

| | | |
|--|--|--|
| <p>«abstract»</p> <p>User</p> <pre># userId: String # email: String # passwordHash: String # createdAt: DateTime # lastLogin: DateTime</pre> <pre>+ login(email, pass): Token + logout(): void + updateProfile(data): Bool # validateCreds(pass): Bool</pre> | <p>RefundDispute</p> <pre>- disputId: String - orderId: String - status: Status - createdAt: DateTime - trustScore: Float - assignedAgent: Agent</pre> <pre>+ updateStatus(stat): void + getEvidence(): List<Evid> + calculateAge(): Duration + assignToAgent(agent): void</pre> | <p>AIsemanticChecker</p> <pre>- endpoint: String - apiKey: String - timeout: Integer</pre> <pre>+ analyzeConsistency(img, txt): Float + sendToGemini(load): AIResp - buildPrompt(img, claim): String</pre> |
| <p>RegisteredUser</p> <pre>- disputeHistory: List<Dispute> - trustLevel: Float - phoneNumber: String</pre> <pre>+ initiateDispute(order): Dispute + uploadEvidence(img, meta): Evid + viewStatus(id): Status + getNotifications(): List<Notif></pre> | <p>Evidence</p> <pre>- evidencId: String - imageData: Bytes - imageHash: String - metaScore: Float - elaScore: Float - aiScore: Float</pre> <pre>+ generateHash(): String + isLiveCapture(): Bool + getScores(): ScoreTriple</pre> | <p>TrustScoreCalculator</p> <pre>- weights: ScoreWeights - autoRefundThresh: Float - fraudAlertThresh: Float</pre> <pre>+ calcTrustScore(m, e, a): Float + setWeights(w1, w2, w3): void + getAction(score): Type</pre> |
| <p>SupportAgent</p> <pre>- assignedDisputes: List<Dispute> - resolvedCount: Integer - department: String</pre> <pre>+ reviewFraudAlert(id): void + requestEvidence(id): void + makeDecision(id, dec): Bool + viewPendingCases(): List<Dispute></pre> | <p>ForensicEngine</p> <pre>- config: AnalysisConfig</pre> <pre>+ analyze(evid): Result + runFullScan(img): Tuple - orchestrate(evid): void</pre> | <p>DecisionRouter</p> <pre>- api: RefundAPI - notify: NotificationService</pre> <pre>+ routeDecision(disp, score): void + processAutoRefund(d): Bool + createFraudAlert(d): Alert</pre> |
| <p>SystemAdmin</p> <pre>- permissions: List<String> - adminLevel: Integer</pre> <pre>+ updateThresholds(w): Bool + viewAnalytics(): Report + configureSystem(set): Bool + genReport(): Report</pre> | <p>MetadataAnalyzer</p> <pre>- suspiciousSoft: List<String> - requiredFields: List<String></pre> <pre>+ extractEXIF(img): Exif + detectSignatures(exif): Bool + validateSource(exif): Float</pre> | <p>«interface»</p> <p>RefundAPI</p> <pre>- endpoint: String - token: String</pre> <pre>+ initiateRefund(id, amt): Resp + checkStatus(id): Status + cancelRefund(id): Bool</pre> |
| | <p>ELAProcessor</p> <pre>- quality: Int = 90 - ampFactor: Int = 50 - varianceThresh: Float</pre> <pre>+ performELA(img): ELAResult + calcPixelDiff(a, b): Matrix + computeVariance(amp): Float</pre> | <p>«interface»</p> <p>NotificationService</p> <pre>- emailProvider: String - smsProvider: String</pre> <pre>+ sendEmail(to, sub, body): Bool + sendSMS(num, msg): Bool + sendPush(id, msg): Bool</pre> |