Requirement ID	Requirement Description	UML reference	Implementation module	Test case id
R1	Input field for entering waypoints (latitude, longitude, altitude).	enterWaypoint method, waypoints and waypointInputField attributes in FlightPlanManagement class in UML class diagram.		
R2	Input fields for speed restrictions and expected times of arrival at each waypoint.	enterSpeedRestriction and enterArrivalTime method and speedInputField and timeInputField attributes in FlightPlanManagement class in UML class diagram		
R3	A submit button to load and activate the flight plan.	loadFlightPlan method and submitButton attribute in FlightPlanManagement class in UML class diagram		
R4	Visual display (map) showing current position, planned route, and waypoints.	displayMap method in FlightPlanManagement class in UML class diagram		
R5	Engage autopilot when at cruising altitude, and disengage it for landing.	engageAutoPilot and disengageAutoPilot method in AutoPilotControlPanel class in UML class diagram		
R6	Manually override autopilot.	engageAutoPilot and disengageAutoPilot method in AutoPilotControlPanel class in UML class diagram		

R7	Respond quickly and appropriately to any hazards.			
R8	Indicator lights for autopilot status (engaged, disengaged, fault condition).	displayIndicatorLights method and autoPilotStatus attribute in AutoPilotControlPanel class in UML class diagram		
R9	Controls for manual override: altitude adjustment, speed, heading.	adjustAltitude, adjustSpeed, adjustHeading method in AutoPilotControlPanel class in UML class diagram		
R10	Digital readouts for airspeed, altitude, pitch, roll, yaw, and engine parameters.	displayAirspeed, displayAltitude,displayPitch, displayRoll, displayYaw, displayEngineParameters method and airspeed, altitude,pitch,roll,yaw and engineParams attributes in SensorDataDisplay class in UML class diagram		
R11	Visual indicators for data update frequency (e.g., colour change or blinking to indicate fresh data).	updateAirspeed, updateAltitude, updatePitch, updateRoll,updateYaw, updateEngineParameters,displayIndicator, colorChange method in SensorDataDisplay class in UML class diagram		
R12	Interface should be responsive to pilot input and			

	button presses.			
R13	The UI should have a dedicated section for hazard warnings and mitigation actions – these actions must be providing as soon as possible after the initial hazard warning has been raised – ideally concurrently. These will be in the form of audible and visual alerts for immediate hazards, and a checklist or action plan for emergency procedures.	Hazard alert class in UML alongside all methods and attributes of that class.		
R14	Attitude Sensor should measure pitch (nose up/down), roll (wing	AttitudeSensor class in UML with the getters and setter methods within the class.		

	up/down), and yaw (nose left/right).			
R15	The data format for orientation data should be of degrees from the horizon for pitch and roll and magnetic heading for yaw with an update frequency of every 500 milliseconds.	AttitudeSensor class in UML with update frequency attribute.		
R16	Operational ranges of orientation data: Pitch: -30° to 30°, Roll: -60° to 60°, Yaw: -180° to 180°. Identify and manage exceedances through system alerts and corrective actions.	checkValidOreintationData method in AttitudeSensor class in UML.		
R17	Altitude Sensor should provide	Altitudesensor class in UML with the getters and setter methods of that class.		

	barometric and gps data.			
R18	Altitude sensor data should be in the format of AMSL with update frequency of every 500 milliseconds.	AltitudeSensor class in UML with update frequency attribute.		
R19	Altitude sensor should be capable of measuring altitude within -1000 to 50000 feet AMSL.	AltitudeSensor class in UML with checkAltitudeWithinRange method with attributes max_altitude and min_altitude.		
R20	The airspeed sensor should be able represent the aircraft's speed relative to the surrounding air and be in knots.	getCurrentAirspeed, setCurrentAirspeed method and currentAirspeed attribute in airspeed sensor class UML.		
R21	The airspeed sensor should also be able to represent aircraft	Updatefrequency attribute in AirspeedSensor class and checkairspeed method.		

	speed between 50 and 500 knots and have update frequency of every second.			
R22	Engine parameters should measure engine thrust and fuel flow in data format of thrust in pounds-force (lbf) and should have update frequency of every second.	setFuelFlow, getFuelFlow, getCurrentThrust, setCurrentThrust method and updatefrequency attribute in Engine class UML.		
R23	The system must prioritize safety with redundancy included for sensors.	3 to 1 relationship from the sensors class to sensordatadisplay and sensordata class in UML, alongside ids for each sensor. As well as with core system and sensor class.		
R24	The system shall handle sensor errors and unexpected events without crashing.	ReportError method in sensors and handlesystemerror in coresystem class for UML.		

R25	Hazard Mitigation Strategies: Implement fault detection, fault tolerance, and fail-safe mechanisms to handle abnormal sensor values, ensuring the system can safely manage potential hazards without catastrophic failures.	detectFaults,ensureFailSafeMechanisms, implementFaultTolerance in coresystem class.		
R26	Sensors including airspeed sensor, altitude sensor, and attitude sensor should operate in a 2oo3 architecture, providing backup in the case of a sensor failing.	3 to 1 relationship from the sensors class to sensordatadisplay and sensordata class in UML, alongside ids for each sensor. As well as with core system and sensor class.		

R27	Execution Check Parameters: After sending a control signal, the system must verify the execution by reading back the relevant sensor data.	Sendcontrolsignals and receivesensordata methods in autopilotsystem. sendSensorData methods for sensors and engine parameters.		
R28	For the Autopilot Control Frequency, the system should send control signals to the aircraft's control surfaces (elevators, ailerons, rudders) and engine control systems.	Sendcontrolsignals method in autopilotsystem.		
R29	Success Criteria: A control signal is considered successfully executed if the sensor data reflects the	verifyExecution method in autopilotsystem class.		

	expected change within a margin of error of ±2% for the control surfaces and ±5% for engine parameters, within 1 second of command issuance.			
R30	Failure Handling: If the execution check fails after sending a control signal, the system must attempt to resend the command up to three times before alerting the pilot to the issue via the user interface.	verifyExecution, resendControlSignal, and alertUserInterface method in autopilotsystem class.		
R31	The system should dynamically adjust engine thrust proportional to airspeed.	setCurrentThrust(double newThrust) in engine class		
R32	System should	setCurrentThrust(double newThrust) in engine class		

	accurately model the effects of thrust adjustments on altitude changes			
R33	System should model how changes in thrust affect the aircraft's pitch, and in scenarios of asymmetric thrust, its roll and yaw.	setCurrentThrust(double newThrust) in engine class		
R34	The maximum thrust will be 374 kN per engine.	MAX_THRUST attribute in engine		
R35	The minimum thrust will be 10kN per engine.	MIN_THRUST attribute in engine		
R36	The cruise thrust will be 56.1kN per engine.	currentThrust attribute in engine		
R37	Integration of autopilot system with sensors to gather data.	Autopilot system->controlsurface->sensordata Sensors->sensordata->controlsurface->autopilotsystem relationship in UML		

R38	Autopilot system should incorporate redundancy in the case of software bugs.	1 to many relationship between AutoPilotControlPanel and AutoPilotSystem.		
R39	The system shall ensure continuous and reliable communication between the autopilot system and critical components, such as the engine control system, to prevent loss of control over throttle settings or other vital parameters.	The bidirectional relationship of Autopilotsystem engine and control system helps ensure the reliable communication alongside methods like sendControlSignals and sendSensorData.		