### 1. \*\*Core Entity: Person and its Subclasses\*\*

#### \*\*Person\*\* \*(Superclass)\*

- \*\*Attributes\*\*:

- `- id: String`

- `- name: String`

- `- contactInfo: ContactInfo` \*(Composition)\*

- `- address: String`

- `# role: String`

- \*\*Methods\*\*:

- `+ viewPersonalDetails(): void`

- `+ updateContactInfo(newContactInfo: ContactInfo): void`

Classes inheriting from `Person`:

- \*\*Administrator\*\*

- \*\*Doctor\*\*

- \*\*Nurse\*\*

- \*\*Receptionist\*\*

- \*\*Patient\*\*

- \*\*LabTechnician\*\*

- \*\*Pharmacist\*\*

- \*\*Radiologist\*\*

- \*\*Surgeon\*\*

- \*\*EmergencyResponseTeam\*\*

- \*\*InsuranceCoordinator`

#### \*\*ContactInfo\*\* \*(Composition within Person)\*

- \*\*Attributes\*\*:

- `- email: String`

- `- phone: String`

- \*\*Methods\*\*:

- `+ updateEmail(newEmail: String): void`

- `+ updatePhone(newPhone: String): void`

### 2. \*\*Roles and Specialized Functionalities\*\*

#### \*\*Administrator\*\*

- \*\*Attributes\*\*:

- Inherits from `Person`

- \*\*Methods\*\*:

- `+ manageUserAccounts(): void`

- `+ assignRolesAndPermissions(): void`

- `+ configureSystemSettings(): void`

- `+ generateReports(): void`

- `+ manageDroneInventory(): void`

- `+ configureDroneRoutes(): void`

#### \*\*Doctor\*\*

- \*\*Attributes\*\*:

- Inherits from `Person`

- \*\*Methods\*\*:

- `+ createPrescription(patient: Patient): Prescription`

- `+ updatePatientStatus(record: MedicalRecord): void`

- `+ scheduleAppointment(patient: Patient): void`

- `+ requestUrgentDelivery(request: DeliveryRequest): void`

#### \*\*Nurse\*\*

- \*\*Attributes\*\*:

- Inherits from `Person`

- \*\*Methods\*\*:

- `+ monitorVitals(record: MedicalRecord, vitals: VitalSigns): void`

- `+ administerMedication(medication: Medication, patient: Patient): void`

- `+ updatePatientStatus(record: MedicalRecord): void`

#### \*\*Receptionist\*\*

- \*\*Attributes\*\*:

- Inherits from `Person`

- \*\*Methods\*\*:

- `+ registerNewPatient(details: PatientDetails): Patient`

- `+ scheduleAppointment(patient: Patient, doctor: Doctor): void`

- `+ managePatientQueries(query: String): String`

#### \*\*Patient\*\*

- \*\*Attributes\*\*:

- Inherits from `Person`

- `- insuranceDetails: InsuranceDetails`

- \*\*Methods\*\*:

- `+ viewAppointmentSchedule(): void`

- `+ accessMedicalRecords(): MedicalRecord`

- `+ requestPrescriptionRefill(): void`

- `+ trackDeliveryStatus(deliveryId: String): String`

#### \*\*LabTechnician\*\*

- \*\*Attributes\*\*:

- Inherits from `Person`

- \*\*Methods\*\*:

- `+ conductLabTest(testRequest: LabTestRequest): LabTest`

- `+ enterLabResults(testId: String, results: String): void`

- `+ requestSampleDelivery(sample: LabSample): void`

#### \*\*Pharmacist\*\*

- \*\*Attributes\*\*:

- Inherits from `Person`

- \*\*Methods\*\*:

- `+ manageMedicationInventory(): void`

- `+ prepareDroneDelivery(request: DeliveryRequest): void`

- `+ updatePatientMedicationRecord(patient: Patient, prescription: Prescription): void`

#### \*\*Radiologist\*\*

- \*\*Attributes\*\*:

- Inherits from `Person`

- \*\*Methods\*\*:

- `+ conductImagingTest(testRequest: LabTestRequest): LabTest`

- `+ interpretTestResults(testId: String, interpretation: String): void`

#### \*\*Surgeon\*\*

- \*\*Attributes\*\*:

- Inherits from `Doctor`

- \*\*Methods\*\*:

- `+ scheduleSurgery(patient: Patient, date: Date): void`

- `+ conductSurgery(patient: Patient): void`

- `+ updatePatientStatusAfterSurgery(record: MedicalRecord): void`

#### \*\*InsuranceCoordinator\*\*

- \*\*Attributes\*\*:

- Inherits from `Person`

- `- insuranceProvider: InsuranceProvider`

- \*\*Methods\*\*:

- `+ integrateWithInsuranceProviders(): void`

- `+ manageClaims(patient: Patient, provider: InsuranceProvider): void`

### 3. \*\*Core System Components and Their Relationships\*\*

#### \*\*Hospital\*\*

- \*\*Attributes\*\*:

- `- name: String`

- `- departments: List<Department>` \*(Aggregation)\*

- `- rooms: List<Room>` \*(Composition)\*

- \*\*Methods\*\*:

- `+ addDepartment(department: Department): void`

- `+ assignRoom(room: Room): void`

#### \*\*Department\*\*

- \*\*Attributes\*\*:

- `- name: String`

- `- staff: List<MedicalStaff>` \*(Aggregation)\*

- \*\*Methods\*\*:

- `+ addStaff(staff: MedicalStaff): void`

#### \*\*MedicalRecord\*\*

- \*\*Attributes\*\*:

- `- patientId: String`

- `- appointments: List<Appointment>` \*(Aggregation)\*

- `- prescriptions: List<Prescription>` \*(Aggregation)\*

- `- labTests: List<LabTest>` \*(Aggregation)\*

- \*\*Methods\*\*:

- `+ addAppointment(appointment: Appointment): void`

- `+ addPrescription(prescription: Prescription): void`

- `+ addLabTest(labTest: LabTest): void`

#### \*\*VitalSigns\*\* \*(Composition within MedicalRecord)\*

- \*\*Attributes\*\*:

- `- temperature: Double`

- `- bloodPressure: String`

- `- heartRate: Int`

- \*\*Methods\*\*:

- `+ recordVitals(temperature: Double, bloodPressure: String, heartRate: Int): void`

### 4. \*\*Drone Integration Components\*\*

#### \*\*DroneSystem\*\*

- \*\*Attributes\*\*:

- `- id: String`

- `- status: String` \*(e.g., “Available” or “In Transit”)\*

- `- batteryLevel: Int`

- `- currentLocation: Location`

- `- currentRoute: Route`

- \*\*Methods\*\*:

- `+ dispatchDrone(request: DeliveryRequest): void`

- `+ monitorBattery(): void`

- `+ updateRoute(route: Route): void`

- `+ sendStatusUpdate(): void`

#### \*\*DeliveryRequest\*\*

- \*\*Attributes\*\*:

- `- requestId: String`

- `- type: String` \*(e.g., “Medication”, “Lab Sample”)\*

- `- pickupLocation: Location`

- `- deliveryLocation: Location`

- `- requestDate: Date`

- \*\*Methods\*\*:

- `+ initializeRequest(type: String, pickupLocation: Location, deliveryLocation: Location): void`

#### \*\*Location\*\* \*(Used in Routing and Delivery)\*

- \*\*Attributes\*\*:

- `- latitude: Double`

- `- longitude: Double`

- \*\*Methods\*\*:

- `+ calculateDistance(otherLocation: Location): Double`

### 5. \*\*Associations, Aggregations, and Compositions\*\*

#### Key Relationships:

- \*\*Doctor -> Appointment (1:\*):\*\* Each doctor can have multiple appointments.

- \*\*Doctor -> Prescription (1:\*):\*\* A doctor can write multiple prescriptions.

- \*\*Patient -> Appointment (1:\*):\*\* Each patient can have multiple appointments.

- \*\*Patient -> MedicalRecord (1:1):\*\* Each patient has a unique medical record.

- \*\*Hospital -> Department (1:\*):\*\* Each hospital has multiple departments.

- \*\*Department -> MedicalStaff (1:\*):\*\* Each department has a collection of medical staff.

#### Aggregations and Compositions:

- \*\*MedicalRecord Aggregates Appointment, Prescription, and LabTest\*\*: Medical records form a comprehensive patient profile, encompassing appointments, prescriptions, and lab tests.

- \*\*Hospital Compositions Room\*\*: Rooms are essential parts of the hospital and cannot exist independently.

- \*\*Inventory Aggregates Medication and Equipment\*\*: The inventory manages the hospital’s supplies, including medications and equipment.