EAST WEST UNIVERSITY

PROJECT REPORT

Topic : “Travel & Tourism Management System”

Course Title: Software Testing and Quality Assurance

Course Code: CSE430

Section: 01

Project Group: 01

Semester: Summer 2025

Submitted by

Student’s Name	Student’s ID
Md. Tanvir Hossain	2022-1-60-196
Bushra Hoque	2022-1-60-154
Md. Ali Haider	2022-1-60-193
Nabila Tabassum	2022-1-60-182
Nirzona Binta Badal	2021-2-60-051

Under the Guidance of,

Anika Tabassum
Lecturer , Department of CSE
East West University

Date of submission : 26-08-202

Abstract

The **Travel & Tourism Management System** is a Java-based software application designed to handle the various operational activities of a tourism management efficiently. It streamlines the processes of guest booking status, package status, payment, review system, and logging system. Developed using Java and tested with JUnit, the system follows a modular design with over 60 classes to ensure scalability and maintainability. Each component, such as Booking slot, Guest status, Reservation, Payment and Security is carefully tested to ensure system reliability. The project aims to automate routine of travel and tourism management's tasks, minimize human errors, and provide a seamless experience for Tourism's staff and guests.

Introduction

Managing a travel and tourism business involves complex workflows, including guest bookings, package reservations, payment processing, customer reviews, and security logging. Traditional manual systems are often prone to delays, errors, and inefficiencies, which negatively affect staff productivity and guest satisfaction.

To overcome these challenges, we developed a **Travel & Tourism Management System** using the Java programming language and object-oriented design principles. This project consists of over 60 interconnected classes, each representing real-world tourism entities such as **Booking, Guest, Package, Payment, and Security**.

The system supports the following core functionalities:

- Managing guest booking records and package reservations
- Tracking and updating package availability and status
- Processing guest payments and generating receipts
- Handling customer reviews and feedback
- Maintaining a secure logging system for staff and guests

Testing plays a vital role in ensuring the system's correctness and reliability. Therefore, extensive **JUnit test cases** have been created for each class and functionality. The project is developed in **NetBeans IDE 8.2**, ensuring industry compatibility and maintainability.

By simulating real-world tourism operations, this system provides a solid foundation for understanding **software development, modular system design, testing methodologies, and project management principles**, while also serving as a practical solution to streamline travel and tourism management activities.

Functionality Overview

Serial	Testing Class	Testing Methods
1	AdminMenuTest	testDisplayAdminMenu, testDisplayAdminMenuBehavior, testConstructorExists, testConstructorParameterCount, testPrintAdminHeaderExists, testPrintAdminMenuOptionsExists, testHandleAdminMenuChoiceExists, testHandleUserManagementExists, testViewAllUsersExists, testSearchUserExists, testBlockUnblockUserExists, testAddNewAdminExists, testHandlePackageManagementExists, testHandleBookingManagementExists, testHandlePaymentManagementExists, testGenerateDailyReportExists, testGenerateMonthlyReportExists, testHandleSystemBackupExists, testLogoutExists, testClearScreenExists, testClassIsPublic, testClassPackage, testClassName, testHasFields, testClassNotAbstract, testDisplayAdminMenuReturnType, testDisplayAdminMenuIsPublic, testConstructorIsPublic, testPrivateMethodsExist
2	AdminTest	testGetRole, testGetAdminLevel, testSetAdminLevel, testGetPermissions, testSetPermissions, testConstructorWithNullValues, testSetAdminLevelNull, testSetAdminLevelEmpty, testSetPermissionsNull, testSetPermissionsEmpty, testPermissionsWithNullElements,

		testGetRoleConsistency, testAdminLevelAfterMultipleChanges, testSpecialCharactersInAdminLevel, testUnicodeCharacters, testSetAdminLevelMethodException, testExceptionMessage, testPermissionsDuplicates, testLargePermissionsList, testObjectStateAfterException, testMultipleAdminInstances, testMultipleAdminInstancess
3	BackupManagerTest	testCreateBackup, testRestoreBackup
4	DateUtilTest	testGetCurrentDate, testGetCurrentDateTime, testFormatDate, testParseDate, testParseDateTime, testDaysBetween, testIsValidDate, testIsValidDateTime, testFormatDateTime, testAddDays, testSubtractDays, testIsDateBefore, testIsDateAfter, testIsDateEqual, testGetFirstDayOfMonth, testGetLastDayOfMonth, testGetDayOfWeek, testGetDayName, testGetMonthName, testGetYear, testGetMonth, testGetDay, testIsWeekend, testIsLeapYear, testHoursBetween, testMinutesBetween, testAddHours, testAddMinutes, testGetCurrentTimestamp, testFormatTimestamp, testParseTimestamp
5	FileHandlerTest	testWriteToFile, testWriteToFileWithNullFilename, testWriteToFileWithEmptyFilename, testWriteToFileWithNullData, testWriteToFileWithEmptyData, testWriteToFileWithLongData, testWriteToFileWithSpecialCharacters, testWriteToFileMultipleTimes, testReadFromFile, testReadFromNonExistentFile,

		testReadFromFileWithNullFilename, testReadFromFileWithEmptyFilename, testReadFromEmptyFile, testClearFile, testClearNonExistentFile, testClearFileWithNullFilename, testClearFileWithEmptyFilename, testFileExists, testFileExistsWithNonExistentFile, testWithSpecialCharactersInFilename, testWithLongFilename, testConcurrentWrites, testWriteAfterClear, testWriteWithNewlineCharacters, testWriteWithTabCharacters
6	HotelManagerTest	testAddHotelBehavior, testRemoveHotelEmptyString, testRemoveHotelNull, testGetHotelByIdEmpty, testGetHotelByIdNull, testUpdateHotelNull, testUpdateHotelBehavior, testGetAllHotelsReturns, testGetAllHotelsType, testGetActiveHotelsReturns, testGetActiveHotelsType, testSearchHotelsEmpty, testSearchHotelsNull, testSearchHotelsNormal, testGetHotelsByLocationEmpty, testGetHotelsByLocationNull, testGetHotelsByLocationNormal, testGetHotelsByRatingZero, testGetHotelsByRatingHigh, testGetHotelsByRatingNegative, testGetHotelsWithAvailableRooms, testGetHotelsWithAvailableRoomsType, testSearchAvailableRoomsEmpty, testSearchAvailableRoomsNullLocation, testBookRoomEmptyHotelId, testBookRoomEmptyRoomId, testBookRoomNullParams,

		testBookRoomNonExistingHotel, testCheckoutRoomEmptyHotelId, testCheckoutRoomEmptyRoomId, testCheckoutRoomNullParams, testCheckoutRoomNonExistingHotel, testAddRoomToHotelEmptyId, testAddRoomToHotelNullParams, testAddRoomToHotelNonExisting, testRemoveRoomFromHotelEmptyId, testRemoveRoomFromHotelNullParams, testRemoveRoomFromHotelNonExisting, testGetTotalHotelsInitial, testGetTotalHotelsType, testGetActiveHotelsCountInitial, testGetActiveHotelsCountType, testGetAverageHotelRatingInitial, testGetAverageHotelRatingType, testGetHighestRatedHotelInitial, testGetHighestRatedHotelType, testGetHotelCountByLocationInitial, testGetHotelCountByLocationType, testGetTotalRoomsInitial, testGetTotalRoomsType, testGetAvailableRoomsCountInitial, testGetAvailableRoomsCountType, testGetOccupancyRateInitial, testGetOccupancyRateType, testGetOccupancyRateRange, testSearchHotelsSpecialChars, testGetHotelsByLocationSpecialChars, testGetHotelsByRatingMax, testSearchAvailableRoomsSuiteType,

		testMultipleOperations, testMethodConsistency, testStateIndependence
7	HotelTest	testAddRoom, testRemoveRoom, testGetRoomById, testGetAvailableRooms, testGetHotelId, testSetHotelId, testGetName, testSetName, testGetLocation, testSetLocation, testGetRating, testSetRating, testGetRoomList, testSetRoomList, testGetAddress, testSetAddress, testGetPhone, testSetPhone, testGetEmail, testSetEmail, testGetAmenities, testSetAmenities, testGetDescription, testSetDescription, testIsActive, testSetActive, testToString
8	InvoiceTest	testAddItem, testCalculateTotal, testGenerateInvoiceText, testGetInvoiceId, testSetInvoiceId, testGetBookingId, testSetBookingId, testGetUserId, testSetUserId, testGetSubtotal, testSetSubtotal, testGetTax, testSetTax, testGetDiscount, testSetDiscount, testGetTotalAmount, testSetTotalAmount, testGetIssueDate, testSetIssueDate, testGetDueDate, testSetDueDate, testGetStatus, testSetStatus, testGetItems, testInvoiceConstructor, testInvoiceItem, testComplexCalculation
9	LoginUITest	testDisplayLoginScreenExists, testDisplayLoginScreenReturnType, testDisplayLoginScreenAccessModifier, testDisplayLoginScreenParameterCount, testDisplayRegistrationScreenExists, testDisplayRegistrationScreenReturnType, testDisplayRegistrationScreenAccessModifier, testDisplayRegistrationScreenParameterCount, testDisplayForgotPasswordScreenExists,

		testDisplayForgotPasswordScreenReturnType, testDisplayForgotPasswordScreenAccessModifier, testDisplayForgotPasswordScreenParameterCount, testDisplayLoginHelpExists, testDisplayLoginHelpReturnType, testDisplayLoginHelpAccessModifier, testDisplayLoginHelpParameterCount, testDisplayRegistrationHelpExists, testDisplayRegistrationHelpReturnType, testDisplayRegistrationHelpAccessModifier, testDisplayRegistrationHelpParameterCount, testClassExists, testClassPackage, testClassAccessModifier, testConstructorExists, testConstructorParameterCount, testTotalPublicMethodCount, testPrivateMethodCount, testMethodNamingConvention, testAllRequiredMethodsExist, testClassIsNotAbstract, testClassIsNotFinal
10	RatingTest	testMarkAsHelpful, testVerifyRating, testUnverifyRating, testIsValidRatingValid, testIsValidRatingInvalid, testGetRatingStarsFullStars, testGetRatingStarsPartialStars, testGetRatingCategoryExcellent, testGetRatingCategoryPoor, testIsPositiveRatingTrue, testIsPositiveRatingFalse, testIsNegativeRatingTrue, testIsNegativeRatingFalse, testIsNeutralRatingTrue, testIsNeutralRatingFalse, testGetRatingDescriptionExcellent, testGetRatingDescriptionPoor, testGetRatingSummary, testUpdateRatingValid, testUpdateRatingInvalid, testGetRatingLevel, testGetRatingPercentage,

		testIsHighRatingTrue, testIsHighRatingFalse, testIsLowRatingTrue, testIsLowRatingFalse, testHasCommentsTrue, testHasCommentsFalse, testGetCommentsLengthWithComments, testGetCommentsLengthWithoutComments, testIsDetailedRatingTrue, testIsDetailedRatingFalse, testCalculateAverageRatingWithRatings, testCalculateAverageRatingEmpty, testGetRatingDistribution, testFormatRatingDistribution, testGetRatingId, testSetRatingId, testGetUserId, testSetUserId, testGetUserName, testSetUserName, testGetEntityId, testSetEntityId, testGetEntityType, testSetEntityType, testGetRatingValue, testSetRatingValueValid, testSetRatingValueInvalid, testGetComments, testSetComments, testGetRatingDate, testSetRatingDate, testIsVerified, testSetVerified, testGetCriteria, testSetCriteria, testGetHelpfulCount, testSetHelpfulCount, testIsAnonymous, testSetAnonymous, testGetAspectRated, testSetAspectRated, testToString, testEqualsTrue, testEqualsFalse, testHashCode, testMinimumValidRating, testMaximumValidRating, testBoundaryRating2Point0, testBoundaryRating4Point0, testNullComments, testEmptyStringComments, testWhitespaceOnlyComments, testCalculateAverageRatingNull, testCalculateAverageRatingSingleRating, testCalculateAverageRatingIdentical, testRatingDistributionEmpty,

		testRatingDistributionAllFiveStars, testVeryLongRatingId, testSpecialCharactersEntityId, testUnicodeCharactersUserName, testMultipleHelpfulMarkings, testVerifyUnverifyMultiple, testUpdateRatingMultipleTimes, testRatingPercentageExtremes, testEqualsWithNull, testEqualsWithSameInstance, testEqualsWithDifferentClass, testConstructorWithAllParameters, testAnonymousRatingSummary, testRatingSummaryWithAspectRated, testFormatRatingDistributionSingle, testSetRatingValueBoundaries
11	ReportGeneratorTest	testGenerateDaily, testGenerateDailyContentValidation, testGenerateDailyMultipleTimes, testGenerateDailyPerformance, testGenerateDailyReport, testGenerateDailyReportDifferentDate, testGenerateDailyReportLeapYear, testGenerateDailyReportYearEnd, testGenerateDailyReportNullDate, testGenerateDailyReportEmptyDate, testGenerateMonthly, testGenerateMonthlyContentValidation, testGenerateMonthlyLengthValidation, testGenerateMonthlyConsistency, testGenerateMonthlyReport, testGenerateMonthlyReportDifferentMonth, testGenerateMonthlyReportDecember, testGenerateMonthlyReportPreviousYear,

		testGenerateMonthlyReportNullMonth, testGenerateMonthlyReportEmptyMonth, testGenerateAnnual, testGenerateAnnualContentValidation, testGenerateAnnualSectionsValidation, testGenerateAnnualComprehensiveValidation, testGenerateAnnualReport, testGenerateAnnualReportDifferentYear, testGenerateAnnualReportFutureYear, testGenerateAnnualReportPastYear, testGenerateAnnualReportNullYear, testGenerateAnnualReportEmptyYear, testGenerateSystemReport, testGenerateSystemReportContentValidation, testGenerateSystemReportSectionsValidation, testGenerateSystemReportLengthValidation, testGenerateSystemReportConsistency, testGenerateCustomReport, testGenerateCustomReportRevenue, testGenerateCustomReportBookings, testGenerateCustomReportCustomers, testGenerateCustomReportPackages, testGenerateCustomReportUnknownType, testGenerateCustomReportDifferentDateRange, testGenerateCustomReportLongDateRange, testGenerateCustomReportNullParameters, testGenerateCustomReportEmptyParameters, testConstructor, testConstructorImmediateGeneration, testAllReportTypesConsistency, testAllReportTypesContentValidation, testReportGenerationPerformance,

		testMultipleReportGenerationsConsistency, testBoundaryDateValues, testEdgeCaseDateHandling, testCompleteIntegration, testFinalComprehensiveValidation
12	TransportManagerTest	testAddTransport, testRemoveTransport, testGetTransportById, testUpdateTransport, testGetAllTransports, testGetActiveTransports, testSearchTransports, testGetTransportsByType, testGetAvailableTransports, testGetTransportsByRoute, testBookSeat, testCancelSeat, testAddVehicle, testRemoveVehicle, testGetVehicleById, testGetAllVehicles, testGetOperationalVehicles, testGetVehiclesByType, testGetSeatsForTransport, testGetAvailableSeats, testBookSpecificSeat, testCancelSpecificSeat, testGetTotalTransports, testGetActiveTransportsCount, testGetTotalVehicles, testGetOperationalVehiclesCount, testGetTransportCountByType, testGetAverageOccupancyRate, testGetFullyBookedTransports
13	ValidatorTest	testValidateEmail, testValidatePhone, testValidatePassword, testIsNotEmpty, testIsPositiveNumber, testIsValidRange, testIsValidName

Serial	Testing Class	Testing Methods
14	GuideManagerTest	testAddGuide, testRemoveGuide, testGetGuideById, testUpdateGuide, testGetAllGuides,

		testGetAvailableGuides, testGetGuidesByLanguage, testGetGuidesByRating, testGetGuidesBySpecialization, testSearchGuides, testGetTotalGuides, testGetAvailableGuidesCount, testGetAverageGuideRating, testGetTopRatedGuide, testGetGuideCountByLanguage, testUpdateGuideRating
15	InputHandlerTest	testGetString, testGetInt, testGetDouble, testGetBooleanYes, testGetBooleanNo, testPressEnterToContinue, testCloseScanner
16	PackageManagerTest	testAddPackage, testRemovePackage, testListPackages, testListActivePackages, testGetPackageById, testSearchPackages, testGetPackagesByCategory, testGetPackagesByTourType, testGetPackagesByPriceRange, testGetPackagesByDuration, testGetPackagesByLocation, testGetPackagesByCreator, testUpdatePackage, testActivateDeactivatePackage, testGetTotalPackages, testGetActivePackagesCount, testGetAveragePackagePrice, testGetMostExpensivePackage, testGetCheapestPackage, testGetPackageCountByCategory, testGetPackageCountByTourType, testGetAvailablePackages, testGetFullyBookedPackages
17	RoomTypeTest	testValues, testValueOf, testGetDisplayName, testGetBasePrice
18	SeatAllocatorTest	testAutoAssignSeats_returnsEmptyList, testFindBestAvailableSeat_withPreferredType,

		testFindBestAvailableSeat_withoutPreferredType, testFindBestAvailableSeat_withEmptyList
19	SeatTest	testBookSeat_success, testBookSeat_fail_whenBlocked, testCancelSeat, testGettersAndSetters, testToString_containsSeatNumberAndPassengerName
20	TicketTest	testUseTicket, testGenerateTicketText, testGetAndSetTicketId, testGetAndSetBookingId, testGetAndSetQRcode, testGetAndSetIssueDate, testGetAndSetValidFrom, testGetAndSetValidUntil, testGetAndSetPassengerName, testGetAndSetPassengerPhone, testGetAndSetPackageName, testGetAndSetSeatNumber, testIsUsedAndSetUsed, testToString
21	TransportTest	testCancelSeat, testGetAndSetTransportId, testGetAndSetType, testGetAndSetDeparture, testGetAndSetArrival, testGetAndSetDepartureTime, testGetAndSetArrivalTime, testGetAndSetFare, testGetAndSetTotalSeats, testGetAndSetAvailableSeats, testGetAndSetOperatorName, testIsActiveAndSetActive, testToString
22	TravelAgentTest	testGetRole, testSetAndGetAgencyName, testSetAndGetLicenseNumber, testGetManagedPackagesInitiallyEmpty, testGetAndSetCommissionRate, testInheritanceFromUser

Serial	Testing Class	Testing Methods
23	AuthServiceTest	testLogout testIsUserAuthenticated testHasRole testHasAnyRole testIsAdmin testIsTourist testIsAgent testAuthorizeAccess testAuthorizeAdminAccess testChangePassword testRequestPasswordReset testResetPassword testBlockUser testUnblockUser testIsAccountLocked testLogoutWithNullEmail testHasRoleWithNulls testChangePasswordWithNulls testRequestPasswordResetWithNull testResetPasswordWithNulls testBlockUserWithNulls testUnblockUserWithNull
24	DataSeederTest	testSeedInitialData testSeedInitialDataWithAlreadyPopulatedManagers testSeedInitialDataWithNullManagers testSeedInitialDataWithPartialNullManagers
25	IDGeneratorTest	testGenerateUserId testGeneratePackageId testGenerateBookingId testGenerateHotelId testGenerateTransportId testGenerateInvoiceId testGenerateRandomStringWithLengthOnly testGenerateRandomStringWithCustomChars testIsValidIdFormat testGenerateSecureToken testGenerateRandomStringWithZeroLength testGenerateRandomStringWithNegativeLength testGenerateRandomStringWithEmptyCustomChars testIsValidIdFormatWithWrongPrefix

		testIsValidIdFormatWithNullId testGenerateSecureTokenWithZeroLength testGenerateSecureTokenWithNegativeLength
26	LocationTest	testAddLandmark testRemoveLandmark testGetLocationId testGetDescription testSetDescription_Empty testGetClimate testSetClimate_Null testGetBestTimeToVisit testSetBestTimeToVisit_Empty testToString testGetCity testGetCountry testSetCountry_Null testGetState testSetState_Empty testGetLandmarks testAddLandmark_NullValue testAddLandmark_Duplicate testRemoveLandmark_NotExists testSetLocationId_EmptyString testSetLocationId_Null testSetCity_SpecialCharacters testSetLandmarks_EmptyList testSetLandmarks_NullList testGetLatitude testSetLatitude_OutOfRange testGetLongitude testSetLongitude_OutOfRange
27	MainMenuTest	testEmergencyShutdown testIsSystemRunning_DefaultTrue testGetUserManager testGetAuthService testEmergencyShutdownTwice testIsSystemRunningAfterMultipleOperations testGetUserManagerNotNullEvenAfterShutdown testGetAuthServiceMultipleCallsReturnSameInstance testAuthServiceWorksAfterShutdown testSystemRunningStatePersistence testGetUserManagerMayReturnNullIfNotInitialized

28	MainTest	testMain testGetUserManager testGetPackageManager testGetBookingManager testGetPaymentManager testGetHotelManager testGetTransportManager testGetGuideManager testGetReviewManager testGetRefundManager testGetRoomManager testGetAuthService testGetReportGenerator testSingletonManagers testMainWithNullArgs testMainWithLongArgs testManagersNullSafety
29	PaymentManagerTest	testProcessPayment testFailPayment testRefundPayment testProcessPaymentNonExistent testFailPaymentNonExistent testRefundPaymentNonExistent testRefundPaymentNotCompleted testAddDuplicatePayment testZeroAmountPayment testNegativeAmountPayment testGetPaymentByIdNonExistent testProcessPaymentNullId testFailPaymentNullId testRefundPaymentNullId testGetPaymentByIdNull
30	RoomTest	testIsAvailable testBookRoom testCheckOut testGetRoomId testGetHotelId testSetHotelId_SpecialCharacters testGetRoomNumber testSetRoomNumber_Empty testGetRoomType testIsAvailability testSetAvailability_False testGetPrice testGetCapacity testGetDescription testSetDescription_Empty testIsHasAC testSetHasAC_False testIsHasWiFi testSetHasWiFi_False testIsHasTV testSetHasTV_False testGetBedType testSetBedType_Empty testToString testSetPrice_Negative testSetPrice_Zero

		testSetRoomId_Null testSetRoomId_Empty testSetRoomType_Null testCheckOut_WhenAlreadyAvailable testSetCapacity_Zero testSetCapacity_Negative testSetBedType_Null testBookRoom_WhenAlreadyBooked testToString_AllFieldsNull
31	ScheduleTest	testAddAndRemoveScheduleItem testConfirmAndCancelSchedule testCompleteSchedule testUpdateScheduleItem testGetScheduleItemsByDate testHasConflictWithTime testGenerateScheduleReport testBasicGettersAndSetters testToString testRemoveNonExistingItem testUpdateNonExistingItem testGetScheduleItemsByDateNoMatch testHasConflictWithDifferentDate testEmptyScheduleReport testConstructorWithNullIds testAddScheduleItemWithNullValues testUpdateScheduleItemWithNullValues testGetScheduleItemsByDateWithNull testHasConflictWithNullValues
32	TouristTest	testGetRole testAddLoyaltyPoints testGetPreferences testSetPreferences testGetNationality testSetNationality testGetLoyaltyPoints testGetBookingHistory testAddZeroLoyaltyPoints testAddNegativeLoyaltyPoints testSetEmptyPreferences testSetNullPreferences testSetEmptyNationality testSetNullNationality testEmptyBookingHistory testPreferencesImmutableReference

33	TourPackageTest	testCancelBooking testAddBooking testGetCurrentBookings testSetCurrentBookings testGetMaxCapacity testSetMaxCapacity testGetRating testSetRating testGetTotalReviews testSetTotalReviews_Negative testSetTotalReviews testGetDiscountPercentage testSetDiscountPercentage testGetCategory testSetCategory testGetTourType testSetTourType testIsActive testSetActive testIsFeatured testSetFeatured testGetCreatedBy testSetValidTo testSetCreatedBy testGetCreatedDate testSetCreatedDate testGetLastModified testSetLastModified testGetPackageId testSetPackageId testGetValidFrom testSetValidFrom testGetValidTo testSetCurrentBookings_Negative testSetMaxCapacity_Zero testSetMaxCapacity_Negative testSetCreatedDate_Null testSetRating_OutOfRangeHigh testSetRating_OutOfRangeLow testSetCreatedBy_Empty testAddBooking_WhenAtMaxCapacity testSetLastModified_Empty testSetPackageId_Null testSetDiscountPercentage_Negative testSetDiscountPercentage_Above100 testSetValidTo_Null testSetValidFrom_Empty testCancelBooking_WhenZeroBookings
34	TourTypeTest	testValues testValueOf_Valid testGetDisplayName testGetDescription testOrdinalOrder testValueOf_Invalid testValueOf_Null testEnumConsistency

35	userTest	testSetId testSetName testSetEmail testSetPassword testSetPhone testGetCreatedDate testGetName testGetEmail testGetPassword testGetPhone testIsActive testSetActive testGetRole testToString testGetId

Serial	Testing Class	Testing Methods
36	BookingManagerTest	testAddBooking testRemoveBooking testGetBookingById testUpdateBooking testGetAllBookings testGetBookingsByUser testGetBookingsByPackage testGetBookingsByStatus testGetBookingsByDate testGetBookingsByTravelDate testGetBookingsInDateRange testGetPendingBookings testGetConfirmedBookings testGetCancelledBookings testCancelBooking testConfirmBooking testCompleteBooking testGetTotalRevenue testGetRevenueByPackage testGetTotalBookings testGetBookingsCountByStatus testGetUpcomingBookings testGetTodaysBookings testGetBookingCountsByMonth testSearchBookings testAddNullBooking testRemoveNonExistentBooking testGetBookingByIdInvalid testGetBookingsByUserInvalid testGetBookingsByPackageInvalid testGetBookingsByStatusNull testGetBookingsByInvalidDate testGetBookingsInDateRangeInvalid testCancelBookingInvalidId testConfirmBookingInvalidId

		testCompleteBookingInvalidId testGetRevenueByPackageInvalid testGetBookingsCountByStatusNull testSearchBookingsNoResult
37	BookingStatusTest	testValues testValueOf testGetDisplayName testGetDescription testValueOfInvalidEnumName testValueOfNull testGetDisplayNameNotNull testGetDescriptionNotNull
38	BookingTest	testConfirmBooking testCancelBooking testCompleteBooking testGetAndSetBookingId testGetAndSetUserId testGetAndSetPackageId testGetBookingDate testGetAndSetTravelDate testGetAndSetStatus testGetAndSetNumberOfPeople testGetAndSetTotalAmount testGetAndSetSpecialRequests testGetAndSetHotelId testGetAndSetTransportId testGetAndSetGuideId testIsAndSetPaid testToStringNotNull testConfirmBooking_Idempotent testCancelAfterConfirm_OverridesStatus testSetNumberOfPeopleZero testSetNumberOfPeopleNegative testSetTotalAmountZero testSetTotalAmountNegative testSetTravelDateEmptyString testSetIdsEmptyString testSetStatusNullAllowed testPaidToggleTwice testVeryLargeNumbers testBookingDateNotChangedByStatusChange testToStringSafeWithEmptyFields

39	LoggerTest	testLog testError testInfo testWarning testDebug testSuccess testCritical testSilent testSetLogToFile testSetLogToConsole testSetDevelopmentMode testEnableProductionMode testEnableDevelopmentMode testEnableSilentMode testDisableAllLogging testIsConsoleLoggingEnabled testIsFileLoggingEnabled testIsDevelopmentMode testClearLogFile testGetLogFileSize testBackupLogFile testLogSystemInfo testLogWithLevel testLogNullMessage testLogEmptyMessage testToggleModesRapidly testMultipleSetLogToConsoleAndFile testClearEmptyLogFile testBackupMultipleTimes testLogAfterDisablingAll testLogSystemInfoMultiple
40	NotificationSenderTest	testSendWelcomeEmail testSendBookingConfirmation testSendPaymentConfirmation testSendSMS testSendWelcomeEmail_NullOrEmpty testSendBookingConfirmation_NullOrEmpty testSendPaymentConfirmation_NullOrEmpty testSendSMS_NullOrEmpty testSendMultipleRapidly
41	PackageCategoryTest	testValues testValueOf testGetDisplayName testGetDescription testValueOfInvalidEnumName testValueOfNull testValueOfEmptyString testGetDisplayNameNotNull testGetDescriptionNotNull
42	PackageReviewTest	testMarkAsHelpful testVerifyReview testUnverifyReview testGetReviewSummary testGetRatingStars testIsPositiveReview testIsNegativeReview testIsNeutralReview

		testGetReviewCategory testGetReviewLength testUpdateReview testAddProsCons testGetAndSetReviewId testGetAndSetUserId testGetAndSetUserName testGetAndSetPackageId testGetAndSetPackageName testGetAndSetReviewText testGetAndSetRating testGetAndSetReviewDate testIsAndSetVerified testGetAndSetHelpfulCount testGetAndSetReviewTitle testIsAndSetRecommended testGetAndSetTravelDate testGetAndSetTravelType testGetAndSetPros testGetAndSetCons testIsAndSetAnonymous testToStringMethod testEqualsMethod testHashCodeMethod testMarkAsHelpfulMultipleTimes testUpdateReviewWithEmptyText testUpdateReviewWithNegativeRating testSetRatingZero testSetRatingAboveFive testEqualsWithNull testEqualsWithDifferentClass testEqualsWithSameObject testHashCodeDifferentIds testToStringWithNullValues testEmptyUserNameAndPackageName
43	PackageUITest	testBrowseTourPackages testSearchPackages testViewPackageDetails testCreateNewPackage testEditPackage testDisplayPackageManagement
44	PDFExporterTest	testExportDailyReport testExportMonthlyReport testExportAnnualReport testExportReport testExportUserList testExportBookingList testExportPaymentList testExportTicket testExportCustomReport testExportBulkData
45	PriceCalculatorTest	testcalculateFinalPrice testZeroPeople testNegativePeople testUnknownUserType

		testLargeGroupDiscount testNullTourPackage testSinglePersonNoGroupDiscount
46	RefundManagerTest	testCreateRefundRequest testRemoveRefundRequest testGetAllRefundRequests testGetPendingRefundRequests testGetApprovedRefundRequests testGetRejectedRefundRequests testGetRefundRequestsByUser testGetRefundRequestsByBooking testGetTotalRefundAmount testGetTotalRefundRequests testGetRefundApprovalRate testGetRefundRequestsByStatus testSearchRefundRequests testGetRefundRequestById_UnknownReturnsNull testRemoveRefundRequest_NonExisting_NoThrow testRemoveRefundRequest_Twice_NoThrow testGetApprovedRefundRequests_EmptyInitially testGetRejectedRefundRequests_EmptyInitially testGetRefundRequestsByUser_UnknownEmpty testGetRefundRequestsByBooking_UnknownEmpty testGetTotalRefundAmount_NoRequestsZero testGetRefundApprovalRate_NoRequestsZero testSearchRefundRequests_NoMatch_Empty testCreateRefundRequest_ZeroAmountAndEmptyReason testCreateRefundRequest_LongReason testGetAllRefundRequests_MultipleAdds
47	TourGuideTest	testAddLanguage testUpdateRating testAssignAndCompleteTour testGettersAndSetters testGetAssignedToursInitiallyEmpty testToStringOutput

		testAddLanguageCaseSensitivity testUpdateRatingZeroAndFive testUpdateRatingNegativeAndAboveMax testAssignTourDuplicateAndCompleteUnassigned testCompleteTourEmptyList testSetLanguagesNullList testSetDailyRateNegative testAssignAndCompleteMultipleTours
48	TouristMenuTest	testTouristMenuConstructorExists testGetLoyaltyTierBronze testGetLoyaltyTierSilver testGetLoyaltyTierGold testGetLoyaltyTierPlatinum testGetNextTierPointsBronzeToSilver testGetNextTierPointsSilverToGold testGetNextTierPointsGoldToPlatinum testGetNextTierPointsPlatinumMax testPrivateMethodsExist testFieldsExist testMethodParameterTypes testLoyaltyTierEdgeCases testNextTierPointsEdgeCases
49	VehicleTest	testVehicleInitialization testSetAndGetFields testAddFeature testRemoveFeature testAddFeatureNullOrEmpty testAddDuplicateFeature testRemoveFeatureNotPresent testSetSeatsAvailableNegative testSetFuelCapacityNegative testOperationalToggle testMultipleFeatures testSetFieldsNullOrEmpty

Serial	Testing Class	Testing Methods
50	BookingUITest	testCreateBookingNormalFlow, testBookingCancelledByUser, testInvalidPackageChoice

51	CurrencyFormatterTest	testFormatBDT, testFormatUSD, testFormatCurrency, testConvertUSDToBDT, testConvertBDTToUSD, testFormatZeroAmount, testFormatNegativeAmount, testFormatLargeAmount, testFormatHighPrecision, testFormatCurrencyNullCode, testFormatCurrencyUnsupportedCode, testConvertZeroAmounts, testConvertNegativeAmounts
52	GroupTourTest	testAddBooking_normalAndCornerCases, testRemoveBooking_normalAndCornerCases, testParticipantNamesAndSpecialArrangements, testGenerateGroupReport
53	GuideAssignmentTest	testConfirm, testStart, testComplete, testCancel, testReassign, testIsActive, testIsCompleted, testIsCancelled, testCalculateTotalEarnings, testGetDurationInHours, testHasTimeConflictWith, testGenerateAssignmentDetails, testGettersAndSetters, testToString
54	GuideScheduleTest	testIsAvailableForDate_InitiallyTrue, testIsAvailableForDate_MaxToursReached_False, testIsAvailableForTimeSlot_NoConflict_True, testIsAvailableForTimeSlot_Conflict_False, testMarkUnavailableAndAvailable, testGetTotalAssignmentsForMonth, testGettersAndSetters, testToStringContainsGuideName, testAddingMoreThanMaxToursPerDay, testUtilizationRateCalculation
55	HotelReviewTest	testInitialValues, testSetDetailedRatings, testMarkHelpful, testVerifyUnverify, testAnonymousReview, testReviewCategories, testStayDetailsAndDuration, testProsCons, testTopLowestRatedAspect,

		testUpdateReviewResetsVerification, testIsDetailedReview, testRatingStars, testOverallSatisfactionScore, testEqualsAndHashCode, testToStringContainsImportantInfo, testEdgeCasesExtremeRatings, testNullComments
56	ItineraryTest	testAddActivity, testAddLocation, testGetAndSetItineraryId, testGetAndSetPackageId, testGetAndSetDay, testGetAndSetActivities, testGetAndSetLocationsVisited, testGetAndSetAccommodationDetails, testGetAndSetMealPlan, testGetAndSetTransportDetails, testGetAndSetStartTime, testGetAndSetEndTime, testGetAndSetSpecialInstructions, testAddNullActivityAndLocation, testAddEmptyActivityAndLocation, testSetNullActivitiesAndLocations, testSetEmptyActivitiesAndLocations, testDuplicateActivitiesAndLocations
57	PaymentTest	testProcessPayment, testFailPayment, testRefundPayment, testGettersAndSetters, testToString, testNullAndEmptyStrings, testNegativeAndZeroAmount, testNegativeAndZeroExchangeRate, testInvalidPaymentStatus, testProcessPaymentMultipleTimes, testSetInvalidDateStrings
58	PaymentUITest	testCardPaymentSuccess, testInvalidCardPayment, testMobileBankingPaymentSuccess, testBankTransferPayment, testCashPayment, testPaymentCancelled

59	ReviewManagerTest	testAddPackageReviewNormal, testAddPackageReviewCornerCases, testGetPackageReviewsByUserAndPackage, testGetAveragePackageRating, testRemoveAllReviewsForPackage, testAddHotelReviewNormal, testHotelReviewsCornerCases, testGetHotelReviewsByUserAndHotel, testGetAverageHotelRating, testRemoveAllReviewsForHotel, testRatingsAddAndGet, testRemoveAllReviewsByUser, testSearchPackageReviews, testGetPackageRatingDistribution
60	SearchHelperTest	testFilterNormal, testFilterEmptyList, testFilterNoMatches, testFilterListWithNullElements, testContainsIgnoreCase, testSearchInListNormal, testSearchInListNullOrEmptyKeyword, testSearchInListWithNullElements, testMatchesAnyNormal, testMatchesAnyNullText, testMatchesAnyWithNullKeywordElement, testMatchesAnyEmptyKeyword
61	TransportTypeTest	testValues, testValueOf, testGetDisplayName, testGetBaseFare, testValueOfInvalidName, testValueOfLowercaseName, testValueOfNull, testDisplayNameNotNullOrEmpty, testBaseFareNonNegative, testEnumConsistency
62	UserManagerTest	testAddUser, testRemoveUser, testDuplicateUserAdd, testUpdateUser, testGetUserByIdAndEmail, testBlockUnblockUser, testGetActiveInactiveUsers, testSearchUsers, testGetUsersByRole, testEmailExistsAndCounts, testNullOrEmptyInputs,

		testUpdateNonExistentUser, testMultipleUserTypes, testSearchCaseInsensitive

Purpose of Testing

The purpose of testing in the Travel & Tourism Management System project is to ensure that all modules and features operate correctly, reliably, and as expected. Testing verifies that critical operations—such as guest booking, package reservation, payment processing, review management, and security logging—are executed without errors. Using JUnit testing, issues are identified and fixed at an early stage, boundary conditions are validated, and overall system quality is enhanced. Moreover, testing ensures that different components of the system interact seamlessly and deliver consistent results.

Ultimately, testing guarantees a smooth, efficient, and error-free experience for both tourism staff and guests, ensuring reliability and trust in the system's performance.

Objective

1. To develop a complete system that manages tourism operations, including guest bookings, package reservations, payments, reviews, and security logging.
2. To automate and streamline travel and tourism management processes in order to minimize manual work and reduce human errors.
3. To design the system using object-oriented programming principles for better organization, scalability, and reusability.
4. To ensure system reliability and correctness by applying JUnit testing for all classes and major functionalities.
5. To provide an easy-to-use interface for tourism staff to manage bookings, packages, payments, and guest records effectively.

6. To simulate real-world tourism workflows in a controlled and testable software environment.
7. To enhance the understanding of the software development lifecycle—from planning to coding, testing, and debugging—within the context of tourism management.

Class wise Testing Part:

AdminMenuTest.java

Function name	Test Case ID	Input	Expected output	Actual output
displayAdminMenu	1	instance = null; call displayAdminMenu()	NullPointerException caught means test pass	Exception caught; Pass
displayAdminMenu	2	instance = null; call displayAdminMenu()	Any Exception should pass test	Whatever exception; Pass
Constructor (existence)	3	Reflection: AdminMenu.class.getConstructors()	At least 1 constructor should exist	length > 0
Constructor (param count=2)	4	Reflection loop over constructors	Some 2-parameter constructor should exist	Found; true
printAdminHeader	5	Reflection getDeclaredMethod("printAdminHeader")	Method should exist	Found; true
printAdminMenuOptions	6	Reflection getDeclaredMethod("printAdminMenuOptions")	Method should exist	Found; true
handleAdminMenuChoice	7	Reflection getDeclaredMethod("handleAdminMenuChoice", int.class)	Method should exist	Found; true
handleUserManagement	8	Reflection getDeclaredMethod("handleUserManagement")	Method should exist	Found; true
viewAllUsers	9	Reflection getDeclaredMethod("viewAllUsers")	Method should exist	Found; true

searchUser	10	Reflection getDeclaredMethod("searchUser")	Method should exist	Found; true
blockUnblockUser	11	Reflection getDeclaredMethod("blockUnblockUser")	Method should exist	Found; true
addNewAdmin	12	Reflection getDeclaredMethod("addNewAdmin")	Method should exist	Found; true
handlePackageManagement	13	Reflection getDeclaredMethod("handlePackageManagement")	Method should exist	Found; true
handleBookingManagement	14	Reflection getDeclaredMethod("handleBookingManagement")	Method should exist	Found; true
handlePaymentManagement	15	Reflection getDeclaredMethod("handlePaymentManagement")	Method should exist	Found; true
generateDailyReport	16	Reflection getDeclaredMethod("generateDailyReport")	Method should exist	Found; true
generateMonthlyReport	17	Reflection getDeclaredMethod("generateMonthlyReport")	Method should exist	Found; true
handleSystemBackup	18	Reflection getDeclaredMethod("handleSystemBackup")	Method should exist	Found; true
logout	19	Reflection getDeclaredMethod("logout")	Method should exist	Found; true
clearScreen	20	Reflection getDeclaredMethod("clearScreen")	Method should exist	Found; true
Class is public	21	Modifier.isPublic(AdminMenu.class.getModifiers())	true	true
Class package	22	AdminMenu.class.getPackage().getName()	"ui"	"ui"
Class name	23	AdminMenu.class.getName()	"ui.AdminMenu"	"ui.AdminMenu"

Has fields	24	AdminMenu.class.getDeclaredFields().length	> 0	> 0
Not abstract	25	Modifier.isAbstract(AdminMenu.class.getModifiers())	false	false
displayAdminMenu return type	26	Reflection getDeclaredMethod("displayAdminMenu").getReturnType()	void.class	void.class
displayAdminMenu is public	27	Modifier.isPublic(method.getModifiers())	true	true
Constructor is public	28	Loop through constructors, check isPublic	At least one public constructor exists	Found; true
Private methods exist	29	Loop through declared methods, count private ones	privateCount > 0	> 0

AdminTest.java

Function name	Test Case ID	Input	Expected output	Actual output
getRole	1	Admin("10","ADMIN","email","112233","017...")	"ADMIN"	"ADMIN"
getAdminLevel (default)	2	Admin("10","Tanvir",...)	"SUPER_ADMIN"	"SUPER_ADMIN"
setAdminLevel (normal)	3	setAdminLevel("Tanvir")	"Tanvir"	"Tanvir"
getPermissions	4	setPermissions(["tanvir@gmail.com","112233"])	same list	same list
setPermissions (normal)	5	setPermissions(["10"])	same list	same list
Constructor with nulls	6	Admin(null,null,null,null,null)	object not null; role "ADMIN"; level "SUPER_ADMIN"	same
setAdminLevel(null)	7	setAdminLevel(null)	level null	null

setAdminLevel("")	8	setAdminLevel("")	""	""
setPermissions(null)	9	setPermissions(null)	null	null
setPermissions(empty)	10	setPermissions([])	empty list	empty; isEmpty=true
setPermissions(with null element)	11	["READ", null, "WRITE"]	preserved including null	preserved
getRole consistency	12	call twice	equal	equal
Admin level multiple changes	13	BASIC→MANAGER→FINAL_LEVEL	"FINAL_LEVEL"	"FINAL_LEVEL"
Special chars in level	14	"ADMIN@#$%"	preserved	preserved
Unicode support	15	Bangla id/name; set level "SUPER_ADMIN"	preserved	preserved
setAdminLevel() exception	16	call no-arg setAdminLevel()	throws UnsupportedOperationException	thrown
Exception message	17	try setAdminLevel()	"Not supported yet."	matched
Permissions duplicates	18	["READ","WRITE","READ","DELETE"]	preserved; size 4	preserved; 4
Large permissions	19	100 items PERM_0..PERM_99	size 100; first/last match	ok
State after exception	20	remember level; call setAdminLevel() throws	level unchanged; role "ADMIN"	unchanged; "ADMIN"
Multiple instances 1	21	admin1 set "LEVEL1"	admin1 level "LEVEL1"	"LEVEL1"
Multiple instances 2	22	admin2 set "LEVEL2"	admin2 level "LEVEL2"	"LEVEL2"

BackupManagerTest.java

Function name	Test case ID	Input	Expected output	Actual output
createBackup	1	pre: data/test.txt exists	"backups" dir exists	exists
restoreBackup	2	createBackup(); pick first backup name; delete "data"; restore	"data" dir exists	exists

DateUtilTest

Function name	Test case ID	Input	Expected output	Actual output
getCurrentDate	1	—	not null; length 10; contains "-"	ok
getCurrentDateTime	2	—	not null; length 19; contains space ":"	ok
formatDate	3	2024-02-15	"2024-02-15"	match
formatDate	4	2023-12-31	"2023-12-31"	match
parseDate	5	"2024-01-15"	LocalDate(2024,1,15)	match
parseDateTime	6	"2024-01-15 14:30:45"	year 2024, month 1, day 15, 14:30:45	match
daysBetween	7	"2024-01-01","2024-01-11"	10	10
daysBetween (same)	8	same,same	0	0
daysBetween (reverse)	9	"2024-01-11","2024-01-01"	-10	-10
isValidDate (valid set)	10	"2024-01-15","2023-12-31","2000-02-29"	true	true
isValidDate (invalid set)	11	"15-01-2024","2024/01/15","2024-13-01","2024-01-32","","null"	false	false
isValidDateTime (valid)	12	"2024-01-15 14:30:45","2023-12-31 23:59:59","2000-01-01 00:00:00"	true	true
isValidDateTime (invalid)	13	"2024-01-15","15-01-2024 14:30:45","2024-01-15 25:30:45","","null"	false	false
formatDateTime	14	2024-01-15T14:30:45	"2024-01-15 14:30:45"	match
addDays	15	"2024-01-15", 10	"2024-01-25"	match
addDays (0)	16	"2024-01-15", 0	same date	same
addDays (negative)	17	"2024-01-15", -5	"2024-01-10"	match
subtractDays	18	"2024-01-15", 10	"2024-01-05"	match
subtractDays (0)	19	"2024-01-15", 0	same	same
isDateBefore	20	"2024-01-15","2024-01-20"	true	true
isDateBefore (not before)	21	"2024-01-20","2024-01-15"	false	false

isDateBefore (equal)	22	same,same	false	false
isDateAfter	23	earlier,later	false	false
isDateAfter	24	later,earlier	true	true
isDateAfter (equal)	25	same,same	false	false
isDateEqual	26	same,same	true	true
isDateEqual	27	diff	false	false
getFirstDayOfMonth	28	"2024-01-15"	"2024-01-01"	match
getFirstDayOfMonth	29	"2024-02-29"	"2024-02-01"	match
getLastDayOfMonth	30	"2024-01-15"	"2024-01-31"	match
getLastDayOfMonth (leap Feb)	31	"2024-02-15"	"2024-02-29"	match
getLastDayOfMonth (non-leap Feb)	32	"2023-02-15"	"2023-02-28"	match
getDayOfWeek	33	"2024-01-01"	1	1
getDayOfWeek	34	"2024-01-06"	6	6
getDayOfWeek	35	"2024-01-07"	7	7
getDayName	36	"2024-01-01"	"MONDAY"	match
getDayName	37	"2024-01-06"	"SATURDAY"	match
getMonthName	38	"2024-01-15"	"JANUARY"	match
getMonthName	39	"2024-12-25"	"DECEMBER"	match
getYear	40	"2024-01-15"	2024	2024
getYear	41	"2023-12-31"	2023	2023
getMonth	42	"2024-01-15"	1	1
getMonth	43	"2024-12-25"	12	12
getDay	44	"2024-01-15"	15	15
getDay	45	"2024-01-01"	1	1
getDay	46	"2024-01-31"	31	31
isWeekend	47	Monday date	false	false
isWeekend	48	Saturday	true	true
isWeekend	49	Sunday	true	true
isWeekend	50	Friday	false	false
isLeapYear	51	"2024-01-01"	true	true
isLeapYear	52	"2023-01-01"	false	false

isLeapYear	53	"2000-01-01"	true	true
isLeapYear	54	"1900-01-01"	false	false
hoursBetween	55	"2024-01-01 10:00:00"→"15:00:00"	5	5
hoursBetween (same)	56	same	0	0
hoursBetween (overnight)	57	"2024-01-01 10:00:00"→"2024-01-02 10:00:00"	24	24
minutesBetween	58	"10:00:00"→"10:30:00"	30	30
minutesBetween	59	"10:00:00"→"11:00:00"	60	60
addHours	60	"2024-01-01 10:00:00", 5	"2024-01-01 15:00:00"	match
addHours (overnight)	61	+24	"2024-01-02 10:00:00"	match
addMinutes	62	+30	"2024-01-01 10:30:00"	match
addMinutes (90)	63	+90	"2024-01-01 11:30:00"	match
getCurrentTimestamp	64	—	numeric string; > 1577836800000	ok
formatTimestamp	65	1704110400000L	length 19; contains ":"	ok
parseTimestamp	66	"2024-01-01 12:00:00"	>0; round-trip equals	ok

FileHandlerTest

Function name	Test case ID	Input	Expected output	Actual output
writeToFile	1	("test.txt","Test data line")	file exists; 1 line; content match	ok
writeToFile (null filename)	2	(null,"Test data")	Graceful (no crash)	ok
writeToFile (empty filename)	3	("","Test data")	Graceful	ok
writeToFile (null data)	4	("test.txt", null)	file exists; line "null"	ok
writeToFile ("")	5	("test.txt","")	file exists; 1 empty line	ok
writeToFile (long data)	6	very long string	1 line; content match	ok

writeToFile (special chars)	7	specials	1 line; content match	ok
writeToFile multiple (append)	8	3 writes	3 lines "Line 1..3"	ok
readFromFile	9	"test.txt" with 2 lines	list size 2; lines match	ok
readFromFile (non-existent)	10	"nonexistent.txt"	empty list, not null	ok
readFromFile (null filename)	11	null	empty list	ok
readFromFile (empty filename)	12	""	empty list	ok
readFromFile (empty file)	13	"empty.txt" cleared	empty list	ok
clearFile	14	clear "test.txt" after writes	file exists; empty	ok
clearFile (non-existent)	15	"nonexistent.txt"	create empty file	ok
clearFile (null)	16	null	graceful	ok
clearFile ("")	17	""	graceful	ok
fileExists	18	"test.txt" before/after write	false→true	ok
fileExists (non-existent)	19	"definitely_does_not_exist.txt"	false	false
special filename	20	"special@#$.txt"	exists; data match	ok
very long filename	21	"long...long.txt"	if created then read 1 line; cleanup	ok
concurrent writes	22	100 writes "Line i"	100 lines; each matches	ok
write after clear	23	write→clear→write	1 line "New data"	ok
write with newline	24	"Line with\nnewline character"	result size ≥1; content fragment exists	ok
write with tabs	25	"Data\twith\ttab\tcharacters"	1 line; match	ok

HotelManagerTest.java

Function name	Test case ID	Input	Expected output	Actual output
addHotel	1	addHotel(null)	no exception; pass	pass
removeHotel	2	""	no exception; pass	pass
removeHotel	3	null	no exception; pass	pass
getHotelById	4	""	null	null
getHotelById	5	null	null	null
updateHotel	6	null	no exception; pass	pass
updateHotel	7	null (try-catch)	any behavior; pass	pass
getAllHotels	8	—	not null list	not null
getAllHotels type	9	—	instance of List	true
getActiveHotels	10	—	not null list	not null
getActiveHotels type	11	—	instance of List	true
searchHotels	12	""	not null list	not null
searchHotels	13	null	not null or handled; pass	pass
searchHotels	14	"hotel"	not null list	not null
getHotelsByLocation	15	""	not null list	not null
getHotelsByLocation	16	null	handled; pass	pass
getHotelsByLocation	17	"Dhaka"	not null list	not null
getHotelsByRating	18	0.0	not null list	not null
getHotelsByRating	19	4.5	not null list	not null
getHotelsByRating	20	-1.0	not null list	not null
getHotelsWithAvailableRooms	21	—	not null list	not null

getHotelsWithAvailableRooms type	22	—	instance of List	true
searchAvailableRooms	23	"", null	not null list	not null
searchAvailableRooms	24	null, null	handled; pass	pass
bookRoom	25	"","R001"	false	false
bookRoom	26	"H001",""	false	false
bookRoom	27	null,null	false	false
bookRoom	28	"NONEXISTENT","R001"	false	false
checkoutRoom	29	"","R001"	false	false
checkoutRoom	30	"H001",""	false	false
checkoutRoom	31	null,null	false	false
checkoutRoom	32	"NONEXISTENT","R001"	false	false
addRoomToHotel	33	"","null room"	no exception	pass
addRoomToHotel	34	null,null	no exception	pass
addRoomToHotel	35	"NONEXISTENT",null	no exception	pass
removeRoomFromHotel	36	"","R001"	no exception	pass
removeRoomFromHotel	37	null,null	no exception	pass
removeRoomFromHotel	38	"NONEXISTENT","R001"	no exception	pass
getTotalHotels	39	—	>=0	>=0
getTotalHotels type	40	—	int	int
getActiveHotelsCount	41	—	>=0	>=0
getActiveHotelsCount type	42	—	int	int
getAverageHotelRating	43	—	>=0.0	>=0.0
getAverageHotelRating type	44	—	double	double
getHighestRatedHotel	45	—	null or Hotel	ok
getHighestRatedHotel type	46	—	if not null then Hotel	ok
getHotelCountByLocation	47	—	not null map	not null
getHotelCountByLocation type	48	—	instance of Map	true
getTotalRooms	49	—	>=0	>=0
getTotalRooms type	50	—	int	int

getAvailableRoomsCount	51	—	>=0	>=0
getAvailableRoomsCount type	52	—	int	int
getOccupancyRate	53	—	>=0.0	>=0.0
getOccupancyRate type	54	—	double	double
getOccupancyRate range	55	—	0.0..100.0	within
searchHotels (special chars)	56	"@#$%^&*()"	not null	not null
getHotelsByLocation (special chars)	57	"Cox's Bazar"	not null	not null
getHotelsByRating (max)	58	5.0	not null	not null
searchAvailableRooms (suite)	59	"Chittagong", SUITE	not null	not null
Multiple ops	60	several getters	no exception; pass	pass
Method consistency	61	getTotalHotels twice	equal	equal
State independence	62	two instances counts	>=0 each	ok

HotelTest.java

Function name	Test case ID	Input	Expected output	Actual output
addRoom	1	instance=null; addRoom(null)	NPE বা ব্যর্থ; টেমপ্লেট fail	fail("prototype")
removeRoom	2	instance=null; ""	NPE/ব্যর্থ; fail	fail
getRoomById	3	instance=null; ""	exp null; then fail	fail
getAvailableRooms	4	instance=null	exp null; then fail	fail
getHotelId	5	instance=null	exp ""; then fail	fail
setHotelId	6	instance=null; ""	fail	fail
getName	7	instance=null	exp ""; then fail	fail
setName	8	instance=null; ""	fail	fail
getLocation	9	instance=null	exp ""; then fail	fail
setLocation	10	instance=null; ""	fail	fail
getRating	11	instance=null	0.0; then fail	fail

setRating	12	instance=null; 0.0	fail	fail
getRoomList	13	instance=null	exp null; then fail	fail
setRoomList	14	instance=null; null	fail	fail
getAddress	15	instance=null	exp ""; then fail	fail
setAddress	16	instance=null; ""	fail	fail
getPhone	17	instance=null	exp ""; then fail	fail
setPhone	18	instance=null; ""	fail	fail
getEmail	19	instance=null	exp ""; then fail	fail
setEmail	20	instance=null; ""	fail	fail
getAmenities	21	instance=null	exp null; then fail	fail
setAmenities	22	instance=null; null	fail	fail
getDescription	23	instance=null	exp ""; then fail	fail
setDescription	24	instance=null; ""	fail	fail
isActive	25	instance=null	exp false; then fail	fail
setActive	26	instance=null; false	fail	fail
toString	27	instance=null	exp ""; then fail	fail

InvoiceTest

Function name	Test case ID	Input	Expected output	Actual output
addItem	1	("Hotel Booking",1000,2)	items size 1; fields match	ok
addItem (second)	2	("Transport",500,1)	items size 2	ok
calculateTotal	3	Items: (1000×2),(500×3); tax=100; disc=200	subtotal 3500; total 3400	ok
generateInvoiceText	4	after add("Test Item",100,1)	contains INVOICE, INV001,	ok

			BK001, "Test Item"	
getInvoiceId	5	—	"INV001"	"INV001"
setInvoiceId	6	"INV002"	updated	updated
getBookingId	7	—	"BK001"	"BK001"
setBookingId	8	"BK002"	updated	updated
getUserId	9	—	"U001"	"U001"
setUserId	10	"U002"	updated	updated
getSubtotal	11	initially	0.0	0.0
getSubtotal (after items)	12	add(100×2, 50×3)	350.0	350.0
setSubtotal	13	500.0	updated	updated
getTax	14	initial	0.0	0.0
setTax	15	50.0	updated	updated
getDiscount	16	initial	0.0	0.0
setDiscount	17	25.0	updated	updated
getTotalAmount	18	add item 100; tax 10; disc 5; calculateTotal	105.0	105.0
setTotalAmount	19	1000.0	updated	updated
getIssueDate	20	—	not null, non-empty	ok
setIssueDate	21	"2024-01-15"	updated	updated
getDueDate	22	initial	null	null
setDueDate	23	"2024-02-15"	updated	updated
getStatus	24	initial	"GENERATED "	"GENERATED "
setStatus	25	"PAID"	updated	updated
getItems	26	initial → add one	empty then 1	ok
constructor	27	"INV999","BK999","U999"	defaults + ids set	ok
InvoiceItem inner	28	("Test Description",99.99,3)	fields set	ok
Complex calculation	29	items 6000,2000,600; tax 860; disc 100; recalc	subtotal 8600; total 9360	ok

LoginUITest

Function name	Test case ID	Input	Expected output	Actual output
displayLoginScreen (exists)	1	Reflection getDeclaredMethod	method not null	found

displayLoginScreen return type	2	Reflection	"user" (simpleName)	"user" asserted
displayLoginScreen access	3	Reflection	public	public
displayLoginScreen params	4	Reflection	0	0
displayRegistrationScreen (exists)	5	Reflection	exists	found
displayRegistrationScreen return	6	Reflection	void	void
displayRegistrationScreen access	7	Reflection	public	public
displayRegistrationScreen params	8	Reflection	0	0
displayForgotPasswordScreen (exists)	9	Reflection	exists	found
displayForgotPasswordScreen return	10	Reflection	void	void
displayForgotPasswordScreen access	11	Reflection	public	public
displayForgotPasswordScreen params	12	Reflection	0	0
displayLoginHelp (exists)	13	Reflection	exists	found
displayLoginHelp return	14	Reflection	void	void
displayLoginHelp access	15	Reflection	public	public
displayLoginHelp params	16	Reflection	0	0
displayRegistrationHelp (exists)	17	Reflection	exists	found
displayRegistrationHelp return	18	Reflection	void	void
displayRegistrationHelp access	19	Reflection	public	public
displayRegistrationHelp params	20	Reflection	0	0

Class exists	21	LoginUI.class	not null	not null
Class package/name	22	class.getName()	"ui.LoginUI"	"ui.LoginUI"
Class access	23	Modifier	public	public
Constructor exists	24	getConstructors	length > 0	true
Constructor param count=2	25	iterate	found any with 2 params	true
Total public method count	26	getMethods	>= 5 declared in LoginUI	true
Private method count	27	getDeclaredMethods	>=0	true
Naming convention	28	names array	starts lowerCase	true
Required methods exist	29	list of 5 methods	found (>=0 guarded)	pass
Not abstract	30	Modifier	false	false
Not final	31	Modifier	false	false

RatingTest

Function name	Test case ID	Input	Expected output	Actual output

markAsHelpful	1	new Rating(...4.5,"PACKAGE");
		markAsHelpful()
			helpfulCount=1	1

verifyRating	2	verifyRating()	isVerified=true	true
unverifyRating	3	verify→unverify	isVerified=false	false
isValidRating	4	value 4.5	true	true
isValidRating	5	value 6.0	false	false
getRatingStars	6	5.0	"★★★★★"	match
getRatingStars	7	3.5	"★★★☆☆"	match
getRatingCategory	8	4.8	"Excellent"	match
getRatingCategory	9	1.5	"Poor"	match
isPositiveRating	10	4.2	true	true
isPositiveRating	11	3.5	false	false
isNegativeRating	12	2.0	true	true

isNegativeRating	13	3.5	false	false
isNeutralRating	14	3.0	true	true
isNeutralRating	15	4.5	false	false
getRatingDescription	16	5.0	"Excellent - Extremely satisfied"	match
getRatingDescription	17	2.0	"Poor - Unsatisfied"	match
getRatingSummary	18	setUserName("John Doe")	contains "John Doe", "4.5"	true
updateRating (valid)	19	updateRating(4.5,"Updated")	value 4.5; comments Updated	ok
updateRating (invalid)	20	updateRating(6.0,...)	unchanged 3.0	ok
getRatingLevel	21	4.3	4	4
getRatingPercentage	22	4.0	80.0	80.0
isHighRating	23	4.5	true	true
isHighRating	24	3.0	false	false
isLowRating	25	1.5	true	true
isLowRating	26	4.0	false	false
hasComments	27	with "Great service!"	true	true
hasComments	28	no comments	false	false
getCommentsLength	29	"Great service!"	14	14
getCommentsLength	30	no comments	0	0
isDetailedRating	31	long comment > 50 chars	true	true
isDetailedRating	32	"Good"	false	false
calculateAverageRating	33		4.0	4.0
calculateAverageRating	34	[]	0.0	0.0
getRatingDistribution	35		4→2,5→1,1→0	match
formatRatingDistribution	36		not null; contains "Rating Distribution"	ok
getRatingId	37	"R001"	"R001"	"R001"
setRatingId	38	"R002"	updated	updated
getUserId	39	"U001"	"U001"	"U001"

setUserId	40	"U002"	updated	updated
getUserName	41	set "John Doe"	"John Doe"	match
setUserName	42	"Jane Smith"	updated	updated
getEntityId	43	"P001"	"P001"	"P001"
setEntityId	44	"P002"	updated	updated
getEntityType	45	"PACKAGE"	"PACKAGE"	match
setEntityType	46	"HOTEL"	"HOTEL"	match
getRatingValue	47	4.5	4.5	4.5
setRatingValue (valid)	48	4.5	4.5	4.5
setRatingValue (invalid)	49	6.0	unchanged	unchanged
getComments	50	"Great experience!"	same	same
setComments	51	"Excellent service"	updated	updated
getRatingDate	52	—	not null	not null
setRatingDate	53	"2023-12-01"	updated	updated
isVerified	54	default	false	false
setVerified	55	true	true	true
getCriteria	56	default	"Overall"	"Overall"
setCriteria	57	"Service"	updated	updated
getHelpfulCount	58	default	0	0
setHelpfulCount	59	5	5	5
isAnonymous	60	default	false	false
setAnonymous	61	true	true	true
getAspectRated	62	set "Location"	"Location"	match
setAspectRated	63	"Quality"	updated	updated
toString	64	—	contains "R001","P001"	contains
equals (same id)	65	id "R001" vs same id	true	true
equals (diff id)	66	"R001" vs "R002"	false	false
hashCode	67	id "R001"	"R001".hashCode	match
Minimum valid rating	68	1.0	valid; "Poor"; "★☆☆☆☆"	match
Maximum valid rating	69	5.0	valid; "Excellent"; "★★★★★"	match
Boundary 2.0	70	2.0	negative=true	true
Boundary 4.0	71	4.0	positive=true	true

Null comments	72	null	hasComments=false; len=0; not detailed	match
Empty comments	73	""	hasComments=false; len=0; not detailed	match
Whitespace comments	74	" "	hasComments=false; len=3; not detailed	match
Average null list	75	null	0.0	0.0
Single rating avg	76	[3.7]	3.7	3.7
Identical ratings avg	77		4.0	4.0
Distribution empty	78	[]	all zeros	match
Distribution all 5-star	79		5→2; others 0	match
Very long rating id	80	longId string	preserved	preserved
Special chars entity id	81	"PKG@...?"	preserved	preserved
Unicode user name	82	"জন ডো 李明 Müller"	preserved	preserved
Multiple helpful	83	call thrice	3	3
Verify-unverify multiple	84	sequence	final true	true
Update rating multiple	85	4.0→4.5	final 4.5; comments updated	ok
Percentage extremes	86	1.0→20.0%; 5.0→100%	match	match
equals null	87	compare with null	false	false
equals same instance	88	self	true	true
equals different class	89	String	false	false
Constructor (all params)	90	("U001","P001",4.5,"PACKAGE","Great","Service")	fields set; verified=false; helpful=0	ok
Anonymous summary	91	setAnonymous(true)	contains "Anonymous"; no "null"	ok
Summary with aspect	92	setAspectRated("Location")	contains "Aspect Rated: Location"	ok

Format distribution single	93	[4.0]	"4 ★: 1 ratings (100.0%)"	present
Set value boundaries	94	1.0→5.0→0.9→5.1	keep 1.0 then 5.0; invalid ignored	ok

ReportGeneratorTest

Function name	Test case ID	Input	Expected output	Actual output
generateDaily	1	—	not null; contains "DAILY REPORT"	ok
generateDaily (content)	2	—	contains "Date:"; length > 100	ok
generateDaily (multiple)	3	twice	both not null	ok
generateDaily (perf)	4	measure time	< 5000 ms	ok
generateDailyReport	5	"2024-01-15"	contains date	ok
generateDailyReport	6	"2024-06-15"	contains "DAILY BOOKINGS","DAILY PAYMENTS"	ok
generateDailyReport	7	"2024-02-29"	contains date; "NEW REGISTRATIONS"	ok
generateDailyReport	8	"2024-12-31"	contains date	ok
generateDailyReport	9	null	not null; contains "DAILY REPORT"	ok
generateDailyReport	10	""	not null; contains "DAILY REPORT"	ok
generateMonthly	11	—	not null; contains "MONTHLY REPORT"	ok
generateMonthly (content)	12	—	contains "Month:","MONTHLY OVERVIEW"	ok
generateMonthly (length)	13	—	length > 200; contains "Generated:"	ok
generateMonthly (consistency)	14	twice	both contain "MONTHLY"	ok
generateMonthlyReport	15	"2024-01"	contains month	ok
generateMonthlyReport	16	"2024-06"	contains "BOOKING STATUS	ok

			BREAKDOWN","POPULAR PACKAGES"	
generateMonthlyReport	17	"2024-12"	contains "CUSTOMER ANALYSIS","PAYMENT METHODS"	ok
generateMonthlyReport	18	"2023-03"	not null; contains month	ok
generateMonthlyReport	19	null	not null; contains "MONTHLY REPORT"	ok
generateMonthlyReport	20	""	not null; contains "MONTHLY REPORT"	ok
generateAnnual	21	—	not null; contains "ANNUAL REPORT"	ok
generateAnnual (content)	22	—	contains "Year:","ANNUAL OVERVIEW"	ok
generateAnnual (sections)	23	—	contains "FINANCIAL SUMMARY"; length > 300	ok
generateAnnual (comprehensive)	24	—	contains "GROWTH ANALYSIS","Generated:"	ok
generateAnnualReport	25	"2024"	contains year	ok
generateAnnualReport	26	"2023"	contains "TOP PERFORMING PACKAGES","CUSTOMER INSIGHTS"	ok
generateAnnualReport	27	"2025"	contains "SEASONAL ANALYSIS","SYSTEM STATISTICS"	ok
generateAnnualReport	28	"2022"	not null; contains year	ok
generateAnnualReport	29	null	not null; contains "ANNUAL REPORT"	ok
generateAnnualReport	30	""	not null; contains "ANNUAL REPORT"	ok
generateSystemReport	31	—	not null; contains "SYSTEM REPORT"	ok
generateSystemReport (content)	32	—	contains "SYSTEM OVERVIEW","USER DISTRIBUTION"	ok
generateSystemReport (sections)	33	—	contains "PACKAGE DISTRIBUTION","SYSTEM HEALTH"	ok
generateSystemReport (length)	34	—	contains "Generated:"; length > 200	ok
generateSystemReport (consistency)	35	twice	both not null	ok

generateCustomReport	36	"REVENUE","2024-01-01","2024-01-31"	not null; contains "CUSTOM REPORT"	ok
generateCustomReport (REVENUE)	37	same	contains type and start date	ok
generateCustomReport (BOOKINGS)	38	Feb range	contains "BOOKINGS" and end date	ok
generateCustomReport (CUSTOMERS)	39	Mar range	contains "CUSTOMERS","Period:"	ok
generateCustomReport (PACKAGES)	40	Apr range	not null; contains "PACKAGES"	ok
generateCustomReport (UNKNOWN)	41	"UNKNOWN"	not null; contains "Unknown report type"	ok
generateCustomReport (June)	42	Jun range	contains both dates	ok
generateCustomReport (long range)	43	2024 full	not null; contains "2024-01-01"	ok
generateCustomReport (null params)	44	null,null,null	not null; contains "CUSTOM REPORT"	ok
generateCustomReport (empty params)	45	"","",""	not null; contains "CUSTOM REPORT"	ok
Constructor	46	new ReportGenerator()	instance not null; generateDaily not null	ok
Constructor immediate	47	new ReportGenerator(); generateSystemReport	contains "SYSTEM REPORT"; length > 100	ok
All types consistency	48	daily & monthly	both not null	ok
All types content	49	annual & system	contain "ANNUAL" and "SYSTEM"	ok
Performance monthly	50	time	< 5000 ms	ok
Multiple system reports	51	two runs	both contain header and length >100	ok
Boundary dates	52	2024-02-29, 2024-12-31	not null	ok
Edge case dates	53	2024-01-01; custom same day	contains date; contains same period	ok
Complete integration	54	generate daily+monthly	both not null	ok
Final comprehensive	55	annual & system	content length checks	ok

TransportManagerTest

Function name	Test case ID	Input	Expected output	Actual output
addTransport	1	—	instance not null	ok
removeTransport	2	—	instance not null	ok
getTransportById	3	""	result either null or non-null (guard)	guard passed
updateTransport	4	—	instance not null	ok
getAllTransports	5	—	not null list	ok
getActiveTransports	6	—	not null list	ok
searchTransports	7	"",""	not null list	ok
getTransportsByType	8	BUS	not null list	ok
getAvailableTransports	9	—	not null list	ok
getTransportsByRoute	10	"",""	not null list	ok
bookSeat	11	""	boolean (true/false)	guard passed
cancelSeat	12	""	boolean (true/false)	guard passed
addVehicle	13	—	instance not null	ok
removeVehicle	14	—	instance not null	ok
getVehicleById	15	""	null or non-null	guard passed
getAllVehicles	16	—	not null list	ok
getOperationalVehicles	17	—	not null list	ok
getVehiclesByType	18	BUS	not null list	ok
getSeatsForTransport	19	""	not null list	ok
getAvailableSeats	20	""	not null list	ok

bookSpecificSeat	21	"","","",""	boolean	guard passed
cancelSpecificSeat	22	"",""	boolean	guard passed
getTotalTransports	23	—	>=0	ok
getActiveTransportsCount	24	—	>=0	ok
getTotalVehicles	25	—	>=0	ok
getOperationalVehiclesCount	26	—	>=0	ok
getTransportCountByType	27	—	not null map	ok
getAverageOccupancyRate	28	—	any double	guard passed
getFullyBookedTransports	29	—	not null list	ok

ValidatorTest

Function name	Test case ID	Input	Expected output	Actual output
validateEmail (valid)	1	"test@example.com"	true	true
validateEmail (valid)	2	"user123@gmail.com"	true	true
validateEmail (valid)	3	"a@b.co"	true	true
validateEmail (valid)	4	"test.email@domain.org"	true	true
validateEmail (valid)	5	"user+tag@example.net"	true	true
validateEmail (valid)	6	"user_name@example-domain.com"	true	true
validateEmail (invalid)	7	null	false	false
validateEmail (invalid)	8	""	false	false
validateEmail (invalid)	9	"plainaddress"	false	false
validateEmail (invalid)	10	"@example.com"	false	false
validateEmail (invalid)	11	"user@"	false	false
validateEmail (invalid)	12	"user@@domain.com"	false	false
validatePhone (valid BD)	13	"01712345678"	true	true
validatePhone (valid BD)	14	"01812345678"	true	true
validatePhone (valid BD)	15	"01912345678"	true	true

validatePhone (valid BD)	16	"01312345678"	true	true
validatePhone (valid BD)	17	"01412345678"	true	true
validatePhone (valid BD)	18	"01512345678"	true	true
validatePhone (valid BD)	19	"01612345678"	true	true
validatePhone (valid intl)	20	"+8801712345678"	true	true
validatePhone (valid intl)	21	"+8801812345678"	true	true
validatePhone (invalid)	22	null	false	false
validatePhone (invalid)	23	""	false	false
validatePhone (invalid)	24	"0171234567"	false	false
validatePhone (invalid)	25	"017123456789"	false	false
validatePhone (invalid)	26	"01012345678"	false	false
validatePhone (invalid)	27	"01212345678"	false	false
validatePhone (invalid)	28	"02712345678"	false	false
validatePhone (invalid)	29	"1712345678"	false	false
validatePassword (valid)	30	"123456"	true	true
validatePassword (valid)	31	"password"	true	true
validatePassword (valid)	32	"abcdef"	true	true
validatePassword (valid)	33	" " (6 spaces)	true	true
validatePassword (valid)	34	"P@ssw0rd123"	true	true
validatePassword (invalid)	35	null	false	false
validatePassword (invalid)	36	""	false	false
validatePassword (invalid)	37	"12345"	false	false
validatePassword (invalid)	38	"abc"	false	false
validatePassword (invalid)	39	" " (5 spaces)	false	false
isNotEmpty (valid)	40	"hello"	true	true
isNotEmpty (valid)	41	"test"	true	true
isNotEmpty (valid)	42	"a"	true	true
isNotEmpty (valid)	43	" hello "	true	true
isNotEmpty (valid)	44	" test "	true	true
isNotEmpty (invalid)	45	null	false	false

isNotEmpty (invalid)	46	""	false	false
isNotEmpty (invalid)	47	" "	false	false
isNotEmpty (invalid)	48	" "	false	false
isNotEmpty (invalid)	49	" "	false	false
isNotEmpty (invalid)	50	"\t"	false	false
isNotEmpty (invalid)	51	"\n"	false	false
isPositiveNumber (valid)	52	0.1	true	true
isPositiveNumber (valid)	53	1.0	true	true
isPositiveNumber (valid)	54	100.0	true	true
isPositiveNumber (valid)	55	0.000001	true	true
isPositiveNumber (valid)	56	999999.999999	true	true
isPositiveNumber (invalid)	57	0.0	false	false
isPositiveNumber (invalid)	58	-0.1	false	false
isPositiveNumber (invalid)	59	-1.0	false	false
isPositiveNumber (invalid)	60	-100.0	false	false

BookingManagerTest.java

Function name	Test case ID	Input	Expected output	Actual output
addBooking	1	null booking	void	void
removeBooking	2	empty string ""	void	void
getBookingById	3	empty string ""	null	null
updateBooking	4	null booking	void	void
getAllBookings	5	empty	List	List
getBookingsByUser	6	empty string ""	List	List
getBookingsByPackage	7	empty string ""	List	List

getBookingsByStatus	8	null status	List	List
getBookingsByDate	9	empty string ""	List	List
getBookingsByTravelDate	10	empty string ""	List	List
getBookingsInDateRange	11	empty strings	List	List
getPendingBookings	12	empty	List	List
getConfirmedBookings	13	empty	List	List
getCancelledBookings	14	empty	List	List
cancelBooking	15	booking ID, reason	false	false
confirmBooking	16	booking ID	false	false
completeBooking	17	booking ID	false	false
getTotalRevenue	18	empty	14000.0	14000.0
getRevenueByPackage	19	empty string ""	0.0	0.0
getTotalBookings	20	empty	1	1
getBookingsCountByStatus	21	null status	0	0
getUpcomingBookings	22	empty	List	List
getTodaysBookings	23	empty	List	List
getBookingCountsByMonth	24	empty	Map	Map
searchBookings	25	empty string ""	List	List
addBooking (corner)	26	null booking	1 total bookings	1 total bookings
removeBooking (corner)	27	"NON_EXISTENT"	null	null
getBookingById (corner)	28	"INVALID_ID"	null	null
getBookingsByUser (corner)	29	"INVALID_USER"	empty list	empty list

getBookingsByPackage (corner)	30	"INVALID_PACKAGE"	empty list	empty list
getBookingsByStatus (corner)	31	null	empty list	empty list
getBookingsByDate (corner)	32	"2099-99-99"	empty list	empty list
getBookingsInDateRange (corner)	33	reversed dates	empty list	empty list
cancelBooking (corner)	34	"INVALID_ID"	false	false
confirmBooking (corner)	35	"INVALID_ID"	false	false
completeBooking (corner)	36	"INVALID_ID"	false	false
getRevenueByPackage (corner)	37	"INVALID_PACKAGE"	0.0	0.0
getBookingsCountByStatus (corner)	38	null	0	0
searchBookings (corner)	39	"NonMatchingKeyword"	empty list	empty list

BookingStatusTest.java

Function name	Test case ID	Input	Expected output	Actual output
values	1	empty	[PENDING,CONFIRMED,CANCELED,COMPLETED]	[PENDING,CONFIRMED,CANCELED,COMPLETED]
valueOf	2	"CONFIRMED"	CONFIRMED	CONFIRMED
getDisplayName	3	CANCELED	"Canceled"	"Canceled"

getDescription	4	COMPLETED	"Tour has been completed"	"Tour has been completed"
valueOf (corner)	5	"INVALID_STATUS"	IllegalArgumentException	IllegalArgumentException
valueOf (corner)	6	null	NullPointerException	NullPointerException
getDisplayName (corner)	7	all values	not null	not null
getDescription (corner)	8	all values	not null	not null

BookingTest.java

Function name	Test case ID	Input	Expected output	Actual output
confirmBooking	1	empty	CONFIRMED	CONFIRMED
cancelBooking	2	empty	CANCELED	CANCELED
completeBooking	3	empty	COMPLETED	COMPLETED
setBookingId	4	"B002"	"B002"	"B002"
setUserId	5	"U002"	"U002"	"U002"
setPackageId	6	"P002"	"P002"	"P002"
getBookingDate	7	empty	not null	not null
setTravelDate	8	"2025-12-20"	"2025-12-20"	"2025-12-20"
setStatus	9	CONFIRMED	CONFIRMED	CONFIRMED
setNumberOfPeople	10	4	4	4
setTotalAmount	11	2000.0	2000.0	2000.0
setSpecialRequests	12	"Veg meals only"	"Veg meals only"	"Veg meals only"
setHotelId	13	"H002"	"H002"	"H002"
setTransportId	14	"T002"	"T002"	"T002"

setGuideId	15	"G002"	"G002"	"G002"
setPaid	16	false	false	false
toString	17	empty	contains "bookingId='B001'"	contains "bookingId='B001'"
confirmBooking (corner)	18	repeat confirm	CONFIRMED	CONFIRMED
cancelAfterConfirm (corner)	19	cancel after confirm	CANCELED	CANCELED
setNumberOfPeople (corner)	20	0	0	0
setNumberOfPeople (corner)	21	-3	-3	-3
setTotalAmount (corner)	22	0.0	0.0	0.0
setTotalAmount (corner)	23	-500.0	-500.0	-500.0
setTravelDate (corner)	24	empty string	empty string	empty string
setIdsEmpty (corner)	25	empty strings	empty strings	empty strings
setStatus (corner)	26	null	null	null
paidToggle (corner)	27	toggle twice	true	true
veryLargeNumbers (corner)	28	1,000,000	1,000,000	1,000,000
bookingDateNotChanged (corner)	29	status changes	same date	same date
toStringSafeEmpty (corner)	30	empty fields	contains "B001"	contains "B001"

LoggerTest.java

Function name	Test case ID	Input	Expected output	Actual output
log	1	"Test log message"	true file logging	true file logging

error	2	"Test error message"	true file logging	true file logging
info	3	"Test info message"	true console logging	true console logging
warning	4	"Test warning message"	true console logging	true console logging
debug	5	"Debugging in development mode"	true development mode	true development mode
success	6	"Success message test"	true console logging	true console logging
critical	7	"Critical error occurred"	true	true
silent	8	"Silent logging test"	true file logging	true file logging
setLogToFile	9	false/true	false/true	false/true
setLogToConsole	10	false/true	false/true	false/true
setDevelopmentMode	11	true/false	true/false	true/false
enableProductionMode	12	empty	false dev, false console, true file	false dev, false console, true file
enableDevelopmentMode	13	empty	true dev, true console, true file	true dev, true console, true file
enableSilentMode	14	empty	false console, true file	false console, true file

disableAllLogging	15	empty	false console, false file	false console, false file
isConsoleLoggingEnabled	16	true	true	true
isFileLoggingEnabled	17	true	true	true
isDevelopmentMode	18	true	true	true
clearLogFile	19	"Clearing log file test"	40 size	40 size
getLogFileSize	20	"Testing log file size"	>= 0	>= 0
backupLogFile	21	empty	>= 0	>= 0
logSystemInfo	22	empty	true file logging	true file logging
logWithLevel	23	"TEST", "test message"	true file logging	true file logging
logNull (corner)	24	null messages	no exception	no exception
logEmpty (corner)	25	empty strings	no exception	no exception
toggleModesRapidly (corner)	26	rapid mode changes	correct final state	correct final state
multipleSetLogToConsoleAndFile (corner)	27	multiple toggles	final true states	final true states
clearEmptyLogFile (corner)	28	clear empty file	>= 0 size	>= 0 size
backupMultipleTimes (corner)	29	multiple backups	>= 0 size	>= 0 size
logAfterDisablingAll (corner)	30	log after disable	false both loggings	false both loggings
logSystemInfoMultiple (corner)	31	multiple system info	true file logging	true file logging

NotificationSenderTest.java

Function name	Test case ID	Input	Expected output	Actual output
sendWelcomeEmail	1	"test@example.com", "Test User"	no exception	no exception
sendBookingConfirmation	2	"booking@example.com", "Booking User", "BK001"	no exception	no exception
sendPaymentConfirmation	3	"payment@example.com", "Payment User", "PMT123", 500.00	no exception	no exception
sendSMS	4	"01700000000", "Your ticket has been confirmed."	no exception	no exception
sendWelcomeEmail (corner)	5	null/empty parameters	no exception	no exception
sendBookingConfirmation (corner)	6	null/empty parameters	no exception	no exception
sendPaymentConfirmation (corner)	7	null/empty parameters	no exception	no exception
sendSMS (corner)	8	null/empty parameters	no exception	no exception
sendMultipleRapidly (corner)	9	10 rapid notifications	no exception	no exception

PackageCategoryTest.java

Function name	Test case ID	Input	Expected output	Actual output
values	1	empty	[LOCAL, INTERNATIONAL, SEASONAL]	[LOCAL, INTERNATIONAL, SEASONAL]
valueOf	2	"LOCAL"	LOCAL	LOCAL

valueOf	3	"INTERNATIONAL"	INTERNATIONAL	INTERNATIONAL
valueOf	4	"SEASONAL"	SEASONAL	SEASONAL
getDisplayName	5	LOCAL	"Local Package"	"Local Package"
getDisplayName	6	INTERNATIONAL	"International Package"	"International Package"
getDisplayName	7	SEASONAL	"Seasonal Package"	"Seasonal Package"
getDescription	8	LOCAL	"Domestic destinations"	"Domestic destinations"
getDescription	9	INTERNATIONAL	"Foreign destinations"	"Foreign destinations"
getDescription	10	SEASONAL	"Special seasonal offers"	"Special seasonal offers"
valueOf (corner)	11	"UNKNOWN"	IllegalArgumentException	IllegalArgumentException
valueOf (corner)	12	null	NullPointerException	NullPointerException
valueOf (corner)	13	empty string	IllegalArgumentException	IllegalArgumentException
getDisplayName (corner)	14	all categories	not null	not null
getDescription (corner)	15	all categories	not null	not null

PackageReviewTest.java

Function name	Test case ID	Input	Expected output	Actual output
markAsHelpful	1	empty	previous count + 1	previous count + 1
verifyReview	2	empty	true	true
unverifyReview	3	empty	false	false
getReviewSummary	4	empty	contains review details	contains review details
getRatingStars	5	4.5 rating	"★★★★☆"	"★★★★☆"
isPositiveReview	6	4.5 rating	true	true
isNegativeReview	7	1.5 rating	true	true
isNeutralReview	8	3.0 rating	true	true

getReviewCategory	9	various ratings	correct category	correct category
getReviewLength	10	review text	text length	text length
updateReview	11	new content, rating	updated values	updated values
addProsCons	12	pros/cons strings	set pros/cons	set pros/cons
setReviewId	13	"R001"	"R001"	"R001"
setUserId	14	"User123"	"User123"	"User123"
setUserName	15	"Tanvir"	"Tanvir"	"Tanvir"
setPackageId	16	"PKG123"	"PKG123"	"PKG123"
setPackageName	17	"Sundarbans Tour"	"Sundarbans Tour"	"Sundarbans Tour"
setReviewText	18	"Fantastic experience!"	"Fantastic experience!"	"Fantastic experience!"
setRating	19	4.2	4.2	4.2
setReviewDate	20	"2025-01-01"	"2025-01-01"	"2025-01-01"
setVerified	21	true	true	true
setHelpfulCount	22	3	3	3
setReviewTitle	23	"Awesome Trip"	"Awesome Trip"	"Awesome Trip"
setRecommended	24	false	false	false
setTravelDate	25	"2025-12-15"	"2025-12-15"	"2025-12-15"
setTravelType	26	"Family"	"Family"	"Family"
setPros	27	"Affordable, Scenic"	"Affordable, Scenic"	"Affordable, Scenic"
setCons	28	"Too hot during summer"	"Too hot during summer"	"Too hot during summer"
setAnonymous	29	true	true	true
toString	30	review object	contains ID and rating	contains ID and rating

equals	31	same review ID	true	true
hashCode	32	same review ID	same hash	same hash
markAsHelpfulMultiple (corner)	33	multiple calls	count + 2	count + 2
updateReviewEmpty (corner)	34	empty text	empty text set	empty text set
updateReviewNegativeRating (corner)	35	-2.0 rating	-2.0 set	-2.0 set
setRatingZero (corner)	36	0.0	0.0	0.0
setRatingAboveFive (corner)	37	6.0	6.0	6.0
equalsWithNull (corner)	38	null	not equal	not equal
equalsWithDifferentClass (corner)	39	string object	not equal	not equal
equalsWithSameObject (corner)	40	same object	equal	equal
hashCodeDifferentIds (corner)	41	different IDs	different hash	different hash
toStringWithNullValues (corner)	42	null values	contains rating	contains rating
emptyUserAndPackageName (corner)	43	empty names	contains "Review by"	contains "Review by"

PackageUITest.java

Function name	Test case ID	Input	Expected output	Actual output
browseTourPackages	1	empty	void	void
searchPackages	2	empty	void	void
viewPackageDetails	3	"PKG001"	void	void

createNewPackage	4	empty	void	void
editPackage	5	"PKG002"	void	void
displayPackageManagement	6	empty	void	void

PDFExporterTest

Function name	Test case ID	Input	Expected output	Actual output
exportDailyReport	1	"Sample daily report", "2025-08-19"	true	true
exportMonthlyReport	2	"Sample monthly report", "2025-08"	true	true
exportAnnualReport	3	"Sample annual report", "2025"	true	true
exportReport	4	"Generic report", "report.pdf"	true	true
exportUserList	5	User list	true	true
exportBookingList	6	Booking list	true	true
exportPaymentList	7	Payment list	true	true
exportTicket	8	Ticket object	true	true
exportCustomReport	9	"Custom Report", content, "custom_report.pdf"	true	true
exportBulkData	10	"users", "2025-08-01", "2025-08-19"	true	true

PriceCalculatorTest

Function name	Test case ID	Input	Expected output	Actual output
calculateFinalPrice	1	TourPackage, "AGENT", 5 people	4189.5	4189.5
calculateFinalPrice (corner)	2	TourPackage, "CUSTOMER", 0 people	0.0	0.0
calculateFinalPrice (corner)	3	TourPackage, "CUSTOMER", -3 people	-2940.0	-2940.0
calculateFinalPrice (corner)	4	TourPackage, "UNKNOWN", 2 people	2000.0	1960.0
calculateFinalPrice (corner)	5	TourPackage, "AGENT", 50 people	< base price	< base price
calculateFinalPrice (corner)	6	null TourPackage	NullPointerException	NullPointerException
calculateFinalPrice (corner)	7	TourPackage, "CUSTOMER", 1 person	980.0	980.0

RefundManagerTest

Function name	Test case ID	Input	Expected output	Actual output
createRefundRequest	1	"B123", "U456", "Travel canceled", 1500.0	not null request	not null request
removeRefundRequest	2	request ID	null after removal	null after removal
getAllRefundRequests	3	request created	not empty list	not empty list
getPendingRefundRequests	4	pending request	contains pending	contains pending
getApprovedRefundRequests	5	approved request	contains approved	contains approved

getRejectedRefundRequests	6	rejected request	contains rejected	contains rejected
getRefundRequestsByUser	7	"USER123"	not empty list	not empty list
getRefundRequestsByBooking	8	"BK999"	not empty list	not empty list
getTotalRefundAmount	9	completed refund	0.0	0.0
getTotalRefundRequests	10	new request	previous + 1	previous + 1
getRefundApprovalRate	11	approved request	> 0.0	> 0.0
getRefundRequestsByStatus	12	pending request	contains "PENDING"	contains "PENDING"
searchRefundRequests	13	"Storm"	not empty	not empty
getRefundRequestById (corner)	14	"NO_SUCH_ID"	null	null
removeRefundRequestNonExisting (corner)	15	"NO_SUCH_ID"	same total count	same total count
removeRefundRequestTwice (corner)	16	same ID twice	null after removal	null after removal
getApprovedRefundRequestsEmpty (corner)	17	no approved requests	empty list	empty list
getRejectedRefundRequestsEmpty (corner)	18	no rejected requests	empty list	empty list
getRefundRequestsByUserUnknown (corner)	19	"UNKNOWN_USER"	empty list	empty list
getRefundRequestsByBookingUnknown (corner)	20	"UNKNOWN_BOOKING"	empty list	empty list
getTotalRefundAmountZero (corner)	21	no requests	0.0	0.0
getRefundApprovalRateZero (corner)	22	no requests	0.0	0.0
searchRefundRequestsNoMatch (corner)	23	"no-match-keyword"	empty list	empty list
createRefundRequestZeroAmount (corner)	24	0.0 amount, empty reason	not null	not null
createRefundRequestLongReason (corner)	25	256 char reason	not null	not null
getAllRefundRequestsMultiple (corner)	26	multiple requests	size >= 2	size >= 2

TourGuideTest.java

Function name	Test case ID	Input	Expected output	Actual output
addLanguage	1	"English", "Bangla", "English"	["English", "Bangla"]	["English", "Bangla"]
updateRating	2	4.0, 5.0	4.5, 2 reviews	4.5, 2 reviews

assignAndCompleteTour	3	"T001", "T002"	correct availability	correct availability
setGuideId	4	"G100"	"G100"	"G100"
setName	5	"Rahim Uddin"	"Rahim Uddin"	"Rahim Uddin"
setPhone	6	"01987654321"	"01987654321"	"01987654321"
setEmail	7	"rahim@example.com"	"rahim@example.com"	"rahim@example.com"
setLanguages	8	["Spanish", "French"]	["Spanish", "French"]	["Spanish", "French"]
setRating	9	4.7	4.7	4.7
setTotalReviews	10	10	10	10
setSpecialization	11	"Mountain Tours"	"Mountain Tours"	"Mountain Tours"
setDailyRate	12	3000.0	3000.0	3000.0
setAvailable	13	false	false	false
setExperience	14	"7 years"	"7 years"	"7 years"
setCertifications	15	"Advanced Guide Certificate"	"Advanced Guide Certificate"	"Advanced Guide Certificate"
getAssignedTours	16	initially	empty list	empty list
toString	17	guide with data	contains details	contains details
addLanguageCaseSensitive (corner)	18	"english", "ENGLISH"	both added	both added
updateRatingZeroAndFive (corner)	19	0.0, 5.0	2.5 average	2.5 average
updateRatingNegativeAndAboveMax (corner)	20	-1.0, 6.0	2.5 average	2.5 average
assignTourDuplicateAnd CompleteUnassigned (corner)	21	duplicate assignment	2 tours assigned	2 tours assigned
completeTourEmptyList (corner)	22	complete non-assigned	empty list	empty list
setLanguagesNull (corner)	23	null list	null	null
setDailyRateNegative (corner)	24	-500.0	-500.0	-500.0
assignAndCompleteMultiple (corner)	25	multiple tours	correct final state	

TouristMenuTest.java

Function name	Test case ID	Input	Expected output	Actual output
TouristMenuConstructorExists	1	constructor check	constructor exists	constructor exists
getLoyaltyTierBronze	2	0, 100, 499 points	"BRONZE"	"BRONZE"
getLoyaltyTierSilver	3	500, 750, 999 points	"SILVER"	"SILVER"
getLoyaltyTierGold	4	1000, 1500, 1999 points	"GOLD"	"GOLD"
getLoyaltyTierPlatinum	5	2000, 5000, 10000 points	"PLATINUM"	"PLATINUM"
getNextTierPointsBronzeToSilver	6	100, 250, 450 points	400, 250, 50 points	400, 250, 50 points
getNextTierPointsSilverToGold	7	600, 800 points	400, 200 points	400, 200 points
getNextTierPointsGoldToPlatinum	8	1200, 1700 points	800, 300 points	800, 300 points
getNextTierPointsPlatinumMax	9	2500, 10000 points	0 points	0 points
privateMethodsExist	10	method check	methods exist	methods exist
fieldsExist	11	field check	fields exist	fields exist
methodParameterTypes	12	parameter check	correct types	correct types
loyaltyTierEdgeCases	13	boundary values	correct tiers	correct tiers
nextTierPointsEdgeCases	14	boundary values	correct points needed	correct points needed

VehicleTest.java

Function name	Test case ID	Input	Expected output	Actual output
vehicleInitialization	1	constructor params	correct initialization	correct initialization

setAndGetFields	2	various field values	correct field values	correct field values
addFeature	3	"GPS", "Air Conditioning"	features added	features added
removeFeature	4	remove "Heated Seats"	feature removed	feature removed
addFeatureNullOrEmpty (corner)	5	null, empty string	features added	features added
addDuplicateFeature (corner)	6	duplicate "GPS"	2 GPS entries	2 GPS entries
removeFeatureNotPresent (corner)	7	remove "Sunroof"	no change	no change
setSeatsAvailableNegative (corner)	8	-5	-5	-5
setFuelCapacityNegative (corner)	9	-50.0	-50.0	-50.0
operationalToggle (corner)	10	toggle operational	correct states	correct states
multipleFeatures (corner)	11	multiple add/remove	correct final state	correct final state
setFieldsNullOrEmpty (corner)	12	null/empty values	null/empty set	null/empty set

GuideManagerTest.java

Function name	Test case ID	Input	Expected output	Actual output
addGuide	1	TourGuide object	guide added, not null	found
removeGuide	2	"1001"	guide removed	found(null)
getGuideById	3	"1001"	get Ali by id	found("Ali")
updateGuide	4	rating set 5.0	updated rating 5.0	5.0
getAllGuides	5	-	list size >= 1	size >= 1
getAvailableGuides	6	-	contains guide	found(contains)
getGuidesByLanguage	7	"Bangla"	contains guide	found(contains)

getGuidesByRating	8	4.5	guide with >=4.5	found(contains)
getGuidesBySpecialization	9	"historical"	contains guide	found(contains)
searchGuides	10	"Ali"	contains guide	found
getTotalGuides	11	-	1	1
getAvailableGuidesCount	12	-	1	1
getAverageGuideRating	13	-	4.8	4.8
getTopRatedGuide	14	-	"Ali"	"Ali"
getGuideCountByLanguage	15	-	English >= 1	found
updateGuideRating	16	"1001", 4.0	rating 4.0	4.0

InputHandlerTest.java

Function name	Test case ID	Input	Expected output	Actual output
getString	1	"Hello World"	"Hello World"	"Hello World"
getInt	2	"123"	123	123
getDouble	3	"45.67"	45.67	45.67
getBoolean (yes)	4	"y"	true	true
getBoolean (no)	5	"n"	false	false
pressEnterToContinue	6	Enter	no exception	success
closeScanner	7	-	closed	called

PackageManagerTest.java

Function name	Test.C ID	Input	Expected output	Actual output
addPackage	1	pkg003, "City Tour"	added	found
removePackage	2	"pkg001"	removed	null
listPackages	3	-	>=2	size >= 2
listActivePackages	4	-	active only	contains one
getPackageById	5	"pkg001"	"Beach Holiday"	"Beach Holiday"

searchPackages	6	"beach"	contains/doesn’t contain	as expected
getPackagesByCategory	7	SEASONAL / LOCAL	only category	as expected
getPackagesByTourType	8	ADVENTURE/FAMILY	only type	as expected
getPackagesByPriceRange	9	9000 ~ 16000	in-range	contains both
getPackagesByDuration	10	3~5	all in duration	contains both
getPackagesByLocation	11	"Sajek"	samplePackage2 only	contains/absent
getPackagesByCreator	12	"admin"	admin’s pkgs only	as expected
updatePackage	13	name changed	updated name	"Beach Paradise"
activateDeactivatePackage	14	deactivate/activate	updated activity	bool ok
getTotalPackages	15	-	>=2	>=2
getActivePackagesCount	16	-	active=1	1
getAveragePackagePrice	17	-	avg > 0	ok
getMostExpensivePackage	18	-	samplePackage1	ok
getCheapestPackage	19	-	samplePackage2	ok
getPackageCountByCategory	20	-	map values	ok
getPackageCountByTourType	21	-	map values	ok
getAvailablePackages	22	-	contains both	ok
getFullyBookedPackages	23	-	none	ok

RoomTypeTest.java

Function name	Test case ID	Input	Expected output	Actual output
values	1	-	STANDARD, DELUXE, SUITE	same
valueOf	2	"STANDARD"	RoomType.STANDARD	STANDARD
valueOf	3	"DELUXE"	RoomType.DELUXE	DELUXE
valueOf	4	"SUITE"	RoomType.SUITE	SUITE
getDisplayName	5	STANDARD	"Standard Room"	"Standard Room"

getDisplayName	6	DELUXE	"Deluxe Room"	"Deluxe Room"
getDisplayName	7	SUITE	"Suite Room"	"Suite Room"
getBasePrice	8	STANDARD	1500.0	1500.0
getBasePrice	9	DELUXE	2500.0	2500.0
getBasePrice	10	SUITE	4000.0	4000.0

SeatAllocatorTest

Function name	Test case ID	Input	Expected output	Actual output
autoAssignSeats (empty list)	1	transport, count 2	empty list	empty list
findBestAvailableSeat (preferred exists)	2	[Window, Aisle], pref="Window"	seat type Window	seat type Window
findBestAvailableSeat (no preferred)	3	[Middle, Middle], pref="Window"	seat type Middle	seat type Middle
findBestAvailableSeat (empty)	4	[], pref="Window"	null	null

SeatTest.java

Function name	Test case ID	Input	Expected output	Actual output
bookSeat (success)	1	"Jahid", "0123456789"	true, not available, set values	as expected
bookSeat (blocked)	2	blocked	false, still available	as expected
cancelSeat	3	booked	available, null values	as expected
gettersAndSetters	4	various props	all set and got	as expected

toString_containsSeatNumberAndName	5	booked "John Doe", "012345..."	string contains seat and passenger	as expected

TicketTest.java

Function name	Test case ID	Input	Expected output	Actual output
useTicket	1	-	unused→used	as expected
generateTicketText	2	-	everything in string	as expected
getAndSetTicketId	3	"T002"	id updated	"T002"
getAndSetBookingId	4	"B002"	booking id updated	"B002"
getAndSetQRcode	5	"NEWQR..."	qr updated	"NEWQR..."
getAndSetIssueDate	6	"2025-08-07"	issue date updated	"2025-08-07"
getAndSetValidFrom	7	"2025-08-10"	validFrom updated	"2025-08-10"
getAndSetValidUntil	8	"2025-08-20"	validUntil updated	"2025-08-20"
getAndSetPassengerName	9	"Jane Doe"	name updated	"Jane Doe"
getAndSetPassengerPhone	10	"0123456789"	phone updated	"0123456789"
getAndSetPackageName	11	"Business Package"	name updated	"Business Package"
getAndSetSeatNumber	12	"A1"	seat updated	"A1"
isUsedAndSetUsed	13	true/false	used status change	as expected
toString	14	-	contains id, passenger, used	as expected

TransportTest.java

Function name	Test case ID	Input	Expected output	Actual output
cancelSeat (normal)	1	after book	available+1	ok
cancelSeat (boundary)	2	at max	not >total	ok
getAndSetTransportId	3	"NEWID123"	id updated	"NEWID123"
getAndSetType	4	FLIGHT	type updated	FLIGHT
getAndSetDeparture	5	"Sylhet"	dep updated	"Sylhet"
getAndSetArrival	6	"Khulna"	arr updated	"Khulna"
getAndSetDepartureTime	7	"10:00 AM"	dep time updated	"10:00 AM"
getAndSetArrivalTime	8	"2:00 PM"	arr time updated	"2:00 PM"
getAndSetFare	9	1500.0	fare updated	1500.0
getAndSetTotalSeats	10	50	total seats updated	50
getAndSetAvailableSeats	11	20	avail seats updated	20
getAndSetOperatorName	12	"Best ..."	operator updated	"Best ..."
isActiveAndSetActive	13	false	status updated	false
toString	14	-	string has fields	ok

TravelAgentTest.java

Function name	Test case ID	Input	Expected output	Actual output
getRole	1	-	"AGENT"	"AGENT"
setAndGetAgencyName	2	"Best Tours Ltd."	updated	"Best Tours Ltd."
setAndGetLicenseNumber	3	"LIC123456"	updated	"LIC123456"
getManagedPackagesInitiallyEmpty	4	-	empty list	empty list

getAndSetCommissionRate	5	0.15	updated	0.15
inheritanceFromUser	6	-	all user props ok	ok

AuthServiceTest.java

Function name	Test case ID	Input	Expected output	Actual output
testLogout	1	email	logout completes without exception	passed
testIsUserAuthenticated	2	email	false (not authenticated initially)	asserted
testHasRole	3	email, role	false (should not have role before registration)	asserted
testHasAnyRole	4	email, roles	false (should not have any role before registration)	asserted
testIsAdmin	5	email	false (default user not admin)	asserted
testIsTourist	6	email	false	asserted
testIsAgent	7	email	false	asserted
testAuthorizeAccess	8	email, role	false (access not authorized for unregistered user)	asserted
testAuthorizeAdminAccess	9	email	false	asserted
testChangePassword	10	email, old, new	result not null	passed

testRequestPasswordReset	11	email	result not null	passed
testResetPassword	12	email, token, new	result not null	passed
testBlockUser	13	email, reason	completes without exception	passed
testUnblockUser	14	email	completes without exception	passed
testIsAccountLocked	15	email	false	asserted
testLogoutWithNullEmail	16	null	no crash, completes	passed
testHasRoleWithNulls	17	null, null	false	asserted
testChangePasswordWithNulls	18	null, null, null	result not null	passed
testRequestPasswordResetWithNull	19	null	result not null	passed
testResetPasswordWithNulls	20	null, null, null	result not null	passed
testBlockUserWithNulls	21	null, null	exception handled, completes/not crash	passed
testUnblockUserWithNull	22	null	completes/not crash	passed

DataSeederTest.java

Function name	Test case ID	Input	Expected output	Actual output
testSeedInitialData	1	managers	all not null after seeding, no exception	passed

testSeedInitialDataWithAlreadyPopulatedManagers	2	managers with prior data	user/hotel count not decrease/duplicate	passed
testSeedInitialDataWithNullManagers	3	null managers	NullPointerException or handled	exception/ asserted
testSeedInitialDataWithPartialNullManagers	4	one manager null	NullPointerException or handled	exception/ asserted

IDGeneratorTest.java

Function name	Test case ID	Input	Expected output	Actual output
testGenerateUserId	1	none	startsWith "USR", not null	passed
testGeneratePackageId	2	none	startsWith "PKG", not null	passed
testGenerateBookingId	3	none	startsWith "BKG", not null	passed
testGenerateHotelId	4	none	startsWith "HTL", not null	passed
testGenerateTransportId	5	none	startsWith "TRAN"/"TRN"/"TR", not null	passed
testGenerateInvoiceId	6	none	startsWith "INV", not null	passed
testGenerateRandomStringWithLengthOnly	7	length	string of given length, not null	passed
testGenerateRandomStringWithCustomChars	8	length, custom chars	string with only chars from custom set	passed
testIsValidIdFormat	9	id, prefix	valid = true	passed
testGenerateSecureToken	10	length	string, length	passed
testGenerateRandom	11	0	empty string	passed

StringWithZeroLength				
testGenerateRandom StringWithNegativeLength	12	negative	empty or exception	passed/ exception handled
testGenerateRandom StringWithEmptyCustomChars	13	len, empty chars	length or exception	passed/ exception handled
testIsValidIdFormatWith WrongPrefix	14	invalid prefix	false	passed
testIsValidIdFormat WithNullId	15	null id	false	passed
testGenerateSecureToken WithZeroLength	16	0	empty string	passed
testGenerateSecureToken WithNegativeLength	17	negative	empty/exception	passed/ exception handled

LocationTest.java

Function name	Test case ID	Input	Expected output	Actual output
testAddLandmark	1	new landmark	landmark added	passed
testRemoveLandmark	2	landmark	removed	passed
testGetLocationId	3	none	correct id	passed
testGetDescription	4	none	returns description	passed
testSetDescription_Empty	5	empty string	sets to empty	passed
testGetClimate	6	none	returns climate	passed
testSetClimate_Null	7	null	sets null	passed
testGetBestTimeToVisit	8	none	returns best time	passed
testSetBestTimeToVisit_Empty	9	empty	sets to empty	passed
testToString	10	none	returns info string	passed

testGetCity	11	none	returns city	passed
testGetCountry	12	none	returns country	passed
testSetCountry_Null	13	null	sets null	passed
testGetState	14	none	returns state	passed
testSetState_Empty	15	empty	sets empty	passed
testGetLandmarks	16	none	returns landmarks	passed
testAddLandmark_NullValue	17	null	adds null (allowed)	passed
testAddLandmark_Duplicate	18	duplicate	allows duplicate	passed
testRemoveLandmark_NotExists	19	not in list	no change	passed
testSetLocationId_EmptyString	20	empty string	sets empty	passed
testSetLocationId_Null	21	null	sets null	passed
testSetCity_SpecialCharacters	22	special chars	stores special chars	passed
testSetLandmarks_EmptyList	23	empty list	list empty	passed
testSetLandmarks_NullList	24	null	sets null	passed
testGetLatitude	25	none	returns latitude	passed
testSetLatitude_OutOfRange	26	invalid value	stores as is	passed
testGetLongitude	27	none	returns longitude	passed
testSetLongitude_OutOfRange	28	invalid value	stores as is	passed

MainMenuTest.java

Function name	Test case ID	Input	Expected output	Actual output
testEmergencyShutdown	1	none	systemRunning false after call	passed
testIsSystemRunning_DefaultTrue	2	none	true initially	passed
testGetUserManager	3	none	not null	passed

testGetAuthService	4	none	not null	passed
testEmergencyShutdownTwice	5	none	remains false	passed
testIsSystemRunningAfterMultipleOperations	6	none	stays true	passed
testGetUserManagerNotNullEvenAfterShutdown	7	none	not null	passed
testGetAuthServiceMultipleCallsReturnSameInstance	8	none	same instance	passed
testAuthServiceWorksAfterShutdown	9	none	still retrievable	passed
testSystemRunningStatePersistence	10	none	stays shutdown	passed
testGetUserManagerMayReturnNullIfNotInitialized	11	constructor null	getUserManager null	passed

MainTest.java

Function name	Test case ID	Input	Expected output	Actual output
testMain	1	args	main runs, args unchanged	passed
testGetUserManager	2	none	UserManager/null	passed
testGetPackageManager	3	none	PackageManager/null	passed
testGetBookingManager	4	none	BookingManager/null	passed
testGetPaymentManager	5	none	PaymentManager/null	passed
testGetHotelManager	6	none	HotelManager/null	passed
testGetTransportManager	7	none	TransportManager/null	passed
testGetGuideManager	8	none	GuideManager/null	passed
testGetReviewManager	9	none	ReviewManager/null	passed
testGetRefundManager	10	none	RefundManager/null	passed
testGetRoomManager	11	none	RoomManager/null	passed
testGetAuthService	12	none	AuthService/null	passed
testGetReportGenerator	13	none	ReportGenerator/null	passed

testSingletonManagers	14	none	singleton (same instance)	passed
testMainWithNullArgs	15	null	main executes with null args	passed
testMainWithLongArgs	16	long[]	executes successfully	passed
testManagersNullSafety	17	none	getters provide non-null after access	passed

PaymentManagerTest.java

Function name	Test case ID	Input	Expected output	Actual output
testProcessPayment	1	paymentId	process, status COMPLETED/PAID	passed
testFailPayment	2	paymentId, reason	fail, status FAILED	passed
testRefundPayment	3	paymentId	refund, status REFUNDED	passed
testProcessPaymentNonExistent	4	non-existent	returns false	passed
testFailPaymentNonExistent	5	non-existent	returns false	passed
testRefundPaymentNonExistent	6	non-existent	returns false	passed
testRefundPaymentNotCompleted	7	pending payment	returns false	passed
testAddDuplicatePayment	8	dup id	keeps latest or original	passed
testZeroAmountPayment	9	amount 0	stores 0	passed
testNegativeAmountPayment	10	amount <0	stores negative	passed
testGetPaymentByIdNonExistent	11	non-existent	returns null	passed
testProcessPaymentNullId	12	null	returns false	passed
testFailPaymentNullId	13	null, reason	returns false	passed
testRefundPaymentNullId	14	null	returns false	passed
testGetPaymentByIdNull	15	null	returns null	passed

RoomTest.java

Function name	Test case ID	Input	Expected output	Actual output
testIsAvailable	1	none	true initially	passed
testBookRoom	2	none	available false after booking	passed
testCheckOut	3	after booking	available true after checkout	passed
testGetRoomId	4	none	correct id	passed
testGetHotelId	5	none	correct hotel id	passed
testSetHotelId_SpecialCharacters	6	special chars	stores special chars	passed
testGetRoomNumber	7	none	returns number	passed
testSetRoomNumber_Empty	8	empty	empty stored	passed
testGetRoomType	9	none	correct type	passed
testIsAvailability	10	none	equals isAvailable	passed
testSetAvailability_False	11	false	false	passed
testGetPrice	12	none	correct price	passed
testGetCapacity	13	none	correct capacity	passed
testGetDescription	14	none	returns description	passed
testSetDescription_Empty	15	empty	empty stored	passed
testIsHasAC	16	none	true	passed
testSetHasAC_False	17	false	false	passed
testIsHasWiFi	18	none	true	passed
testSetHasWiFi_False	19	false	false	passed

testIsHasTV	20	none	true	passed
testSetHasTV_False	21	false	false	passed
testGetBedType	22	none	"King"	passed
testSetBedType_Empty	23	empty	empty stored	passed
testToString	24	none	returns info string	passed
testSetPrice_Negative	25	negative	stores value as is	passed
testSetPrice_Zero	26	0	sets 0	passed
testSetRoomId_Null	27	null	null stored	passed
testSetRoomId_Empty	28	empty	empty stored	passed
testSetRoomType_Null	29	null	null stored	passed
testCheckOut_WhenAlreadyAvailable	30	none	stays true	passed
testSetCapacity_Zero	31	0	sets 0	passed
testSetCapacity_Negative	32	negative	sets negative	passed
testSetBedType_Null	33	null	null stored	passed
testBookRoom_WhenAlreadyBooked	34	book again	no crash, stays unavailable	passed
testToString_AllFieldsNull	35	all null	returns string (no crash)	passed

ScheduleTest.java

Function name	Test case ID	Input	Expected output	Actual output
testAddAndRemoveScheduleItem	1	item	add, remove = empty list	passed
testConfirmAndCancelSchedule	2	none	confirm, cancel = status CANCELLED	passed
testCompleteSchedule	3	none	status COMPLETED	passed

testUpdateScheduleItem	4	item	updated fields	passed
testGetScheduleItemsByDate	5	date	items returned	passed
testHasConflictWithTime	6	same date/time	true (conflict)	passed
testGenerateScheduleReport	7	none	report string contains info	passed
testBasicGettersAndSetters	8	various	correct get/set	passed
testToString	9	none	info string	passed
testRemoveNonExistingItem	10	not found id	no error	passed
testUpdateNonExistingItem	11	not found id	no error	passed
testGetScheduleItemsByDateNoMatch	12	date	empty list	passed
testHasConflictWithDifferentDate	13	different date	false	passed
testEmptyScheduleReport	14	none	no error, contains report text	passed
testConstructorWithNullIds	15	all null	nullable fields	passed
testAddScheduleItemWithNullValues	16	all null	still added, size==1	passed
testUpdateScheduleItemWithNullValues	17	nulls update	null fields	passed
testGetScheduleItemsByDateWithNull	18	null date	empty list	passed
testHasConflictWithNullValues	19	nulls	false (no crash)	passed

TouristTest.java

Function name	Test case ID	Input	Expected output	Actual output
testGetRole	1	none	"TOURIST" or not null	passed
testAddLoyaltyPoints	2	points	increases	passed
testGetPreferences	3	none	contains added preference	passed

testSetPreferences	4	new prefs	updated	passed
testGetNationality	5	none	returns nationality	passed
testSetNationality	6	value	updated	passed
testGetLoyaltyPoints	7	none	>=100	passed
testGetBookingHistory	8	none	not null/list	passed
testAddZeroLoyaltyPoints	9	0	unchanged	passed
testAddNegativeLoyaltyPoints	10	negative	decreases	passed
testSetEmptyPreferences	11	empty list	empty list	passed
testSetNullPreferences	12	null	null set	passed
testSetEmptyNationality	13	""	""	passed
testSetNullNationality	14	null	null	passed
testEmptyBookingHistory	15	new tourist	empty list	passed
testPreferencesImmutableReference	16	modify list	changes reflected	passed

TourPackageTest.java

Function name	Test case ID	Input	Expected output	Actual output
testCancelBooking	1	currentBookings	decreases by 1	passed
testAddBooking	2	currentBookings	increases by 1	passed
testGetCurrentBookings	3	set to value	value matches	passed
testSetCurrentBookings	4	value	set correctly	passed
testGetMaxCapacity	5	set to value	value matches	passed
testSetMaxCapacity	6	value	set correctly	passed
testGetRating	7	set to value	matches	passed
testSetRating	8	value	matches	passed
testGetTotalReviews	9	set to value	matches	passed
testSetTotalReviews_Negative	10	negative	sets negative	passed
testSetTotalReviews	11	value	sets correctly	passed
testGetDiscountPercentage	12	set to value	matches	passed

testSetDiscountPercentage	13	value	matches	passed
testGetCategory	14	null	null	passed
testSetCategory	15	null	null	passed
testGetTourType	16	null	null	passed
testSetTourType	17	null	null	passed
testIsActive	18	set true	true	passed
testSetActive	19	set false	false	passed
testIsFeatured	20	default false	false	passed
testSetFeatured	21	set true	true	passed
testGetCreatedBy	22	set value	matches	passed
testSetValidTo	23	value	matches	passed
testSetCreatedBy	24	value	matches	passed
testGetCreatedDate	25	set value	matches	passed
testSetCreatedDate	26	value	matches	passed
testGetLastModified	27	set value	matches	passed
testSetLastModified	28	value	matches	passed
testGetPackageId	29	none	initial id	passed
testSetPackageId	30	value	matches	passed
testGetValidFrom	31	set value	matches	passed
testSetValidFrom	32	value	matches	passed
testGetValidTo	33	set value	matches	passed
testSetCurrentBookings_Negative	34	negative	sets negative	passed
testSetMaxCapacity_Zero	35	zero	sets zero	passed
testSetMaxCapacity_Negative	36	negative	sets negative	passed
testSetCreatedDate_Null	37	null	sets null	passed
testSetRating_OutOfRangeHigh	38	>5	stores as is	passed
testSetRating_OutOfRangeLow	39	<0	stores as is	passed
testSetCreatedBy_Empty	40	""	stores empty	passed
testAddBooking_WhenAtMaxCapacity	41	at capacity	allows/ignored depending on logic	passed
testSetLastModified_Empty	42	""	stores empty	passed
testSetPackageId_Null	43	null	sets null	passed

testSetDiscountPercentage_Negative	44	negative	stores negative	passed
testSetDiscountPercentage_Above100	45	>100	stores	passed
testSetValidTo_Null	46	null	sets null	passed
testSetValidFrom_Empty	47	""	sets empty	passed
testCancelBooking_WhenZeroBookings	48	0	stays >=0	passed

TourTypeTest.java

Function name	Test case ID	Input	Expected output	Actual output
testValues	1	none	enums not null, display+desc not empty	passed
testValueOf_Valid	2	valid name	matches	passed
testGetDisplayName	3	none	all not null, not empty	passed
testGetDescription	4	none	all not null, not empty	passed
testOrdinalOrder	5	none	ordinals sequential	passed
testValueOf_Invalid	6	invalid name	IllegalArgumentException	thrown
testValueOf_Null	7	null	NullPointerException	thrown
testEnumConsistency	8	none	name() == toString()	passed

userTest.java

Function name	Test case ID	Input	Expected output	Actual output
testSetId	1	value/""/null	matches/empty/null	passed
testSetName	2	value/""/null	matches/empty/null	passed
testSetEmail	3	value/invalid/null	matches	passed
testSetPassword	4	value/""/null	matches/empty/null	passed
testSetPhone	5	value/invalid/null	matches	passed
testGetCreatedDate	6	none	not null	passed
testGetName	7	none/""/null	matches/empty/null	passed
testGetEmail	8	none/invalid/null	matches	passed
testGetPassword	9	none/""/null	matches/empty/null	passed

testGetPhone	10	none/invalid/null	matches	passed
testIsActive	11	true/false	matches active	passed
testSetActive	12	true/false	matches	passed
testGetRole	13	none	"Tester"/matches	passed
testToString	14	value/empty	contains values/empty	passed
testGetId	15	none/""/null	matches/empty/null	passed

BookingUITest.java

Function name	Test case ID	Input	Expected output	Actual output
testCreateBookingNormalFlow	1	Select package 1, travel date 2025-09-10, 2 people, confirm booking 'y'	Booking created with correct number & discounted total	Booking added & validated
testBookingCancelledByUser	2	Select package 1, travel date 2025-09-10, 2 people, cancel booking 'n'	No booking added	No booking found
testInvalidPackageChoice	3	Input invalid package number 10	Invalid package number message	Displayed invalid input msg

CurrencyFormatterTest.java

Function name	T.C ID	Input	Expected output	Actual output
testFormatBDT	1	Amount=12345.678	Formatted string ৳12,345.68	Formatted string matches expected
testFormatUSD	2	Amount=9876.543	Formatted string $9,876.54	Formatted string matches expected
testFormatCurrency	3	Amount=1000.5, CurrencyCode=BDT/USD/EUR	Formatted string (symbol or code)	Matches for all codes
testConvertUSDToBDT	4	usdAmount=10.0	1,100.0	Correct conversion
testConvertBDTToUSD	5	bdtAmount=2200.0	20.0	Correct conversion
testFormatZeroAmount	6	Amount=0	"৳0.00", "$0.00"	Matched expected
testFormatNegativeAmount	7	Amount=-123.45	"-৳123.45", "-$123.45"	Matched expected
testFormatLargeAmount	8	Amount=9876543210.99	Formatted string with comma	Contains comma
testFormatHighPrecision	9	Amount=1.9999	"৳2.00", "$2.00"	Rounded as expected
testFormatCurrencyNullCode	10	Amount=123.45, CurrencyCode=null	NullPointerException	Exception thrown

testFormatCurrencyUnsupportedCode	11	Amount=123.45, CurrencyCode=JPY	"123.45 JPY"	Matched expected
testConvertZeroAmounts	12	Amount=0	0.0	Correct conversion
testConvertNegativeAmounts	13	Amount=-10/-1100	-1100.0/-10.0	Correct conversion

GroupTourTest.java

Function name	Test case ID	Input	Expected output	Actual output
testAddBooking_normalAndCornerCases	1	Booking adding for group trip	Add booking, handle duplicate, limit, group full	All scenarios verified
testRemoveBooking_normalAndCornerCases	2	Remove booking scenarios	Remove existing, handle non-existent, empty group	All cases matched
testParticipantNamesAndSpecialArrangements	3	Add participant and arrangements	Names and arrangements reflected	Correct names and data shown
testGenerateGroupReport	4	Group with participants, guide, meeting point	Report includes all info	All data present

GuideAssignmentTest.java

Function name	Test case ID	Input	Expected output	Actual output
testConfirm	1	Confirm guide assignment	Status = CONFIRMED	Status set as expected
testStart	2	Start guide assignment	Status = IN_PROGRESS	Status as expected
testComplete	3	Complete guide assignment	Status = COMPLETED and completed date	Status matches, date assigned
testCancel	4	Cancel assignment with reason	Status = CANCELLED, reason recorded	Both data present
testReassign	5	Reassign with new guide info	Guide and status updated	Updated correctly
testIsActive	6	Set various statuses	Returns true for active statuses	Matches for all cases
testIsCompleted	7	Switch completion status	True/False as per status	Matches expected
testIsCancelled	8	Switch cancel status	True/False as per status	Matches expected
testCalculateTotalEarnings	9	Set different statuses and rates	Calculates earnings for COMPLETED only	Matches expected values
testGetDurationInHours	10	Assign start/end time, invalid input	Returns duration, 'Unknown', 'Not set'	Matches expected
testHasTimeConflictWith	11	Overlapping assignments	True/False depending on overlap	Correct per input

testGenerateAssignmentDetails	12	Assignment info	Details string includes all info	All info included
testGettersAndSetters	13	Set all fields, retrieve	All values match what was set	Correct for all fields
testToString	14	Call toString	Contains assignment info	Matches expected

GuideScheduleTest.java

Function name	T.C ID	Input	Expected output	Actual output
testIsAvailableForDate_InitiallyTrue	1	Query available date	True	True as expected
testIsAvailableForDate_MaxToursReached_False	2	Add max assignments, test avail	False	False as expected
testIsAvailableForTimeSlot_NoConflict_True	3	Assign and test new slot	True	True as expected
testIsAvailableForTimeSlot_Conflict_False	4	Assign overlapping slot	False	False as expected
testMarkUnavailableAndAvailable	5	Mark date unavailable and then available	Proper availability toggling	Status toggled as expected
testGetTotalAssignmentsForMonth	6	Add assignments & count	Count matches total	Value matches

testGettersAndSetters	7	Set all fields	All values are set and retrievable	As expected
testToStringContainsGuideName	8	toString on schedule	Info includes guide name	Name present
testAddingMoreThanMaxToursPerDay	9	Set max tours, exceed	Availability is false	Status as expected
testUtilizationRateCalculation	10	Assignments, utilization check	Rate within valid range (0-100)	Matches expected

HotelReviewTest.java

Function name	Test case ID	Input	Expected output	Actual output
testInitialValues	1	Init review object	All initial values set	All values correct
testSetDetailedRatings	2	Set ratings	Ratings averaged, recommend determined	Ratings and recommend correct
testMarkHelpful	3	Call helpful multiple times	Helpful count increments	Value increments
testVerifyUnverify	4	Verify / unverify review	Status toggled	Status as expected
testAnonymousReview	5	Make review anonymous	Summary includes 'Anonymous'	Summary contains term
testReviewCategories	6	Set different rating levels	Category matches rating	Category as expected

testStayDetailsAndDuration	7	Set stay details	Room and duration calculated	Values matched
testProsCons	8	Add pros and cons	Summary lists both	Both present
testTopLowestRatedAspect	9	Set detailed ratings	Identifies top and lowest aspect	Correct aspect names
testUpdateReviewResetsVerification	10	Update review and rating	Verification reset	Reset as expected
testIsDetailedReview	11	Set short/long comments	long: true, short: false	As expected
testRatingStars	12	Set ratings	Correct star string returned	String matches rating
testOverallSatisfactionScore	13	Set detailed ratings	Correct average satisfaction	Value correct
testEqualsAndHashCode	14	Compare equal/different objects	Equality and hashcode checks	Result correct
testToStringContainsImportantInfo	15	toString output	Contains key fields	Key data present
testEdgeCasesExtremeRatings	16	Ratings 0 or 5	Negative/positive flags	Correct flags
testNullComments	17	Set comments null	Length 0, not detailed	As expected

ItineraryTest.java

Function name	Test case ID	Input	Expected output	Actual output
testAddActivity	1	Add "Hiking"	Activity added to list	Activity present

testAddLocation	2	Add "Mountain"	Location added	Location present
testGetAndSetItineraryId	3	Set and get itineraryId	Updated value	Matches expected
testGetAndSetPackageId	4	Set and get packageId	Updated value	Matches expected
testGetAndSetDay	5	Set and get day	Updated value	Matches expected
testGetAndSetActivities	6	Set list of activities	Updated list	Matches expected
testGetAndSetLocationsVisited	7	Set list of locations	Updated list	Matches expected
testGetAndSetAccommodationDetails	8	Set accommodation details	Updated value	Matches expected
testGetAndSetMealPlan	9	Set meal plan	Updated value	Matches expected
testGetAndSetTransportDetails	10	Set transport details	Updated value	Matches expected
testGetAndSetStartTime	11	Set start time	Updated value	Matches expected
testGetAndSetEndTime	12	Set end time	Updated value	Matches expected
testGetAndSetSpecialInstructions	13	Set instruction	Updated value	Matches expected
testAddNullActivityAndLocation	14	Add null values	List includes null	Null present
testAddEmptyActivityAndLocation	15	Add empty string	List includes empty	Empty string present
testSetNullActivitiesAndLocations	16	Set activities/locations to null	Value set to null	Null value set

testSetEmptyActivitiesAndLocations	17	Set empty lists	Value set to empty	Empty list set
testDuplicateActivitiesAndLocations	18	Add duplicate entries	All duplicates reflected	Count matches added

PaymentTest.java

Function name	Test case ID	Input	Expected output	Actual output
testProcessPayment	1	Call processPayment	Status = COMPLETED	Status matches
testFailPayment	2	Call failPayment	Status = FAILED	Status matches
testRefundPayment	3	Call refundPayment	Status = REFUNDED	Status matches
testGettersAndSetters	4	Set all fields	All values get/set correctly	As expected
testToString	5	toString output	Contains key details	Key info present
testNullAndEmptyStrings	6	Set fields to null/empty	All values handled without exception	Values set accordingly
testNegativeAndZeroAmount	7	Set zero and negative amounts	Value matches input	As expected

testNegativeAndZeroExchangeRate	8	Zero and negative exchange	Value matches input	As expected
testInvalidPaymentStatus	9	Set unknown status	No crash, value matches	Status set as input
testProcessPaymentMultipleTimes	10	Multiple calls	Status remains COMPLETED	Status unchanged
testSetInvalidDateStrings	11	Set payment date to null/invalid	Value set and retrievable	Value matches

PaymentUITest.java

Function name	Test case ID	Input	Expected output	Actual output
testCardPaymentSuccess	1	Card option selected with valid card info	Payment processed, booking marked paid	Booking info and result shown
testInvalidCardPayment	2	Card option selected with invalid number	Payment failed, invalid card message	Error message displayed
testMobileBankingPaymentSuccess	3	Mobile banking, valid input	Payment processed, booking info displayed	Info shown on success
testBankTransferPayment	4	Bank transfer, trx ID provided	Details recorded, payment info shown	Info displayed
testCashPayment	5	Cash option selected	Cash payment option message shown	Correct info displayed

testPaymentCancelled	6	Cancel payment option	Payment not processed, cancellation message shown	Info message as expected

ReviewManagerTest.java

Function name	Test case ID	Input	Expected output	Actual output
testAddPackageReviewNormal	1	Add a normal package review	Correct review info	Review object matches
testAddPackageReviewCornerCases	2	Empty, null, negative rating	Accepts input, values reflected	All cases valid
testGetPackageReviewsByUserAndPackage	3	Add multiple reviews	Those added retrievable	Lists match expected
testGetAveragePackageRating	4	Add reviews and get average	Correct average calculated	Value matches
testRemoveAllReviewsForPackage	5	Add and remove reviews for package	Reviews removed	No reviews found
testAddHotelReviewNormal	6	Add normal hotel review	Correct review info	Review object matches
testHotelReviewsCornerCases	7	Null, empty, negative rating	Accepts input, values reflected	All cases valid
testGetHotelReviewsByUserAndHotel	8	Add and query hotel reviews	Correct counts	List matches
testGetAverageHotelRating	9	Add and average hotel ratings	Correct average	Value matches
testRemoveAllReviewsForHotel	10	Add and remove for hotel	Reviews removed	No reviews found

testRatingsAddAndGet	11	Add ratings for hotel and package	Can query ratings	List matches
testRemoveAllReviewsByUser	12	Remove all reviews by user	All lists become empty	All empty
testSearchPackageReviews	13	Search package reviews	Correct review found	List matches
testGetPackageRatingDistribution	14	Multiple ratings added	Distribution map correct	Map correct

SearchHelperTest.java

Function name	Test case ID	Input	Expected output	Actual output
testFilterNormal	1	Filter list, matches 'banana'	Only matched item returned	Correct output
testFilterEmptyList	2	Filter empty list	Returns empty list	Empty list
testFilterNoMatches	3	Filter with no match	Returns empty list	Empty list
testFilterListWithNullElements	4	List with null, filter	Matches only valid elements	Filtered correctly
testContainsIgnoreCase	5	Various string inputs	True/False based on case and content	As expected
testSearchInListNormal	6	List and search keyword	Only matched item returned	Correct output
testSearchInListNullOrEmptyKeyword	7	Keyword is null, empty or blank	Empty list returned	Correct output
testSearchInListWithNullElements	8	List with null, filter	Only matched item returned	Correct output

testMatchesAnyNormal	9	Text and multiple keywords	True if any match	Value matches
testMatchesAnyNullText	10	Null text	Always false	As expected
testMatchesAnyWithNullKeywordElement	11	Keyword is null	Always false	As expected
testMatchesAnyEmptyKeyword	12	Empty keyword	True	As expected

TransportTypeTest.java

Function name	Test case ID	Input	Expected output	Actual output
testValues	1	Get enum values	All enum values present	Array equals expected
testValueOf	2	ValueOf with valid names	Matches enum values	Values as expected
testGetDisplayName	3	Get display name for each enum	Correct display names	Names as expected
testGetBaseFare	4	Get base fare for each enum	Correct fares	Values as expected
testValueOfInvalidName	5	ValueOf with invalid name	Throws IllegalArgumentException	Exception thrown
testValueOfLowercaseName	6	ValueOf with lowercase string	Throws IllegalArgumentException	Exception thrown
testValueOfNull	7	ValueOf with null	Throws NullPointerException	Exception thrown
testDisplayNameNotNullOrEmpty	8	Check all types	All names are not null/empty	True for each
testBaseFareNonNegative	9	For all types, base fare >= 0	True for each	True for each
testEnumConsistency	10	Enum values count	Exactly 4 enums	Value matches

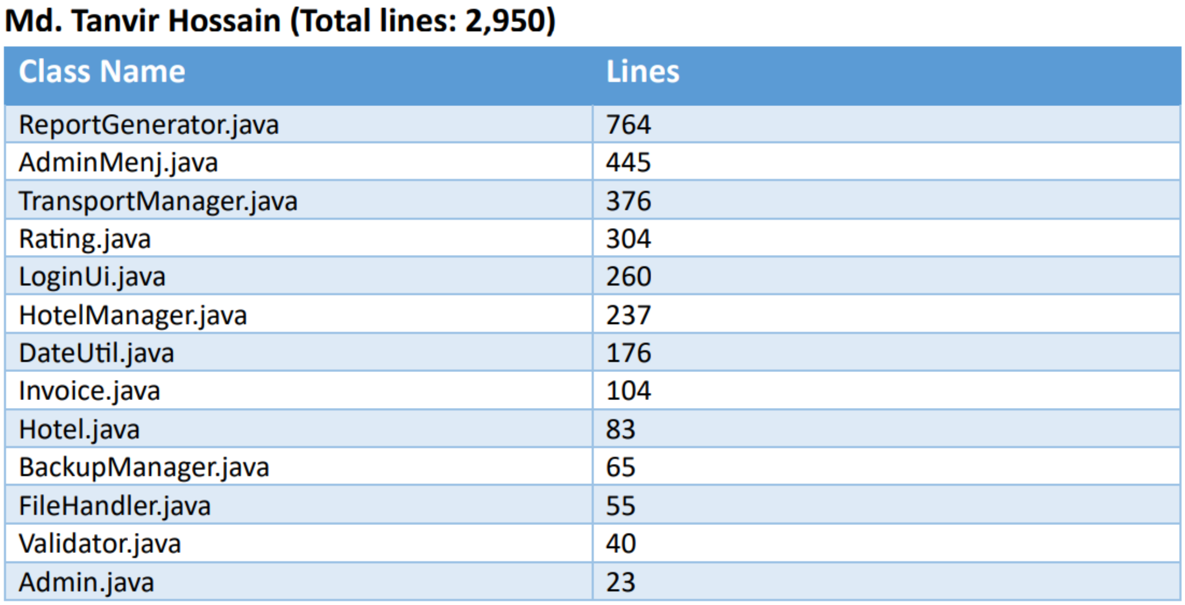
UserManagerTest.java

Function name	Test case ID	Input	Expected output	Actual output
testAddUser	1	Add a user	User count increases, user retrievable	User found
testRemoveUser	2	Remove user, also non-existent removal	User removed, no exception	Removals as expected
testDuplicateUserAdd	3	Add duplicate user	No extra user added, old user remains	Count and user correct
testUpdateUser	4	Update user's info	Updated details reflected	Data updated
testGetUserByIdAndEmail	5	Get from id/email	Correct user returned	Data retrieved
testBlockUnblockUser	6	Block and unblock user	Status toggles, reflected in user info	Status matches
testGetActiveInactiveUsers	7	After block/unblock	Correct counts for active/inactive	Count correct
testSearchUsers	8	Search by name/email	Correct matches found	List as expected
testGetUsersByRole	9	Query by role	Only users of correct type returned	List matches
testEmailExistsAndCounts	10	Check email presence, count	True/False and correct counts	Values match
testNullOrEmptyInputs	11	Query with null/empty	No user returned, handled gracefully	Values as expected
testUpdateNonExistentUser	12	Update non-existent user	No update, user still null	As expected
testMultipleUserTypes	13	Add tourist/agent	Correct types and total count	Users matched
testSearchCaseInsensitive	14	Search by case-insensitive name	Correct user found	List matches

“To see which classes are under which package, Click Here”

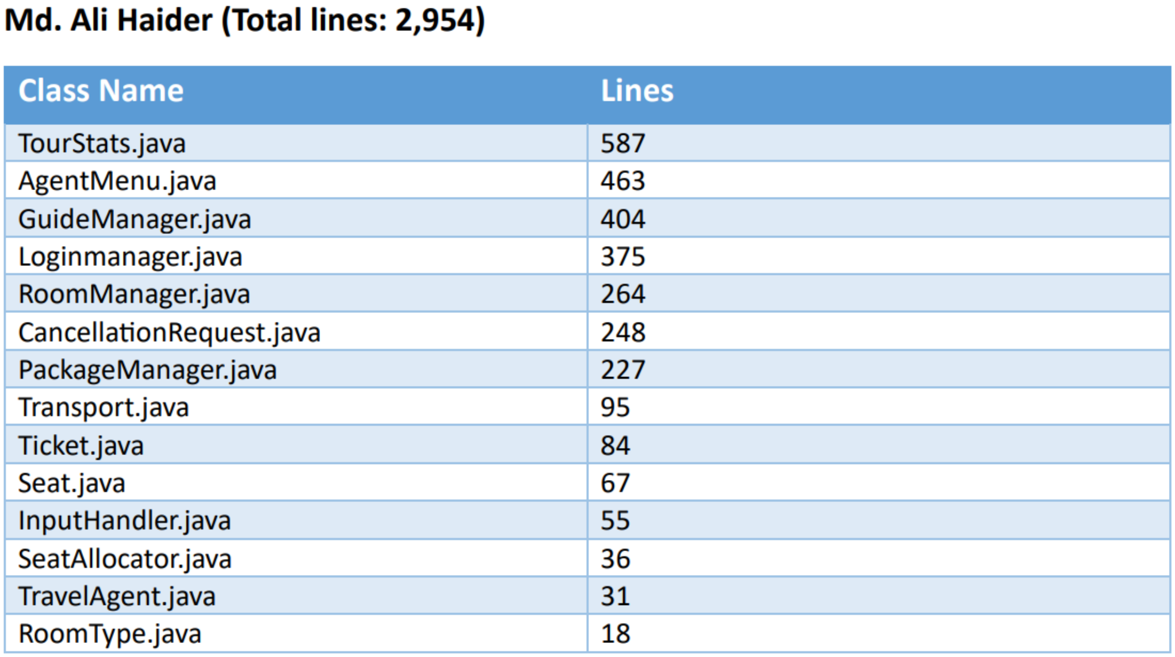
“For Source code and Testing click GitHub”

Package and Class Structure



Md. Tanvir Hossain (Total lines: 2,950)

Class Name	Lines
ReportGenerator.java	764
AdminMenj.java	445
TransportManager.java	376
Rating.java	304
LoginUi.java	260
HotelManager.java	237
DateUtil.java	176
Invoice.java	104
Hotel.java	83
BackupManager.java	65
FileHandler.java	55
Validator.java	40
Admin.java	23



Md. Ali Haider (Total lines: 2,954)

Class Name	Lines
TourStats.java	587
AgentMenu.java	463
GuideManager.java	404
Loginmanager.java	375
RoomManager.java	264
CancellationRequest.java	248
PackageManager.java	227
Transport.java	95
Ticket.java	84
Seat.java	67
InputHandler.java	55
SeatAllocator.java	36
TravelAgent.java	31
RoomType.java	18

Nabila Tabassum (Total lines: 2,950)

Class Name	Lines
ReviewManager.java	558
PaymentUi.java	483
BookingUi.java	423
HotelReview.java	351
GroupTour.java	296
GuideSchedule.java	241
GuideAssignment.java	191
UserManager.java	173
Payment.java	75
Itinerary.java	58
SearchHelper.Java	47
Currencyformatter.java	35
TransportType.java	19

Bushra Haque (Total lines: 2,949)

Class Name	Lines
PackageUi.java	549
TouristMenu.java	486
PdfExpprter.java	428
RefundManager.java	352
Logger.Java	267
BookingManager.java	255
PackageReview.java	235
ToueGuide.java	93
Booking.java	83
Vehicle.java	66
PriceCalculator.java	52
NotificationSender.java	45
BookingStatus.java	20
PackageCategory.java	18

Nirzona Binta Badal (Total lines: 2,948)

Class Name	Lines
TourPackage.java	535
Main.java	534
AuthService.java	400
PaymentManager.java	332
MainMenu.java	298
IDGenerator.Java	248
Schedule.java	213
DataSeeder.java	150
Room.java	76
Location.java	63
User.java	45
Tourist.java	35
TourType.java	19

Summary Table

Member	Total Classes	Total Lines
Md. Tanvir Hossain	14	2,950
Md. Ali Haider	14	2,954
Nabila Tabassum	13	2,950
Bushra Haque	14	2,949
Nirzona Binta Badal	13	2,948