

# **FLIGHT BOOKING SYSTEM**

**OUTPUTS, REPORTS AND ANALYSIS**

**BY ANANYA NAYAK**

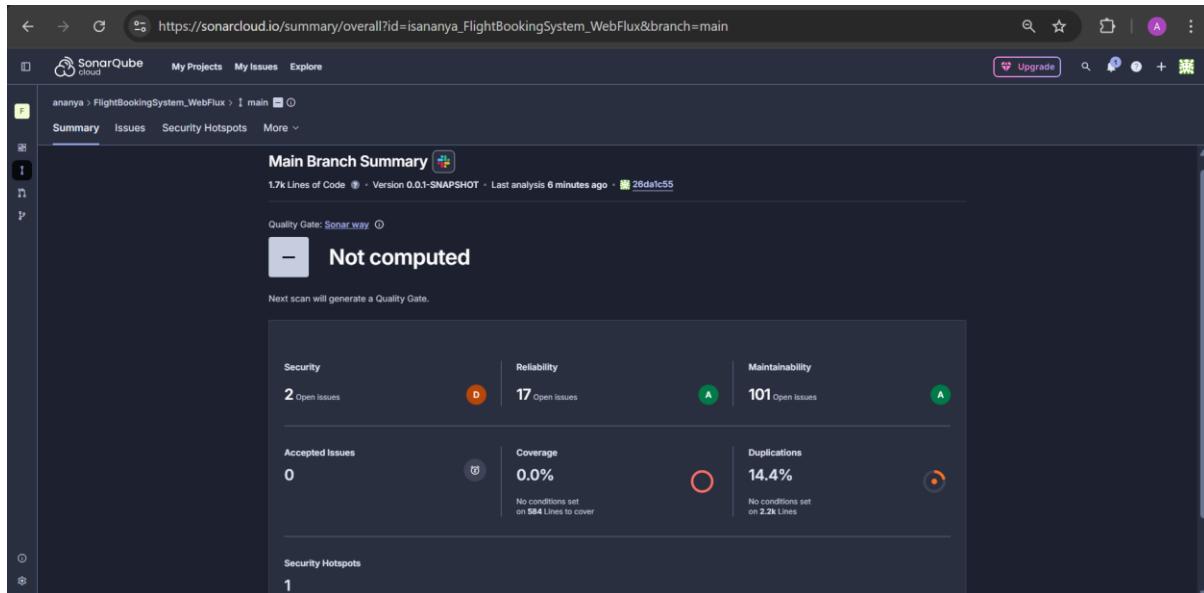
# SonarQube Analysis Reports

## Running analysis :

```
C:\Windows\System32\cmd.e > + <

[INFO] 12:51:25.448 * Files successfully loaded: "40" out of "40"
[INFO] 12:51:25.448 * Protobuf reading complete | memory total=1478 | free=799 | used=678 (MB)
[INFO] 12:51:25.517 * Build architecture.graph.java.namespace_graph.default_perspective hierarchy graph complete (filtered=false) | memory total=1478 | free =799 | used=687 (MB)
[INFO] 12:51:25.517 Sensor JavaArchitectureSensor [architecture] (done) | time=1324ms
[INFO] 12:51:25.582 Sensor Zero Coverage Sensor
[INFO] 12:51:25.583 Sensor Java CPD Block Indexer
[INFO] 12:51:25.649 Sensor Java CPD Block Indexer (done) | time=66ms
[INFO] 12:51:26.346 Inferred api base url 'https://api.sonarcloud.io' from host url 'https://sonarcloud.io'.
[INFO] 12:51:26.349 ----- Gather SCA dependencies on project
[INFO] 12:51:26.349 Checking if SCA is enabled for organization ananya
[INFO] 12:51:26.349 Dependency analysis skipped
[INFO] 12:51:26.918 CPD Executor 15 files had no CPD blocks
[INFO] 12:51:26.918 CPD Executor Calculating CPD for 21 files
[INFO] 12:51:26.937 CPD Executor CPD calculation finished (done) | time=27ms
[INFO] 12:51:26.938 SCM Publisher SCM provider for this project is: git
[INFO] 12:51:26.939 SCM Publisher 41 source files to be analyzed
[INFO] 12:51:29.013 SCM Publisher 39/41 source files have been analyzed (done) | time=2072ms
[WARNING] 12:51:29.013 Missing blame information for the following files:
[WARNING] 12:51:29.013   pom.xml
[WARNING] 12:51:29.013   * src/main/java/com/chubb/FlightBookingSystem/controller/BookingController.java
[WARNING] 12:51:29.013 This may lead to missing/broken features in SonarCloud
[INFO] 12:51:32.187 Analysis report generated in 271ms, dir size=490 KB
[INFO] 12:51:32.349 Analysis report compressed in 136ms, zip size=169 KB
[INFO] 12:51:33.532 Analysis report uploaded in 1182ms
[INFO] 12:51:33.534 ANALYSIS SUCCESSFUL, you can find the results at: https://sonarcloud.io/dashboard?id=isananya_FlightBookingSystem_WebFlux
[INFO] 12:51:33.534 Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
[INFO] 12:51:33.534 More about the report processing at https://sonarcloud.io/api/ce/task?id=AZq0vT6YCaV_j4Mn-3MB
[INFO] 12:51:33.534 ----- Upload SCA dependency files
[INFO] 12:51:35.740 Sensor cache published successfully
[INFO] 12:51:35.815 Analysis total time: 58.099 s
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:28 min
[INFO] Finished at: 2025-11-24T12:51:35+05:30
[INFO]
```

## Before Improvement :



After setting up coverage (jacoco) :

SonarCloud Main Branch Summary for FlightBookingSystem\_WebFlux (main branch). The analysis was run 1 minute ago. The project has 1.7k Lines of Code, Version 0.0.1-SNAPSHOT, and a Quality Gate: Sonar way.

**Status:** Passed

**Metrics:**

- Security:** 2 Open issues (D)
- Reliability:** 17 Open issues (A)
- Maintainability:** 101 Open issues (A)
- Accepted Issues:** 0
- Coverage:** 54.5% (No conditions set on 622 Lines to cover)
- Duplications:** 14.2% (No conditions set on 2.2k Lines)
- Security Hotspots:** 1

## Final Analysis

SonarCloud Main Branch Summary for FlightBookingSystem\_WebFlux (main branch). The analysis was run 1 minute ago. The project has 1.6k Lines of Code, Version 0.0.1-SNAPSHOT, and a Quality Gate: Sonar way.

**Status:** Passed

**Metrics:**

- Security:** 0 Open issues (A)
- Reliability:** 0 Open issues (A)
- Maintainability:** 88 Open issues (A)
- Accepted Issues:** 0
- Coverage:** 65.0% (No conditions set on 558 Lines to cover)
- Duplications:** 3.8% (No conditions set on 2.3k Lines)
- Security Hotspots:** 1

# JMeter Summary Reports

## 20 samples :

Summary Report										
Name:	Summary Report									
Comments:										
Write results to file / Read from file										
Filename	<input type="button" value="Browse..."/>		Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes		<input type="button" value="Configure"/>					
Label	# Samples ↓	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB...	Sent KB/sec	Avg. Bytes
Add User	20	5	4	16	2.55	0.00%	20.9/sec	2.21	7.70	108.0
Add Flight	20	3	2	5	1.02	0.00%	21.2/sec	1.83	7.99	88.5
Add Schedule	20	4	2	8	1.20	100.00%	21.2/sec	244.19	7.69	11815.0
Search Flights	20	2	2	4	0.62	0.00%	21.2/sec	8.16	7.34	395.0
Add Booking	20	4	3	6	0.81	0.00%	21.1/sec	0.78	14.81	38.0
Get ticket usi...	20	6	4	10	1.32	0.00%	21.1/sec	11.41	3.68	555.0
Booking His...	20	7	5	10	1.26	0.00%	21.1/sec	11.41	4.03	555.0
Cancel Booki...	20	2	1	4	0.62	100.00%	21.2/sec	0.97	4.32	47.0
TOTAL	160	4	1	16	2.14	25.00%	162.3/sec	269.43	55.30	1700.2

## 50 samples:

Summary Report										
Name:	Summary Report									
Comments:										
Write results to file / Read from file										
Filename	<input type="button" value="Browse..."/>		Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes		<input type="button" value="Configure"/>					
Label	# Samples ↓	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB...	Sent KB/sec	Avg. Bytes
Add User	50	5	3	17	1.88	0.00%	50.8/sec	5.35	18.72	108.0
Add Flight	50	4	3	12	1.49	0.00%	51.4/sec	4.46	19.38	88.5
Add Schedule	50	5	4	8	1.21	100.00%	51.5/sec	594.13	18.71	11815.0
Search Flights	50	3	2	5	0.75	0.00%	51.7/sec	19.92	17.91	395.0
Add Booking	50	5	4	8	1.04	0.00%	51.7/sec	1.92	36.22	38.0
Get ticket usi...	50	7	5	14	1.55	0.00%	51.7/sec	28.00	9.03	555.0
Booking His...	50	7	5	12	1.47	0.00%	51.8/sec	28.05	9.91	555.0
Cancel Booki...	50	2	1	6	0.89	100.00%	52.2/sec	2.40	10.66	47.0
TOTAL	400	5	1	17	2.24	25.00%	395.3/sec	656.28	134.73	1700.2

## 100 Samples :

Summary Report										
Name:	Summary Report									
Comments:										
Write results to file / Read from file										
Filename	<input type="button" value="Browse..."/>		Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes		<input type="button" value="Configure"/>					
Label	# Samples ↓	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB...	Sent KB/sec	Avg. Bytes
Add User	100	17	3	63	12.71	1.00%	97.3/sec	21.31	35.89	224.4
Add Flight	100	15	3	58	11.80	0.00%	95.8/sec	8.32	36.10	88.9
Add Schedule	100	16	3	39	10.63	100.00%	93.0/sec	1073.31	33.79	11815.0
Search Flights	100	10	2	51	8.14	0.00%	92.5/sec	35.68	32.07	395.0
Add Booking	100	23	4	61	15.03	0.00%	90.3/sec	3.35	63.34	38.0
Get ticket usi...	100	28	6	79	18.09	0.00%	87.3/sec	47.29	15.25	555.0
Booking His...	100	27	6	96	18.98	0.00%	86.7/sec	46.97	16.59	555.0
Cancel Booki...	100	8	2	46	6.96	100.00%	86.8/sec	3.98	17.72	47.0
TOTAL	800	18	2	96	15.10	25.12%	664.5/sec	1112.69	226.52	1714.8

# Postman Screenshots

## Add User:

A screenshot of the Postman application interface. The request method is POST, the URL is <http://localhost:9090/signup>, and the body is a JSON object with fields: name, emailId, mobileNo, gender, age, and password. The response status is 201 Created, and the response body is a single string: 692467c9c1155f4c5682eae8.

```
1 {  
2   "name": "ananya",  
3   "emailId": "ananya@example.com",  
4   "mobileNo": "1234567890",  
5   "gender": "Female",  
6   "age": 21,  
7   "password": "SecurePass123!"  
8 }
```

Body Cookies Headers (2) Test Results ⏱ 201 Created • 9 ms • 108 B • 🌐 | e.g. Save Response ⚙️ Raw ▾ ▶ Preview ⚡ Visualize ▾ 1 692467c9c1155f4c5682eae8

## Add User fail:

A screenshot of the Postman application interface. The request method is POST, the URL is <http://localhost:9090/signup>, and the body is the same JSON object as the successful request. The response status is 409 Conflict, and the response body is a JSON object indicating a conflict because the user with email id ananya@example.com already exists.

```
1 {  
2   "timestamp": "2025-11-24T14:14:32.048Z",  
3   "path": "/signup",  
4   "status": 409,  
5   "error": "Conflict",  
6   "requestId": "1e66b074-6642",  
7   "message": "User with email id ananya@example.com already exists",  
8 }
```

Body Cookies Headers (2) Test Results ⏱ 409 Conflict • 5 ms • 11.44 KB • 🌐 | e.g. Save Response ⚙️ { } JSON ▾ ▶ Preview ⚡ Debug with AI ▾ 1 { 2 "timestamp": "2025-11-24T14:14:32.048Z", 3 "path": "/signup", 4 "status": 409, 5 "error": "Conflict", 6 "requestId": "1e66b074-6642", 7 "message": "User with email id ananya@example.com already exists", 8 }

## Add Flight:

The screenshot shows a POST request to <http://localhost:9090/airline/route/add>. The request body is a JSON object with the following fields:

```
1 {
2   "flightNumber": "AI102",
3   "sourceAirport": "BOM",
4   "destinationAirport": "DEL",
5   "departureTime": "08:30:00",
6   "arrivalTime": "10:45:00",
7   "duration": "PT2H15M"
8 }
```

The response status is 201 Created, with a response time of 16 ms and a response size of 88 B. The response body contains the flight number AI102.

## Add flight fail:

The screenshot shows a POST request to <http://localhost:9090/airline/route/add>. The request body is identical to the successful one above. The response status is 409 Conflict, with a response time of 4 ms and a response size of 11.45 KB. The response body is a JSON object with the following fields:

```
{}
{
  "timestamp": "2025-11-24T14:13:43.702Z",
  "path": "/airline/route/add",
  "status": 409,
  "error": "Conflict",
  "requestId": "1e66b074-6641",
  "message": "Flight Number AI102 already exists",
}
```

## Add schedule:

The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:9090/airline/inventory/add`. The response status is `201 Created` with a response time of 7 ms and a body size of 108 B. The response body is a unique identifier: `69246880c1155f4c5682eae9`.

```
{ "airlineName": "Air India Express", "departureDate": "2025-11-27", "basePrice": 4500.0, "totalSeats": 180, "availableSeats": 180, "flightNumber": "AI102" }
```

## Add schedule fail:

The screenshot shows the Postman interface with a failed API call. The URL is `http://localhost:9090/airline/inventory/add`. The response status is `409 Conflict` with a response time of 6 ms and a body size of 11.54 KB. The response body is a JSON object containing details of the conflict.

```
{ "timestamp": "2025-11-24T14:15:42.368Z", "path": "/airline/inventory/add", "status": 409, "error": "Conflict", "requestId": "1e66b074-6645", "message": "Schedule with Flight NumberAI102 already scheduled on 2025-11-27", }
```

## Search flight:

The screenshot shows a Postman request for searching flights. The URL is `http://localhost:9090/flights/search`. The request method is POST. The body is a JSON object:

```
1 {
2     "sourceAirport": "BOM",
3     "destinationAirport": "DEL",
4     "roundTrip": false,
5     "departureDate": "2025-11-27",
6     "passengerCount": 2
7 }
```

The response status is 200 OK, with a response time of 5 ms and a response size of 395 B. The response body is:

```
1 {
2     "departureCount": 1,
3     "returnCount": 0,
4     "departure": [
5         {
6             "airlineName": "Air India Express",
7             "arrivalTime": "10:45:00",
8             "basePrice": 4500.0,
9         }
10    ]
11 }
```

## Booking:

The screenshot shows a Postman request for booking a flight. The URL is `http://localhost:9090/booking`. The request method is POST. The body is a JSON object:

```
1 {
2     "emailId": "user@gmail.com",
3     "departureScheduleId": "692466ecc1155f4c5682ea81",
4     //   "returnScheduleId" : "6923f9cb4a5569b042ffaa45",
5     "roundTrip": false,
6     "passengerCount": 2,
7     "passengers": [
8         {
9             "firstName": "ananya",
10            "lastName": "nayak",
11            "age": 21,
12            "departureSeatNumber": "2B",
13            //   "returnSeatNumber": "2B",
14            "gender": "FEMALE",
15            "otherDetails": "none"
16        }
17    ]
18 }
```

The response status is 201 Created, with a response time of 11 ms and a response size of 116 B. The response body is:

```
1 Booking Successful! PNR : XUg5FM
```

## Get tickets by pnr:

The screenshot shows a Postman interface with the following details:

- Method:** GET
- URL:** <http://localhost:9090/ticket/2ubYPL>
- Headers:** (6) - Includes Authorization, Content-Type, and Accept.
- Body:** None selected.
- Response:** Status: 200 OK, Time: 27 ms, Size: 533 B. The response body is a JSON array containing one ticket object:

```
[{"id": "2ubYPL", "firstName": "ananya", "lastName": "nayak", "age": 21, "gender": "FEMALE", "seatNumber": "2A", "mealOption": "VEG", "date": "2025-11-29", "fromAirport": "BOM", "toAirport": "DEL", "fromTime": "08:30:00", "toTime": "10:45:00", "status": "CONFIRMED"}]
```

## Booking history by email:

The screenshot shows a Postman interface with the following details:

- Method:** GET
- URL:** <http://localhost:9090/booking/history/user@gmail.com>
- Headers:** (6) - Includes Authorization, Content-Type, and Accept.
- Body:** None selected.
- Response:** Status: 200 OK, Time: 8 ms, Size: 986 B. The response body is a JSON array containing two booking history entries:

```
[{"id": "1", "firstName": "Riya", "lastName": "Rao", "age": 22, "gender": "FEMALE", "seatNumber": "2D", "mealOption": "NONVEG", "date": "2025-11-29", "fromAirport": "BOM", "toAirport": "DEL", "fromTime": "08:30:00", "toTime": "10:45:00", "status": "CONFIRMED"}, {"id": "2", "firstName": "ananya", "lastName": "nayak", "age": 21, "gender": "FEMALE", "seatNumber": "2A", "mealOption": "VEG", "date": "2025-11-29", "fromAirport": "BOM", "toAirport": "DEL", "fromTime": "08:30:00", "toTime": "10:45:00", "status": "CONFIRMED"}]
```

## Delete booking :

The screenshot shows the POSTMAN API client interface. At the top, the URL is set to `http://localhost:9090/booking/cancel/XUg5FM`. The method is selected as `DELETE`. The `Body` tab is active, showing the message "This request does not have a body". Below the request details, the response is displayed. The status code is `200 OK`, with a response time of `11 ms` and a size of `38 B`. The response body contains the number `1`.