scanning for projects...
[INFO]
[INFO] -----< org.example:CMPE202_Individual_Project >-----
[INFO] Building CMPE202_Individual_Project 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- exec-maven-plugin:3.0.0:java (default-cli) @ CMPE202_Individual_Project ---
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 0.274 s
[INFO] Finished at: 2022-04-29T11:47:28-07:00
[INFO]
[INFO] -----
(base) nishadevadiga@Nishas-MacBook-Pro CMPE202_Individual_Project %
```

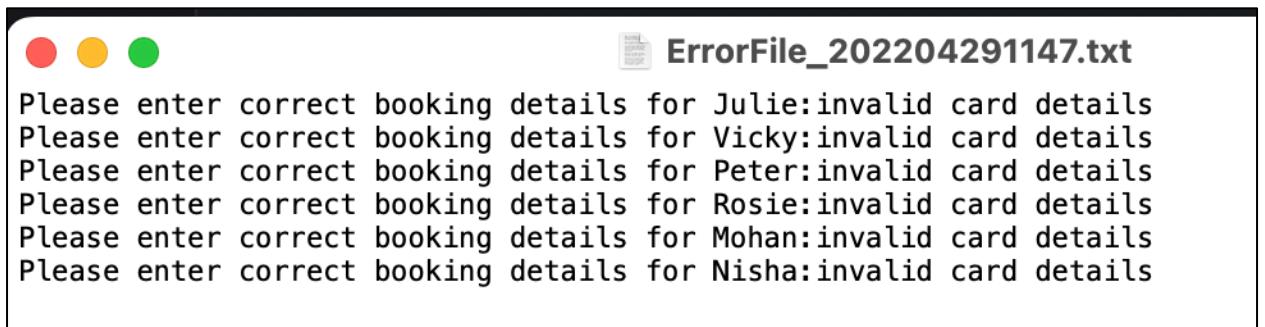
- Output File:



The screenshot shows a Microsoft Excel spreadsheet titled "InvoiceFile_202204291147". The table has columns: Booking Name, Flight Number, Category, No of Seats, and Total Price. The data is as follows:

	A	B	C	D	E	F	G	H
1	Booking Name	Flight Number	Category	No of Seats	Total Price			
2	Sam	SJ456	Economy	1	250			
3	Richard	BY110	Premium Economy	1	500			
4	Anna	SJ456	Economy	1	250			
5	John	CA453	Economy	1	300			
6	Sierra	CA453	Business	1	1500			
7	Jonny	BY110	Premium Economy	1	500			
8	Yohaan	SJ456	Economy	1	250			
9	Rihaan	SJ456	Economy	1	250			
10	Palak	SJ456	Economy	1	250			
11								

- Error File:



The screenshot shows a text file titled "ErrorFile_202204291147.txt". The content of the file is as follows:

```
Please enter correct booking details for Julie:invalid card details
Please enter correct booking details for Vicky:invalid card details
Please enter correct booking details for Peter:invalid card details
Please enter correct booking details for Rosie:invalid card details
Please enter correct booking details for Mohan:invalid card details
Please enter correct booking details for Nisha:invalid card details
```

JUNIT Test Snapshots: -

- Static Database Creation:

```

Run: DatasetControllerTest
Tests passed: 2 of 2 tests - 4 ms
DatasetControllerTest (test.controller) 4 ms /Library/Java/JavaVirtualMachines/jdk-18.jdk/Contents/Home/bin/java -ea -Didea.test.cyclic.buffer.size=1048576 -javaagent:/Applications/IntelliJ IDEA.app/Content:
  ✓ datasetFound 4 ms
  ✓ datasetNotFound 0 ms
Process finished with exit code 0

Tests passed: 2 (moments ago)

```

- Input Controller Validation:

```

Run: InputControllerTest
Tests passed: 4 of 4 tests - 30 ms
InputControllerTest (test.controller) 30 ms /Library/Java/JavaVirtualMachines/jdk-18.jdk/Contents/Home/bin/java -ea -Didea.test.cyclic.buffer.size=1048576 -javaagent:/Applications/IntelliJ IDEA.app/Content:
  ✓ generateOutputFile 24 ms
  ✓ startOrderFail 0 ms
  ✓ startOrderPass 0 ms
  ✓ checkFlight 6 ms
Process finished with exit code 0

Tests passed: 4 (moments ago)

```

- Seat Validation:

```

Run: FlightSeatValidationTest
Tests passed: 2 of 2 tests - 4 ms
FlightSeatValidationTest (test.stateHandle) 4 ms /Library/Java/JavaVirtualMachines/jdk-18.jdk/Contents/Home/bin/java -ea -Didea.test.cyclic.buffer.size=1048576 -javaagent:/Applications/IntelliJ IDEA.app/Content:
  ✓ validateFlightSeatAbsence 4 ms Seats Validated successfully
  ✓ validateFlightSeatPresence 0 ms
Process finished with exit code 0

Tests passed: 2 (moments ago)

```

- Flight Validation:

```

Run: FlightPresenceValidationTest
Tests passed: 2 of 2 tests – 5ms
FlightPresenceValidationTest (test.state) 5ms /Library/Java/JavaVirtualMachines/jdk-18.jdk/Contents/Home/bin/java -ea -Didea.test.cyclic.buffer.size=1048576 -javaagent:/Applications/IntelliJ IDEA.app/Content: ↑
  ✓ validateFlightAbsence      5ms Flight Availability Validated successfully
  ✓ validateFlightPresence    0ms
Process finished with exit code 0

Event Log
47:13 LF UTF-8 4 spaces P main

```

- Credit Card Validation:

```

Run: CardValidationTest
Tests passed: 2 of 2 tests – 4ms
CardValidationTest (test.stateHandler) 4ms /Library/Java/JavaVirtualMachines/jdk-18.jdk/Contents/Home/bin/java -ea -Didea.test.cyclic.buffer.size=1048576 -javaagent:/Applications/IntelliJ IDEA.app/Content: ↑
  ✓ validateCreditCardFail     4ms Credit Card Validated successfully
  ✓ validateCreditCardSuccess 0ms
Process finished with exit code 0

Event Log
29:48 LF UTF-8 4 spaces P main

```

- Error File Creation validation:

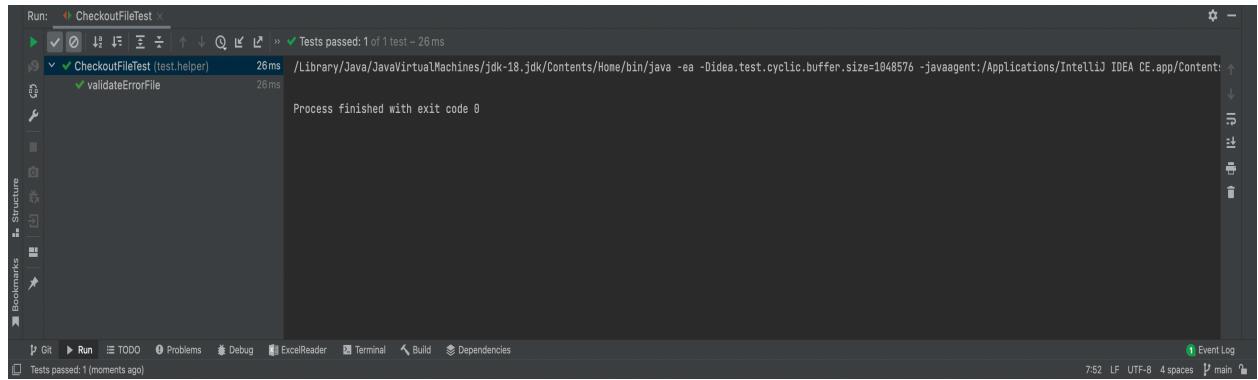
```

Run: ErrorFileTest
Tests passed: 1 of 1 test – 25ms
ErrorFileTest (test.helper) 25ms /Library/Java/JavaVirtualMachines/jdk-18.jdk/Contents/Home/bin/java -ea -Didea.test.cyclic.buffer.size=1048576 -javaagent:/Applications/IntelliJ IDEA.app/Content: ↑
  ✓ validateErrorFile          25ms
Process finished with exit code 0

Event Log
14:22 LF UTF-8 4 spaces P main

```

- Checkout File Creation validation:



GITHUB Link - <https://github.com/gopinathsjsu/individual-project-NishaMDev>