**Law enforcement mangement system**

The software serves as a vital tool in modernizing crime reporting and law enforcement operations, offering citizens a convenient platform to report crimes from anywhere. Through the software, citizens can swiftly submit detailed reports, including essential information such as the date, time, and location of the incident. This streamlined reporting process enables law enforcement agencies to respond promptly to reported crimes, enhancing public safety and facilitating timely interventions. Additionally, the software assists police stations in managing their resources effectively by automating tasks such as vehicle and equipment allocation. The software categorizes officers into different roles, each with varying levels of access tailored to their responsibilities within law enforcement. For example, constables may have access to basic functionalities such as filing reports and responding to calls, while higher-ranking officers like inspectors or superintendents may have access to more extensive features, including case management and resource allocation. This tiered access ensures that officers can perform their duties efficiently while maintaining data security and integrity. Furthermore, the software facilitates the scheduling and management of patrol times through a user-friendly interface accessible to authorized personnel. Supervisors can define patrol schedules based on factors like crime hotspots, community needs, and available resources. Officers can view their assigned patrol times, update their schedules as needed, and report their activities in the software. This enables law enforcement agencies to optimize patrol coverage, allocate resources effectively, and respond promptly to emerging incidents, thereby enhancing public safety and reducing crime rates. Additionally, supervisors can monitor patrol activities, track officer locations, and analyze patrol data to identify trends and allocate resources strategically, further improving law enforcement outcomes.

**1. User Management:**

Citizens interact with the system through a simplified user entity, bypassing the need for account creation. The fileComplain method allows citizens to submit complaints directly, requiring only complaint information encapsulated in a Complain object.

-password: string

-username: string

+ User(username: string, password: string)  
+ changePassword(newPassword: string): void

+ Login(inputUsername: string, inputPassword: string): void  
+ getUsername(): string   
+ getPassword() : string

**2. Citizen Interaction:**

- Citizens can file complaints through the system, providing details such as incident date, time, location, and description. They have the capability to view the status of their filed complaints, enhancing transparency and accountability.

- citizenName : string (CITIZEN)

- contactNo :int

- address : string

- CNIC : int

- email : string

- Complain \*complain

+ fileComplain(const Complain &complain): void

+viewComplainStatus() : string

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

- complainBefore : bool(COMPLAIN)

- incidentDate : string

- incidentTime : string

- complainAddress : string

- FIRnumber : string

- PoliceStationID :int

- gender : string

- incidentDetail : string

-status (e.g., pending, assigned, resolved) : bool

+ getComplainBefore : bool

+ incidentDate : string

+ getIncidentTime : string

+ getFIRnumber : string

+ getPoliceStationID :int

.....getters

+ assign\_to\_officer()

+update\_status()

**3. Administrative Staff:**

- Administrative staff oversee officer data management, including adding new officers, updating records, and removing outdated data. They have access to citizen complaints for review and resolution, while also managing budget allocations without direct creation or deletion of budget items.

staffID: string

- fullName: string

- department : string

+ AdmisnistrativeStaff()

+ AdministrativeStafff(par)

+ ~Administrator()

+ setStaffID(id: String): void

+ getStaffID(): String

+ setFullName(name: String): void

+ getFullName(): String

+ setDepartment(dept: String): void

+ getDepartment(): String

+ enterNewOfficerData(officer: Officer): void

+ updateOfficerData(officer: Officer): void

+ deleteOfficerData(officerID: String): void

+ view\_complaints()

**4. Resource Management:**

- The system efficiently manages resources such as vehicles and equipment allocated to police stations. It facilitates the assignment of vehicles to officers and the allocation of equipment as per operational requirements.

budgetID: String (BUDGET)

- fiscalYear: int

- allocatedAmount: float

- remainingAmount: float

+ setBudgetID(id: String): void

+ getBudgetID(): String

+ setFiscalYear(year: int): void

+ getFiscalYear(): int

+ setAllocatedAmount(amount: float): void

+ getAllocatedAmount(): float

+ setRemainingAmount(amount: float): void

+ getRemainingAmount(): float

+ updateRemainingAmount(spentAmount: float): void

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

- vehicleID : int(VEHICLE)

-make : string

-model : string

-year : int

-Officer \*officer

+ AssignVehicleToOfficer() : string

+ updateVehicleInformation() : string

+ viewVehicles() : string

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

- equipmentID : int (EQUIPMENT)

- name : string

- description : string

- quantity : int

- Officer \*officer

+ Equipment()   
+ Equipment(int, string, string, int)  
+ ~Equipment()  
+ assignEquipmentToOfficer(Officer \*officer) : void   
+ updateEquipmentInformation(int, string, string, int) : void   
+ viewEquipment() const : void

**5. Police Station Operations:**

- Police stations utilize the system to streamline officer management, resource allocation, and administrative tasks.- They can add or remove officers, view officer details, manage resources, and address citizen complaints effectively.

-stationID :int(POLICE STATION)

-name :string

-location : string

-contactInfo : int

-staffCount : int

-crimeStatistics : double

-chiefOfficerInCharge : string

-Vehicles \*vehicles

-Equipment \*equipment

-Officer \*officer

+addOfficer (Officers \*officers) : void

+removeOfficer(Officers \*officers) : void

+viewOfficers(Officers \*officers) :void

+manageResources() : void

+add\_administrative\_staff() : void

+view\_complaints() : void

**6. Patrolling and Surveillance:**

- The system supports patrolling and surveillance activities, enabling law enforcement personnel to start and end patrolling sessions seamlessly. It records crucial details such as location and time, aiding in monitoring and reporting activities.

- Location: string (location where patrolling is conducted)

-StartTime: datetime (start time of patrolling)

- EndTime:datetime (end time of patrolling)

**7. Investigation:**

- Investigation units leverage the system to manage cases by assigning lead officers, updating case statuses, and monitoring ongoing investigations. The system facilitates collaboration and information sharing among investigation teams, enhancing efficiency and effectiveness

-caseID: string (unique identifier for the investigation case)

-status: string (current status of the investigation, e.g., ongoing, closed)

-leadOfficer: Officer (lead officer in charge of the investigation)

+assignLeadOfficer(Officer): assigns a lead officer to the investigation

+updateStatus(string): updates the status of the investigation

**8. Officer Management:**

- Officers' details, including unique identifiers, names, ranks, and badge numbers, are stored within the system. Functionalities such as promoting or demoting officers, assigning tasks, and updating case statuses ensure efficient coordination and management of law enforcement personnel.

- officerID: int

- name: string

- rank: string

- badgeNumber: int

+ Officer()   
+ Officer(int, string, string, int)  
+ ~Officer()

+ virtual void displayInfo() const

+ void viewCasesAssigned() const

+ void updateCaseStatus()

+ void logActivity()

+ resolve\_complaint()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

- yearsOfService: int (**JuniorPoliceForce**)  
- department: string

+ promote()

+ demote()   
+ assignCase()   
+ displayInfo() const

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

- patrolArea: string(**Constable)**

- jurisdiction: String

- shift: String

- patrolVehicle: String

- performanceEvaluations: String

- equipment:array  
+ Constable(name: String, badgeNumber: String, jurisdiction: String)

+ patrol()   
+ conductInvestigation()

+ arrestSuspect(suspectName: String): void

+ issueCitation(violationType: String, offenderName: String): void

+ respondToCall(callDetails: String): void

+ conductInvestigation(caseNumber: String): void

+ updateShiftSchedule(newSchedule: String): void

+ provideAssistance(location: String): void

+ generateReport(): String

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

+ HeadConstable(name: String, badgeNumber: String, jurisdiction: String) (**HeadConstable**)

+ arrestSuspect(suspectName: String): void

+ issueCitation(violationType: String, offenderName: String): void

+ respondToCall(callDetails: String): void

+ conductInvestigation(caseNumber: String): void

+ updateShiftSchedule(newSchedule: String): void

+ provideAssistance(location: String): void

+ generateReport(): String

+ supervisePatrol(): void

+ managePersonnel(): void

**Sub-Inspector (SI)**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**AssitantSub-Inspector (ASI)**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**InspectorOfPolice (IP)**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

- yearsOfService: int (**SeniorPoliceForce**)  
- department: string  
+ promote()

+ demote()   
+ assignCase()   
+ displayInfo() const

**AssistantSuperintendent**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Superintentendant**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**SeniorSuperintendent**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**AdditionalInspectorGeneral**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**InspectorGeneral**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**DeputyInspectorGeneral**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CorrectionalOfficer**

**9. Correctional Facilities:**

- Correctional officers manage cell assignments for criminals and update cell information as needed, ensuring optimal utilization of correctional resources.

- celID: int

- location: string

- capacity: int

- occupancy: int

+ assignCellToCriminal()  :  void

+ updateCellInformation()  : void

+ viewCellsAvalaible() : int

**10. Crime Management:**

- The system comprehensively records details of various crime types, including white-collar crimes, violent crimes, and cybercrimes. It captures crime descriptions, dates, locations, and status updates, facilitating effective crime tracking and management.

- crimeID : int (**Crime)**

- description : string

- date : int

- location : string

- status : bool

 - motive : string

 - time : string

+ Crime()   
+ Crime(para constructor)  
+ ~Crime()   
+ virtual void recordNewCrime()   
+ virtual bool updateCrimeStatus() const   
+ virtual void viewListOfCrimes() const

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

- financialLoss : double(**WhiteCollarCrimes**)

+ descriptionOfTheCrime() : string

+ WhiteCollarCrimes()   
+ WhiteCollarCrimes(ipara constructor)  
+ ~WhiteCollarCrimes()   
+ void recordNewCrime()   
+ bool updateCrimeStatus() const   
+ void viewListOfCrimes() const

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-weaponsInvolved : string(**ViolentCrimes**)

+ descriptionOfTheCrime(): string

+ ViolentCrimes()   
+ ViolentCrimes(int, string, int, string, bool, string, string, string)   
+ ~ViolentCrimes()   
+ void recordNewCrime()   
+ bool updateCrimeStatus() const  
+ void viewListOfCrimes() const

- cyberAttackType : string (**CyberCrimes**)

- securityBreaches : int

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

+ descriptionOfTheCrime(): string

+ CyberCrimes()   
+ CyberCrimes(int, string, int, string, bool, string, string, string, int)  
+ ~CyberCrimes()   
+ void recordNewCrime() override  
+ bool updateCrimeStatus() const override  
+ void viewListOfCrimes() const override

**11. Evidence Handling:**

- The system supports evidence collection and analysis through forensic labs and experts, assisting in criminal investigations and enhancing the integrity of evidence handling processes.

-id: int(**Evidence )**

-description: string  
-locationFound: string

+collect(): void

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-location: string(**ForensicLab**)  
-equipment: Equipment[]

+analyzeEvidence(): void

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-id: int(**ForensicExpert)**  
-name: string  
-specialization: string

+analyzeEvidence(): void

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**scenarios**:

**Citizen Crime Reporting:**

- A citizen witnesses a robbery in progress and immediately uses the law enforcement management system to report the crime. They provide details such as the time, location, and description of the suspects. The system assigns a unique complaint ID and categorizes the report as "Robbery."

- Administrative staff at the police station receive the complaint and assign it to an available officer for investigation.

- The assigned officer reviews the complaint, updates its status to "Assigned," and begins investigating the robbery.

- Meanwhile, patrol officers receive a notification about the reported robbery and adjust their patrolling routes to focus on the area where the crime occurred.

- The investigating officer collaborates with forensic experts to analyze any evidence collected from the crime scene, such as fingerprints or CCTV footage.

- After gathering sufficient evidence, the officer updates the status of the complaint to "Resolved" once the suspects are apprehended, and they are successfully prosecuted.

**Resource Management and Allocation**:

- The police station receives an allocation of new vehicles and equipment for patrol and law enforcement activities.

- Administrative staff use the law enforcement management system to update the inventory of vehicles and equipment, assigning them unique IDs and recording their specifications.

- Officers responsible for managing resources review the available inventory and allocate vehicles and equipment to patrol units based on operational needs and officer requests.

- Patrol officers receive notifications about the assignment of vehicles and equipment and use the system to confirm receipt.

- The system tracks the usage of vehicles and equipment, monitoring their availability and scheduling maintenance as needed.

**Patrol Scheduling and Monitoring:**

- Supervisors access the law enforcement management system to create patrol schedules for officers based on factors such as crime trends, community events, and staffing levels.

- Patrol officers receive their assigned schedules through the system and use it to plan their patrols accordingly.

- During patrols, officers use the system to log their activities, including locations visited, incidents encountered, and interactions with citizens.

- Supervisors monitor patrol activities in real-time through the system, tracking officer locations and ensuring adherence to assigned schedules.

- If an emergency or crime occurs, officers can request assistance or report incidents through the system, allowing supervisors to deploy additional resources as needed.