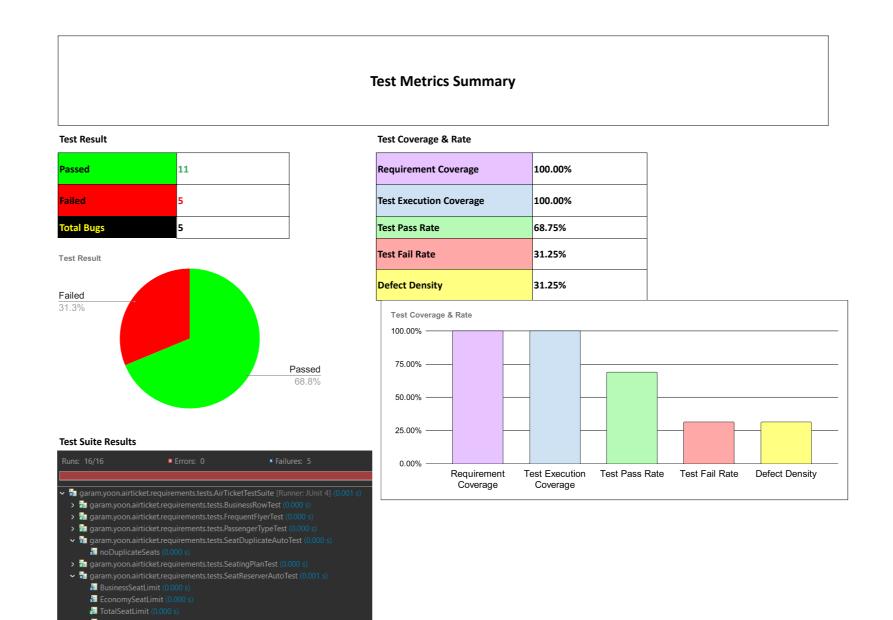
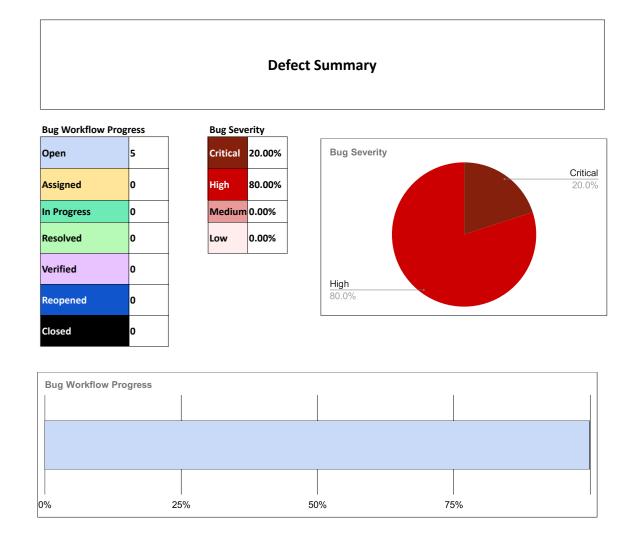
	Test Sum	nmary and Execution R	eport						
		Test Types:	Functional, Validation, and Negative Testing. Automated unit and integration testing using the JUnit framework; requirement-based and scenario-driven validation of core modules (SeatReserver, SeatingPlan, SelfTicket, Passenger).						
	Personal Project	Test Methodologies:							
		Pre-condition	System initialized with 14 total seats (2 Business, 12 Economy); ticket sequence starts at 1,000,000; console I/O simulation enabled for test data DS1–DS9.						
Project Name:	Airplane Ticket Java Application	Dependencies:	Modules: SeatingPlan, SeatReserver, SellTicket, Passenger, and Ticket class dependent on internal seat and passenger state persistence.						
Module Name:	Ticket Purchase Flow	Test Priority	P1 - Seat allocation limits, ticket pricing, and rule validation P2 - Input handling, passenger type, and business logic conditions P3 - Edge-case prompt and loop tests						
Release Version:	v1.0	Total Requirements	11						
Test Designed by:	Garam Yoon	Covered Requirements	BR: 01, 02, 03, 04, 05, 07 NO: 01, 02, 03, 04						
Test Designed date:	2025-09-02	Total Test Cases	16						
Test Executed by:	Garam Yoon	Tested Requirements	11						
Test Execution date:	2025-09-07 ~ 2025-09-13	Executed Tests	16						
		Legends							
Test Data Sets Definition (DS): DS1: "Y", "y", "Yes" - Affirmative input to DS2: "N", "n", "No" - Negative input to DS3: (empty) - Null or empty input to DS4: "quit", 123 - Invalid input for pro DS5: "Garam", "garam" - Sample first DS6: "Yoon", "yoon" - Sample last na	used for No-type prompts. sed for validation testing. ompt validation testing. name input.	Q2-B: "Do you want BUSINESS class?	Q1: "Do you want to purchase a ticket?" Q2-B: "Do you want BUSINESS class?" Q2-E: "Do you want ECONOMY class?" Q3: "First name?" Q4: "Last name?"						
Output Messages:		Repetition Map:							
	seat 1B at \$750.0 - Standard success message after ticke		1R: Repeat 1 - First repetition due to invalid input. 2R: Repeat 2 - Second repetition due to invalid input.						
ssuance. Validation Message: "Please answer	Y or N" - Prompt validation message for incorrect input								
.oop Definition:									
.1: Loop1 - First full execution of the .2: Loop2 - Second purchase cycle ite .3: Loop3 - Third loop or system re-ei	ration.								





Feature (Class)	Module (Method)	Depth	Case Type	Priority	Pre-Condition	Test Case ID	Business Requirement ID (s)	Test Title	Test Summary	Test Steps	Test Data (Data Sets)	Expected Result	Post-condition	Actual Result	Status	Retested Date	Bug ID	Reference	Bug Status	Severity	ScreenShot
Seat Management (BusinessRowTest)	Business Row Configuration (row1Seats)	L1	Unit	P1	App initialized with seat map	JUNIT-BUSROW-TC01	BR-01, NO-02	Business Row Setup	Validate business row seats are 1A/1B only	Load plan → read row1 seats → assert count=2 & labels	N/A	1A,1B only	Seat map retained	As expected	Pass	2025-09-07	N/A	Console log	N/A	N/A	N/A
Seat Management (SeatDuplicateAutoTest)	noDuplicateSeats I	L1	Unit	P1	14 seats available	JUNIT-SEATDUP-TC02	BR-01, BR-04, NO-01, NO-02	Seat Uniqueness	No seat should be assigned twice	Auto-sell all → collect seat labels → check uniqueness	N/A	14 unique seats	Assignment log saved	Duplicate detected (e.g., 1A repeated)	Fail	2025-09-07	BUG-JUNIT-TC02-001	Stacktrace + console	Open	Critical	https://drive.google. com/file/d/1qiXrPICFIUYX 7a2DfetQ8xm- RPpSesqP/view? usp=sharing
Seating Layout (SeatingPlanTest)	testSeatingPlan I	L1	Unit	P1	App started	JUNIT-SEATPLAN-TC03	BR-01, NO-01, NO-02, NO-03	Seating Plan Verification	Map equals 14 seats: row1 business, rows2–economy	Build plan → count seats → verify rows/letters	N/A	14 seats; 1A–4D	Plan cached	As expected	Pass	2025-09-07	N/A	Console log	N/A	N/A	N/A
Seat Management (SeatReserverAutoTest)	BusinessSeatLimit I	L1	Unit	P1	Business seats free	JUNIT-SEATRES-TC04	BR-01, NO-02	Business Seat Limit	Only two business seats may be issued	Sell 1A,1B → attempt 3rd → expect reject	N/A	3rd rejected	Counters unchanged	3rd allowed	Fail	2025-09-07	BUG-JUNIT-TC04-002	Stacktrace + console	Open	High	https://drive.google. com/file/d/1C3wUXFBOd BHRI14rdEmxDt9fVuaHIT s8/view?usp=sharing
Seat Management (SeatReserverAutoTest)	EconomySeatLimit I	L1	Unit	P1	Economy seats free	JUNIT-SEATRES-TC05	NO-03, NO-01	Economy Seat Limit	Only twelve economy seats may be issued	Sell 12 → attempt 13th → expect reject	N/A	13th rejected	Counters unchanged	13th allowed	Fail	2025-09-08	BUG-JUNIT-TC05-003	Stacktrace + console	Open	High	https://drive.google. com/file/d/194cjlLmripA hE0L5WUfs1DdwC094T_f i/view?usp=sharing
Seat Management (SeatReserverAutoTest)	TotalSeatLimit I	L1	Unit	P1	All seats free	JUNIT-SEATRES-TC06	BR-01, NO-02, NO-03	Total Capacity Limit	No sale after 14th ticket	Sell 14 any order → attempt 15th	N/A	15th rejected	Capacity enforced	As expected	Pass	2025-09-08	N/A	Console log	N/A	N/A	N/A
Ticket Issuance (SellTicketTest)	sellTicket1	L1	Unit	P1	Business seat available	JUNIT-SELL-TC07	BR-02	Business Ticket \$750	Price must be \$750 (no discount)	Build passenger/seat → create ticket	N/A	\$750	Ticket persisted	As expected	Pass	2025-09-09	N/A	Assertion output	N/A	N/A	N/A
Ticket Issuance (SellTicketTest)	sellTicket2	L1	Unit	P1	Economy seats available	JUNIT-SELL-TC08	BR-05, NO-03, NO-4	Employee 50% Economy	Employee economy price \$250 and cap 12	Create staff → economy seat → assert \$250	DS1	\$250	Ticket persisted	Price=\$500 or 13th allowed	Fail	2025-09-10	BUG-JUNIT-TC08-005	Stacktrace + console	Open	High	https://drive.google. com/file/d/1DAW7Lpou wZen3nhEqW5L9LWxK1V eEr99/view?usp=sharing
Discount Policy (FrequentFlyerTest)	testFrequentFlyerIsNotDi scountable	L1	Unit	P2	Passenger type=Frequent Flyer	JUNIT-FFLY-TC09	BR-04, BR-07	FF Not Discountable	FF should not implement Discountable	Instantiate FF → instanceof check	N/A	No discount capability	N/A	As expected	Pass	2025-09-11	N/A	Assertion output	N/A	N/A	N/A
Passenger Management (PassengerTypeTest)	frequentFlyerType I	L1	Unit	P2	App running	JUNIT-PASS-TC10	NO-04	FF Type Recognition	Stores type=FF	Set flags → assert type	DS1	Type=FF	Passenger stored	As expected	Pass	2025-09-11	N/A	Test log	N/A	N/A	N/A
Passenger Management (PassengerTypeTest)	employeeType I	L1	Unit	P2	App running	JUNIT-PASS-TC11	NO-04	Employee Type Recognition	Stores type=EMP	Set flags → assert type	DS1	Type=EMP	Passenger stored	As expected	Pass	2025-09-11	N/A	Test log	N/A	N/A	N/A
Seat Management (SeatReserverAutoTest)	BusinessSoldOutAllowsEc onomySales	12	Unit	P2	Business sold out	JUNIT-SEATRES-TC12	NO-01, NO-03	Class Independence	Economy sales continue after business sold out	Sell 2 business → sell 12 economy	N/A	All 12 economy succeed	Seats reduced accordingly	Economy blocked unexpectedly	Fail	2025-09-12	BUG-JUNIT-TC12-004	Stacktrace + console	Open	High	https://drive.google. com/file/d/1hDJGlhMtQp Vp_om5RUvFAvPuLa0N5 zu3/view?usp=sharing
Ticket Issuance (SellTicketTest)	sellTicket3	L1	Unit	P2	FF passenger active	JUNIT-SELL-TC13	BR-03	FF No Discount	FF pays base price	Create FF → business/economy price check	N/A	Base price	Ticket persisted	As expected	Pass	2025-09-12	N/A	Assertion output	N/A	N/A	N/A
Ticket Issuance (SellTicketTest)	sellTicket4	L1	Unit	P2	Regular passenger	JUNIT-SELL-TC14	BR-06	Regular Economy \$500	Regular passenger pays \$500	Create regular → economy ticket	N/A	\$500	Ticket persisted	As expected	Pass	2025-09-12	N/A	Assertion output	N/A	N/A	N/A
Ticket Issuance (SellTicketTest)	sellTicket5	L1	Unit	P2	App active	JUNIT-SELL-TC15	BR-02	Sequential Ticket Numbers	Numbers increment by 1	Issue multiple tickets → compare IDs	N/A	#ERROR!	Counter updated	As expected	Pass	2025-09-13	N/A	Console log	N/A	N/A	N/A
Passenger Management (PassengerTypeTest)	regularType I	L1	Unit	Р3	App running	JUNIT-PASS-TC16	NO-04	Regular Default	Default to Regular if not FF/EMP	Set both N → assert type	DS2	Type=REG	Passenger stored	As expected	Pass	2025-09-13	N/A	Test log	N/A	N/A	N/A