Unit Testing Specification

UNIT TESTING SPECIFICATION

This document outlines each controller method suitable for unit testing, focusing on isolated logic validation.

AssetController

Methods for Unit Testing:

- `List<Asset> getAllAssets()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `List<Asset> getAssetsByCategory(@PathVariable String category)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `List<Asset> getAssetsByStatus(@PathVariable String status)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `List<Asset> searchAssetsByAssignedTo(@RequestParam String keyword)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `List<Asset> getAssetsByEmployeeDbId(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `Asset addAsset(@RequestBody Asset asset)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `Asset updateAsset(@PathVariable Long id, @RequestBody Asset asset)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `void deleteAsset(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

AttendanceController

Methods for Unit Testing:

- `ResponseEntity<List<AttendanceWithEmployeeDTO>> getAllAttendance()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<List<Attendance>> getEmployeeAttendance(@PathVariable String employeeId)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Attendance> markAttendance(@RequestBody Attendance attendanceRequest)`
- **Focus:** Test return value, mock dependencies, check input edge cases

Unit Testing Specification

AuthController

Methods for Unit Testing:

- `ResponseEntity<AuthResponse> register(@RequestBody RegisterRequest request)`
- **Focus:** Test return value, mock dependencies, check input edge cases

BankDocumentController

Methods for Unit Testing:

- `ResponseEntity<List<BankDocument>> getAllBankDocuments()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<BankDocument> getBankDocumentById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<BankDocument> createBankDocument(@RequestBody BankDocument bankDocument)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<BankDocument> updateBankDocument(@PathVariable Long id, @RequestBody BankDocument bankDocumentDetails)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deleteBankDocument(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

BillingController

Methods for Unit Testing:

- `ResponseEntity<List<Billing>> getAllBillings()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Billing> getBillingById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Billing> createBilling(@RequestBody Billing billing)`
- **Focus:** Test return value, mock dependencies, check input edge cases

- `ResponseEntity<Billing> updateBilling(@PathVariable Long id, @RequestBody Billing billingDetails)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deleteBilling(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

CaDocumentController

Methods for Unit Testing:

- `ResponseEntity<List<CaDocument>> getAllCaDocuments()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<CaDocument> getCaDocumentById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<CaDocument> createCaDocument(@RequestBody CaDocument caDocument)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<CaDocument> updateCaDocument(@PathVariable Long id, @RequestBody CaDocument caDocumentDetails)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deleteCaDocument(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

CommissionsController

Methods for Unit Testing:

- `ResponseEntity<CommissionsDTO> createCommission(@RequestBody CommissionsDTO commissionsDTO)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<List<CommissionsDTO>> getAllCommissions()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<CommissionsDTO> updateCommission(@PathVariable Long id, @RequestBody CommissionsDTO commissionsDTO)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deleteCommission(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

Unit Testing Specification

CompanyRegistrationController

Methods for Unit Testing:

- `ResponseEntity<List<CompanyRegistration>> getAllCompanyRegistrations()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<CompanyRegistration> getCompanyRegistrationById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<CompanyRegistration> createCompanyRegistration(@RequestBody CompanyRegistration companyRegistration)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<CompanyRegistration> updateCompanyRegistration(@PathVariable Long id, @RequestBody CompanyRegistration companyRegistrationDetails)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deleteCompanyRegistration(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

ElectricBillsController

Methods for Unit Testing:

- `ResponseEntity<ElectricBillsDTO> createElectricBill(@RequestBody ElectricBillsDTO electricBillsDTO)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<List<ElectricBillsDTO>> getAllElectricBills()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<ElectricBillsDTO> updateElectricBill(@PathVariable Long id, @RequestBody ElectricBillsDTO electricBillsDTO)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deleteElectricBill(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

EmployeeController

Methods for Unit Testing:

- `ResponseEntity<Employee> createEmployee(@RequestBody Employee employee)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Employee> getEmployeeById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Employee> getEmployeeByEmployeeId(@PathVariable String employeeId)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<List<Employee>> getAllEmployees()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Employee> updateEmployee(@PathVariable Long id, @RequestBody Employee employeeDetails)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<HttpStatus> deleteEmployee(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Employee> updateEmployeeStatus(@PathVariable Long id, @RequestParam String status)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Employee> registerEmployee(@RequestBody com.example.storemanagementbackend.dto.RegisterRequest request)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Employee> hrRegisterEmployee(@RequestBody EmployeeRegistrationRequest request)`
- **Focus:** Test return value, mock dependencies, check input edge cases

EmployeeDocumentController

Methods for Unit Testing:

- `ResponseEntity<List<EmployeeDocumentDTO>> getAllDocuments()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<List<EmployeeDocumentDTO>> getDocumentsByEmployeeId(@PathVariable String employeeId)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<List<EmployeeDocumentDTO>> getDocumentsByType(@PathVariable String documentType)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<EmployeeDocumentDTO> getDocumentById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Resource> downloadDocument(@PathVariable String employeeId, @PathVariable String docType, HttpServletRequest request)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<HttpStatus> deleteDocument(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

Unit Testing Specification

ExpoAdvertisementController

Methods for Unit Testing:

- `ResponseEntity<ExpoAdvertisementDTO> createExpoAdvertisement(@RequestBody ExpoAdvertisementDTO expoAdvertisementDTO)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<List<ExpoAdvertisementDTO>> getAllExpoAdvertisements()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<ExpoAdvertisementDTO> updateExpoAdvertisement(@PathVariable Long id, @RequestBody ExpoAdvertisementDTO expoAdvertisementDTO)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deleteExpoAdvertisement(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

FinanceReportController

Methods for Unit Testing:

- `ResponseEntity<List<FinanceReport>> getAllFinanceReports()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<FinanceReport> getFinanceReportById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<FinanceReport> createFinanceReport(@RequestBody FinanceReport financeReport)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<FinanceReport> updateFinanceReport(@PathVariable Long id, @RequestBody FinanceReport financeReportDetails)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deleteFinanceReport(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

FixedStationaryController

Unit Testing Specification

Methods for Unit Testing:

- `List<FixedStationary> getAll()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<FixedStationary> getById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `FixedStationary create(@RequestBody FixedStationary stationary)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<FixedStationary> update(@PathVariable Long id, @RequestBody FixedStationary stationary)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> delete(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

FurnitureController

Methods for Unit Testing:

- `List<Furniture> getAll()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Furniture> getById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `Furniture create(@RequestBody Furniture furniture)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Furniture> update(@PathVariable Long id, @RequestBody Furniture furniture)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> delete(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

IncentivesController

Methods for Unit Testing:

- `ResponseEntity<IncentivesDTO> createIncentive(@RequestBody IncentivesDTO incentivesDTO)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<List<IncentivesDTO>> getAllIncentives()`
- **Focus:** Test return value, mock dependencies, check input edge cases

- `ResponseEntity<IncentivesDTO> updateIncentive(@PathVariable Long id, @RequestBody IncentivesDTO incentivesDTO)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deleteIncentive(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

InternetBillsController

Methods for Unit Testing:

- `ResponseEntity<InternetBillsDTO> createInternetBill(@RequestBody InternetBillsDTO internetBillsDTO)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<List<InternetBillsDTO>> getAllInternetBills()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<InternetBillsDTO> updateInternetBill(@PathVariable Long id, @RequestBody InternetBillsDTO internetBillsDTO)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deleteInternetBill(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

LabComponentController

Methods for Unit Testing:

- `List<LabComponent> getAll()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<LabComponent> getById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `LabComponent create(@RequestBody LabComponent component)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<LabComponent> update(@PathVariable Long id, @RequestBody LabComponent component)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> delete(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

Unit Testing Specification

LabInstrumentController

Methods for Unit Testing:

- `List<LabInstrument> getAll()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<LabInstrument> getById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `LabInstrument create(@RequestBody LabInstrument instrument)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<LabInstrument> update(@PathVariable Long id, @RequestBody LabInstrument instrument)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> delete(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

LabInventoryTransactionController

Methods for Unit Testing:

- `List<LabInventoryTransaction> getAll()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<LabInventoryTransaction> getById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `LabInventoryTransaction create(@RequestBody LabInventoryTransaction transaction)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<LabInventoryTransaction> update(@PathVariable Long id, @RequestBody LabInventoryTransaction transaction)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> delete(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

LabMaterialController

Methods for Unit Testing:

Unit Testing Specification

- `List<LabMaterial> getAll()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<LabMaterial> getById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `LabMaterial create(@RequestBody LabMaterial material)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<LabMaterial> update(@PathVariable Long id, @RequestBody LabMaterial material)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> delete(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

LeaveController

Methods for Unit Testing:

- `ResponseEntity<List<Leave>> getAllLeaves()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Leave> createLeave(@RequestBody Leave leave)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deleteLeave(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

LeaveRequestController

Methods for Unit Testing:

- `ResponseEntity<LeaveRequestDTO> submitLeaveRequest(@RequestBody LeaveRequestDTO dto)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<List<LeaveRequestDTO>> getLeaveRequestsByEmployee(@PathVariable String employeeId)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<List<LeaveRequestDTO>> getAllLeaveRequests()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<List<LeaveRequestDTO>> getLeaveRequestsByStatus(@PathVariable String status)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<List<LeaveRequestDTO>> getNonApprovedLeaveRequests()`

Unit Testing Specification

- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deleteLeaveRequest(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `List<LeaveRequestDTO> getAllHolidays()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `LeaveRequestDTO getHolidayById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `LeaveRequestDTO addHoliday(@RequestBody LeaveRequestDTO dto)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `LeaveRequestDTO updateHoliday(@PathVariable Long id, @RequestBody LeaveRequestDTO dto)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `void deleteHoliday(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

LogisticsDocumentController

Methods for Unit Testing:

- `ResponseEntity<List<LogisticsDocument>> getAllLogisticsDocuments()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<LogisticsDocument> getLogisticsDocumentById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<LogisticsDocument> createLogisticsDocument(@RequestBody LogisticsDocument logisticsDocument)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<LogisticsDocument> updateLogisticsDocument(@PathVariable Long id, @RequestBody LogisticsDocument logisticsDocumentDetails)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deleteLogisticsDocument(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

PerformanceController

Methods for Unit Testing:

- `ResponseEntity<List<Performance>> getAllPerformances()`

Unit Testing Specification

- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Performance> getPerformanceById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Performance> createPerformance(@RequestBody Performance performance)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Performance> updatePerformance(@PathVariable Long id, @RequestBody Performance performanceDetails)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deletePerformance(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

PerformanceReviewController

Methods for Unit Testing:

- `ResponseEntity<PerformanceReview> createPerformanceReview(@RequestBody PerformanceReviewDTO performanceReviewDTO)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<PerformanceReview> getPerformanceReviewById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<List<PerformanceReview>> getAllPerformanceReviews()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<List<PerformanceReview>> getPerformanceReviewsByEmployeeId(@PathVariable String employeeId)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<List<PerformanceReview>> getPerformanceReviewsByEmployeeDbId(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<PerformanceReview> updatePerformanceReview(@PathVariable Long id, @RequestBody PerformanceReview performanceReviewDetails)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<HttpStatus> deletePerformanceReview(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

PrinterController

Methods for Unit Testing:

Unit Testing Specification

- `List<Printer> getAll()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Printer> getById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `Printer create(@RequestBody Printer printer)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Printer> update(@PathVariable Long id, @RequestBody Printer printer)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> delete(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

PurchaseController

Methods for Unit Testing:

- `ResponseEntity<List<Purchase>> getAllPurchases()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Purchase> getPurchaseById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Purchase> createPurchase(@RequestBody Purchase purchase)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Purchase> updatePurchase(@PathVariable Long id, @RequestBody Purchase purchaseDetails)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deletePurchase(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

RegularStationaryController

Methods for Unit Testing:

- `List<RegularStationary> getAll()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<RegularStationary> getById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `RegularStationary create(@RequestBody RegularStationary stationary)`

Unit Testing Specification

- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<RegularStationary> update(@PathVariable Long id, @RequestBody RegularStationary stationary)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> delete(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

RentController

Methods for Unit Testing:

- `ResponseEntity<RentDTO> createRent(@RequestBody RentDTO rentDTO)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<List<RentDTO>> getAllRents()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<RentDTO> updateRent(@PathVariable Long id, @RequestBody RentDTO rentDTO)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deleteRent(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

ReportController

Methods for Unit Testing:

- `ResponseEntity<Report> getReportById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Report> createReport(@RequestBody Report report)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Report> updateReport(@PathVariable Long id, @RequestBody Report reportDetails)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deleteReport(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

SalariesController

Unit Testing Specification

Methods for Unit Testing:

- `ResponseEntity<SalariesDTO> createSalary(@RequestBody SalariesDTO salariesDTO)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<List<SalariesDTO>> getAllSalaries()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<SalariesDTO> updateSalary(@PathVariable Long id, @RequestBody SalariesDTO salariesDTO)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deleteSalary(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

SaleController

Methods for Unit Testing:

- `ResponseEntity<List<Sale>> getAllSales()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Sale> getSaleById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Sale> createSale(@RequestBody Sale sale)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Sale> updateSale(@PathVariable Long id, @RequestBody Sale saleDetails)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deleteSale(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

SIMBillsController

Methods for Unit Testing:

- `ResponseEntity<SIMBillsDTO> createSIMBillExpense(@RequestBody SIMBillsDTO simBillsDTO)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<List<SIMBillsDTO>> getSIMBillExpenses()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<SIMBillsDTO> updateSIMBillExpense(@PathVariable Long id, @RequestBody SIMBillsDTO

simBillsDTO)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deleteSIMBillExpense(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

StaffMemberController

Methods for Unit Testing:

- `List<StaffMember> getAllStaff()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<StaffMember> getStaffById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `StaffMember createStaff(@RequestBody StaffMember staffMember)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<StaffMember> updateStaff(@PathVariable Long id, @RequestBody StaffMember updatedStaff)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `void deleteStaff(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

StationaryInventoryTransactionController

Methods for Unit Testing:

- `List<StationaryInventoryTransaction> getAll()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<StationaryInventoryTransaction> getById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `StationaryInventoryTransaction create(@RequestBody StationaryInventoryTransaction transaction)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<StationaryInventoryTransaction> update(@PathVariable Long id, @RequestBody StationaryInventoryTransaction transaction)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> delete(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

Unit Testing Specification

SystemController

Methods for Unit Testing:

- `List<System> getAll()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<System> getById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `System create(@RequestBody System system)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<System> update(@PathVariable Long id, @RequestBody System system)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> delete(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

TenderController

Methods for Unit Testing:

- `ResponseEntity<List<Tender>> getAllTenders()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Tender> getTenderById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Tender> createTender(@RequestBody Tender tender)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Tender> updateTender(@PathVariable Long id, @RequestBody Tender tenderDetails)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deleteTender(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

TravelController

Methods for Unit Testing:

Unit Testing Specification

- `ResponseEntity<TravelDTO> createTravel(@RequestBody TravelDTO travelDTO)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<List<TravelDTO>> getAllTravels()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<TravelDTO> updateTravel(@PathVariable Long id, @RequestBody TravelDTO travelDTO)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deleteTravel(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

WaterBillsController

Methods for Unit Testing:

- `ResponseEntity<WaterBillsDTO> createWaterBillExpense(@RequestBody WaterBillsDTO request)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<List<WaterBillsDTO>> getWaterBillExpenses()`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<WaterBillsDTO> updateWaterBillExpense(@PathVariable Long id, @RequestBody WaterBillsDTO request)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<Void> deleteWaterBillExpense(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases

WeeklyActivityController

Methods for Unit Testing:

- `ResponseEntity<WeeklyActivity> createWeeklyActivity(@RequestBody WeeklyActivity weeklyActivity)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<WeeklyActivity> getWeeklyActivityById(@PathVariable Long id)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<WeeklyActivity> updateWeeklyActivity(@PathVariable Long id, @RequestBody WeeklyActivity weeklyActivityDetails)`
- **Focus:** Test return value, mock dependencies, check input edge cases
- `ResponseEntity<HttpStatus> deleteWeeklyActivity(@PathVariable Long id)`

Unit Testing Specification

- **Focus:** Test return value, mock dependencies, check input edge cases