Application Flow Diagrams

MehfoozPakistan Crime Pattern Analysis System

December 9, 2025

Contents

1 Introduction

This document presents comprehensive flow diagrams demonstrating how users interact with the MehfoozPakistan Crime Pattern Analysis System (CPAS). The system supports three user roles: Officers, Victims, and Witnesses, each with distinct workflows and access permissions.

2 System Architecture Overview

The MehfoozPakistan CPAS follows a three-tier architecture:

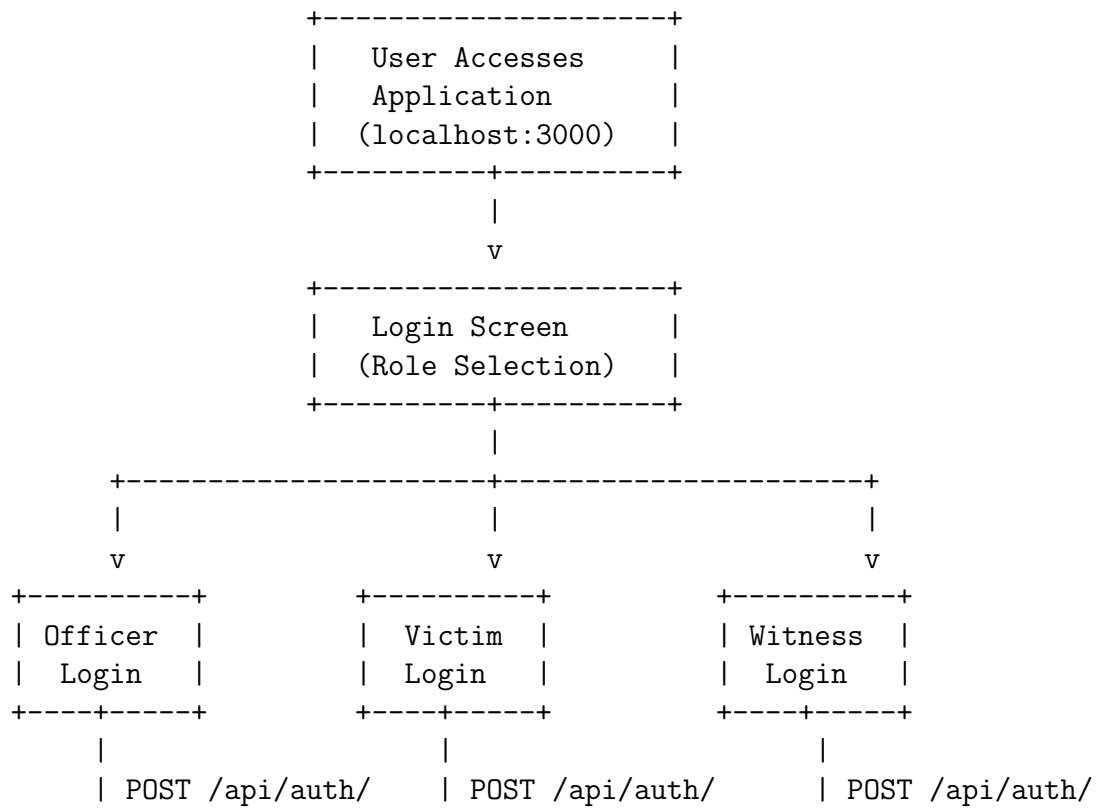
- **Frontend:** React application running on port 3000
- **Backend:** Node.js Express server running on port 5000
- **Database:** Oracle Database 19c running on port 1521

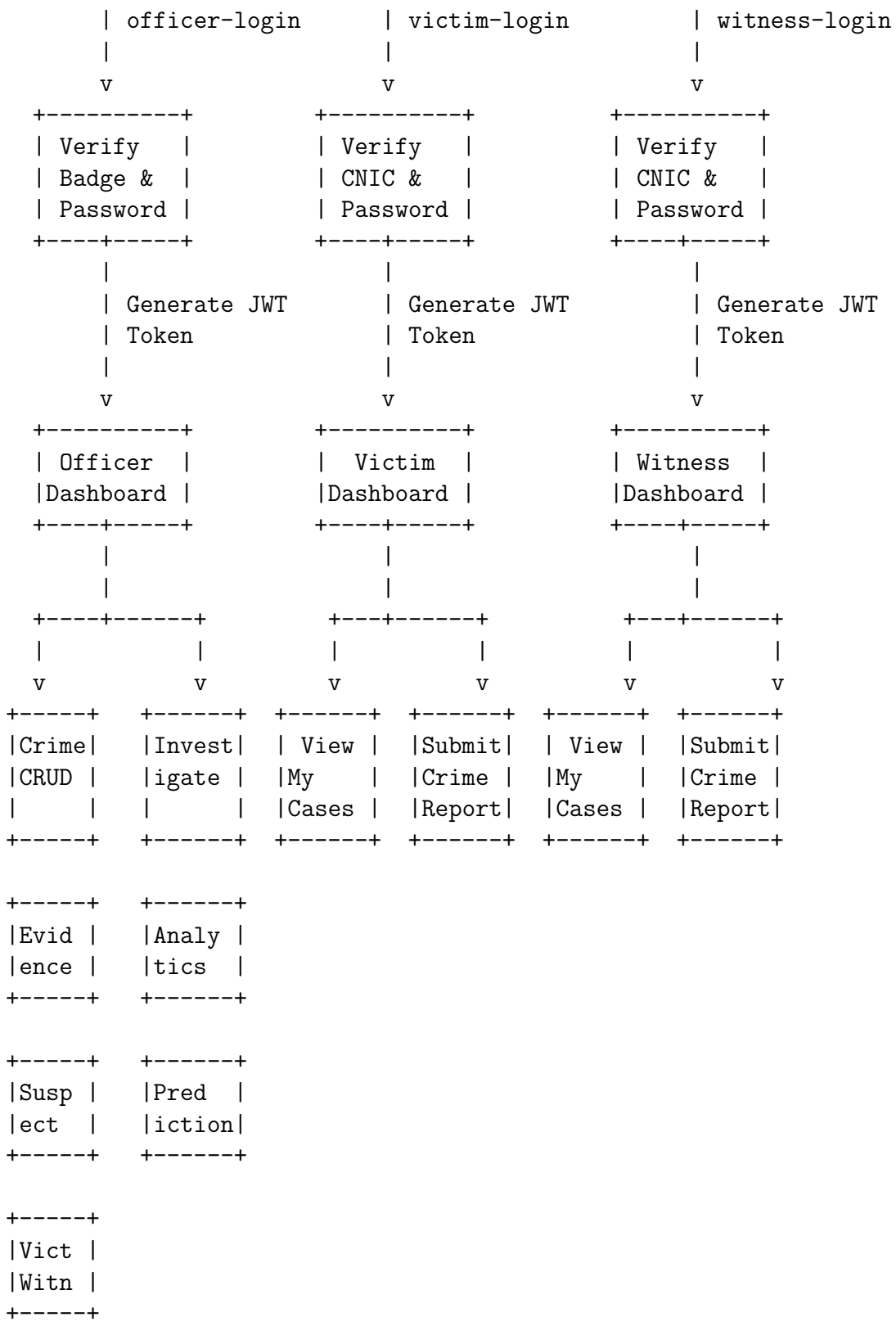
User authentication is handled via JWT tokens with bcrypt-hashed passwords stored securely in the database. Role-based access control ensures that each user type can only access authorized features.

3 Application Flow Diagrams

3.1 Overall System Flow

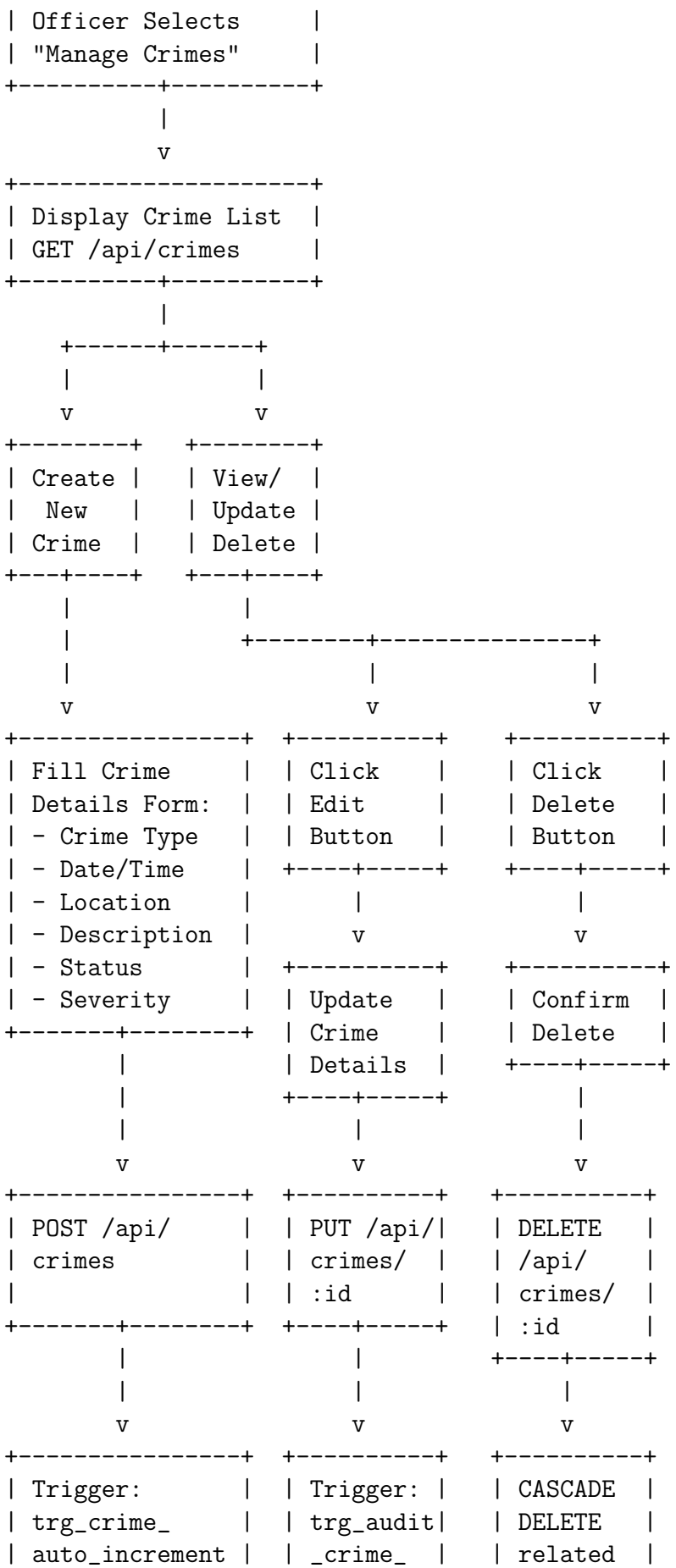
The following diagram illustrates the complete user journey from initial access through authentication to role-specific operations:

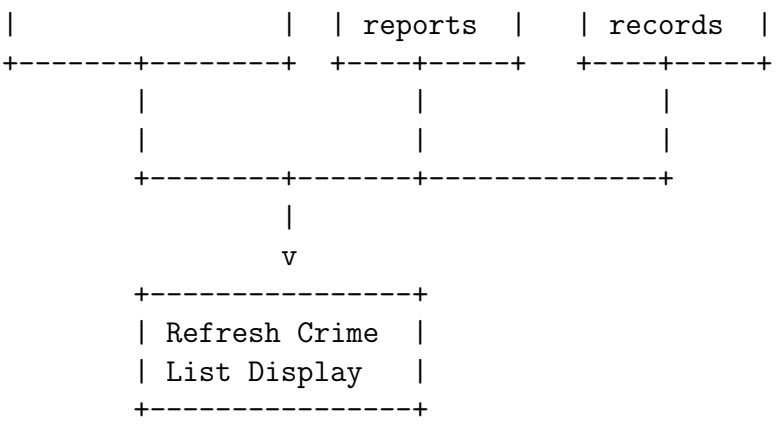




3.2 Crime Management Workflow (Officer)

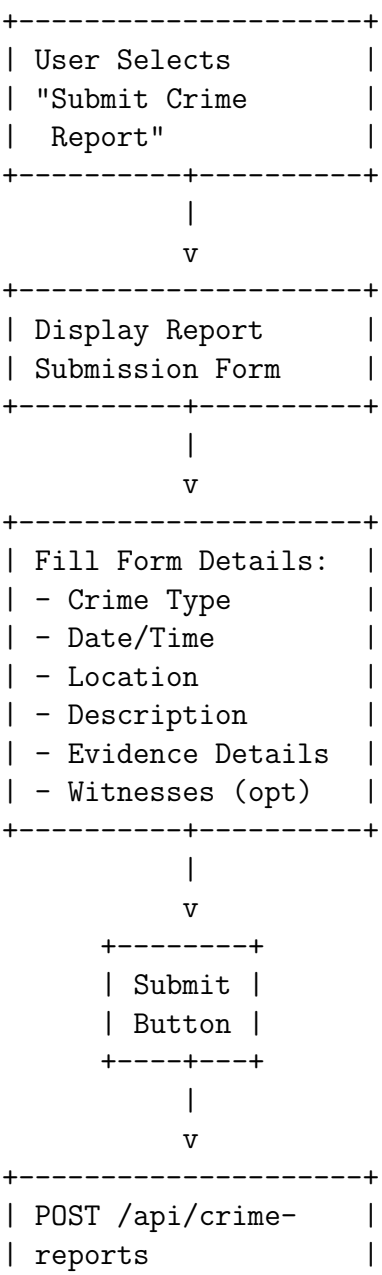
This diagram shows the complete CRUD operations for crime management available to officers:

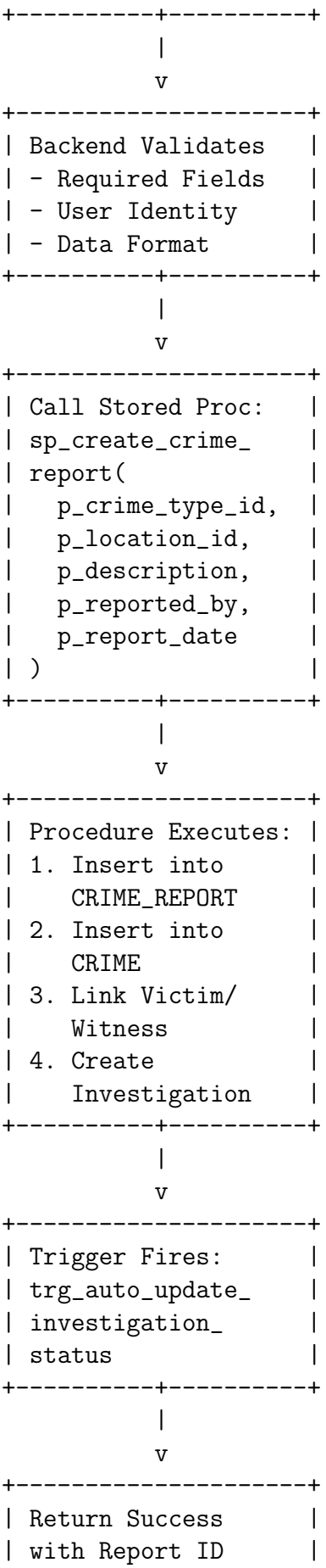

Officer Selects "Manage Crimes"
Display Crime List GET /api/crimes
Create New Crime
View/ Update Delete
Fill Crime Details Form: - Crime Type - Date/Time - Location - Description - Status - Severity
Click Edit Button
Click Delete Button
POST /api/ crimes
PUT /api/ crimes/ :id
DELETE /api/ crimes/ :id
Trigger: trg_crime_ auto_increment
Trigger: trg_audit _crime_
Update Crime Details
Confirm Delete
CASCADE DELETE related

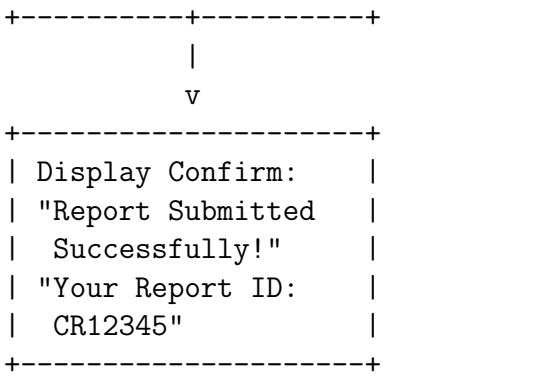


3.3 Crime Report Submission Workflow (Victim/Witness)

This diagram illustrates how victims and witnesses submit crime reports:

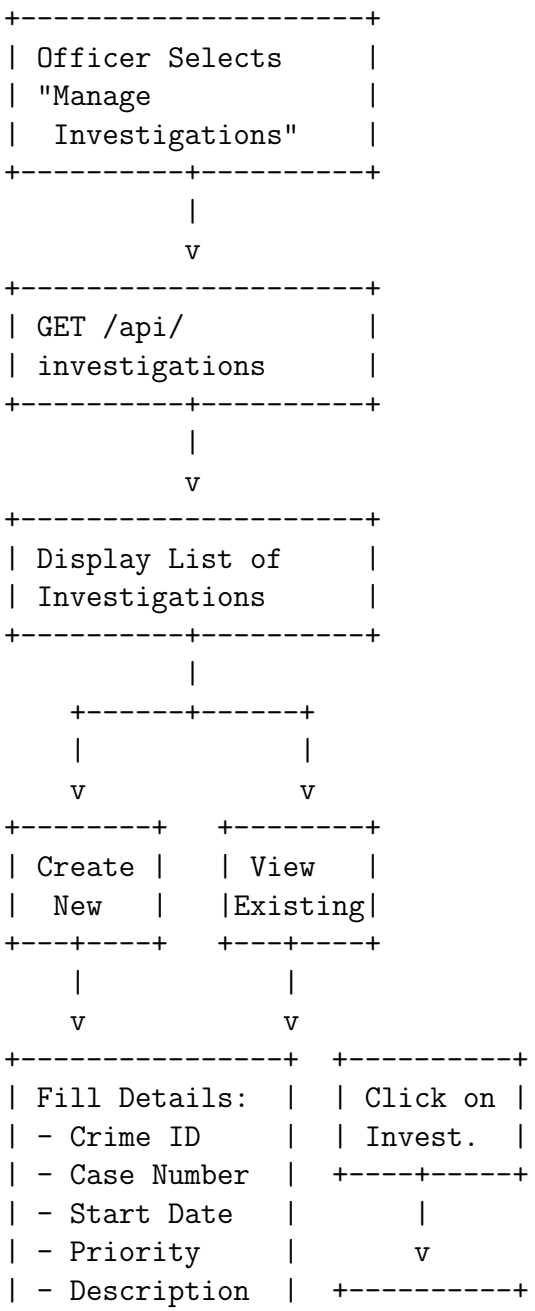


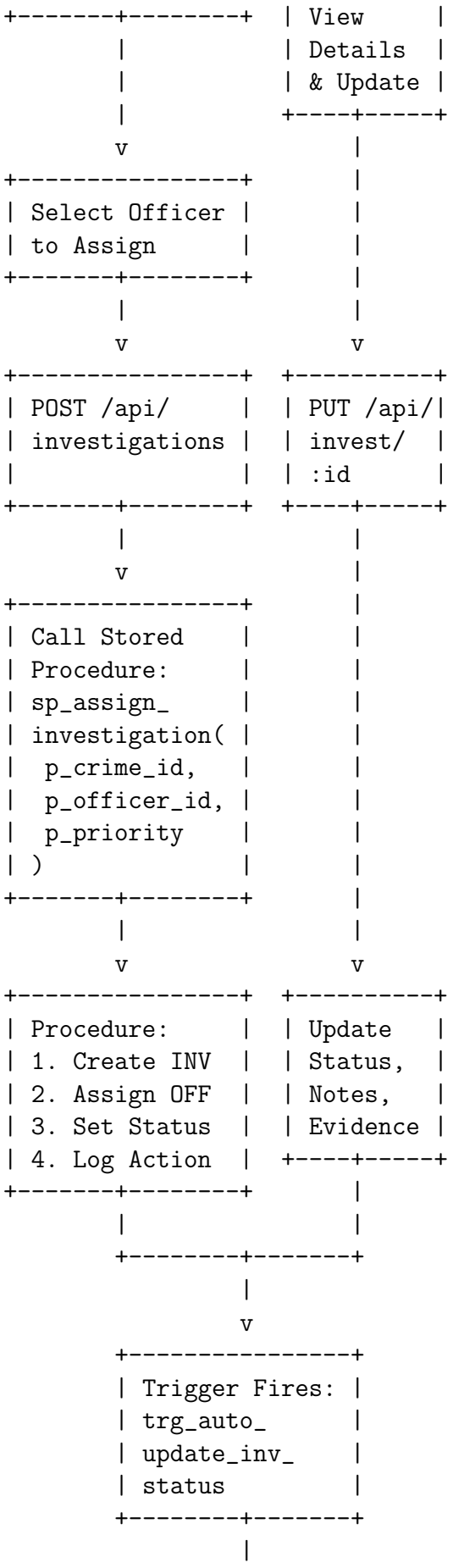

Backend Validates
- Required Fields
- User Identity
- Data Format
Call Stored Proc:
sp_create_crime_
report(
p_crime_type_id,
p_location_id,
p_description,
p_reported_by,
p_report_date
)
Procedure Executes:
1. Insert into
CRIME_REPORT
2. Insert into
CRIME
3. Link Victim/
Witness
4. Create
Investigation
Trigger Fires:
trg_auto_update_
investigation_
status
Return Success
with Report ID

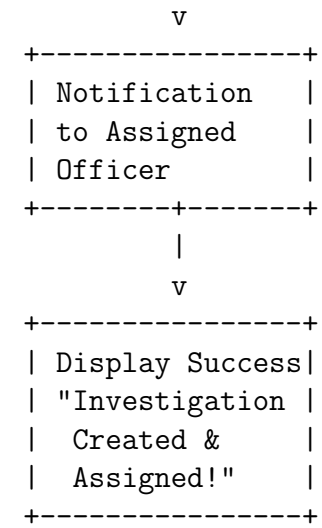


3.4 Investigation Assignment Workflow (Officer)

This diagram shows how officers create and assign investigations:

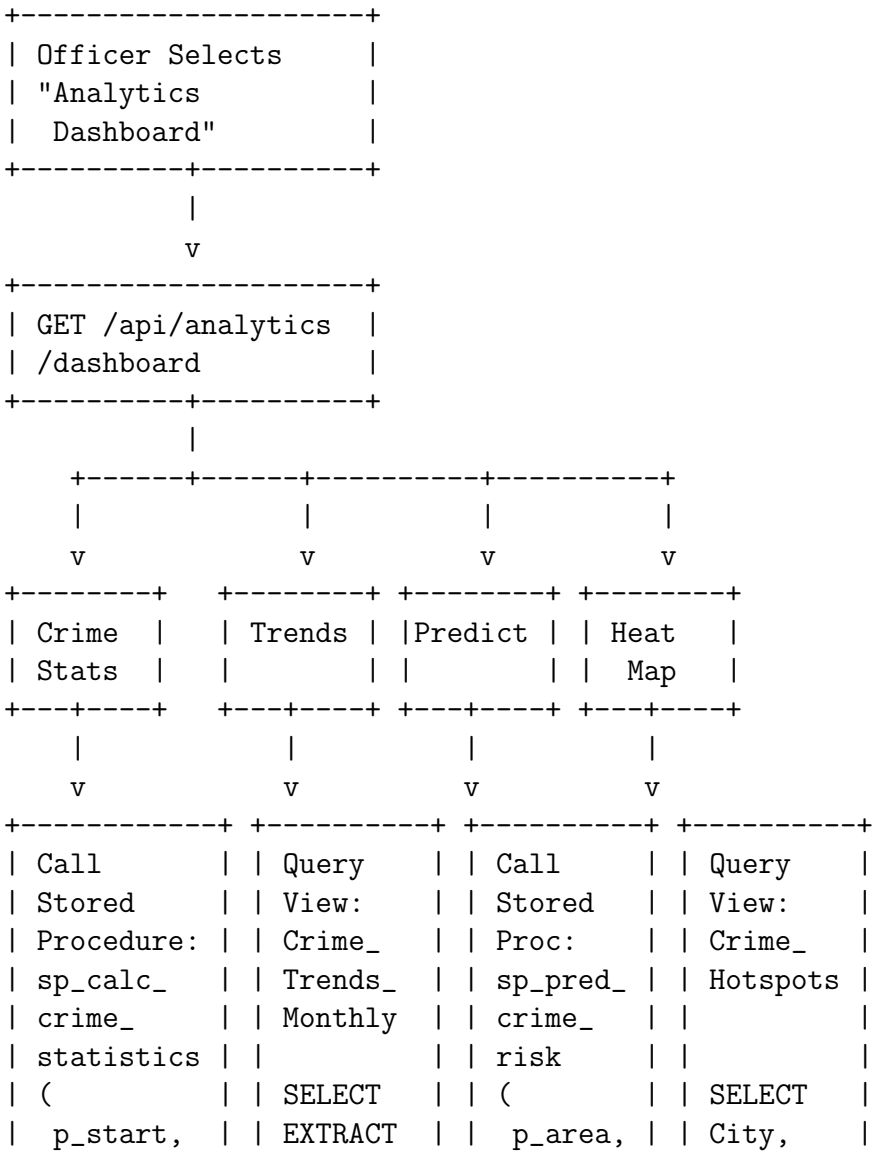



View Details & Update
Select Officer to Assign
POST /api/investigations
PUT /api/invest/:id
Call Stored Procedure: sp_assign_investigation(p_crime_id, p_officer_id, p_priority)
Procedure: 1. Create INV 2. Assign OFF 3. Set Status 4. Log Action
Update Status, Notes, Evidence
Trigger Fires: trg_auto_update_inv_status



3.5 Analytics Dashboard Workflow (Officer)

This diagram shows how officers access crime analytics and predictions:



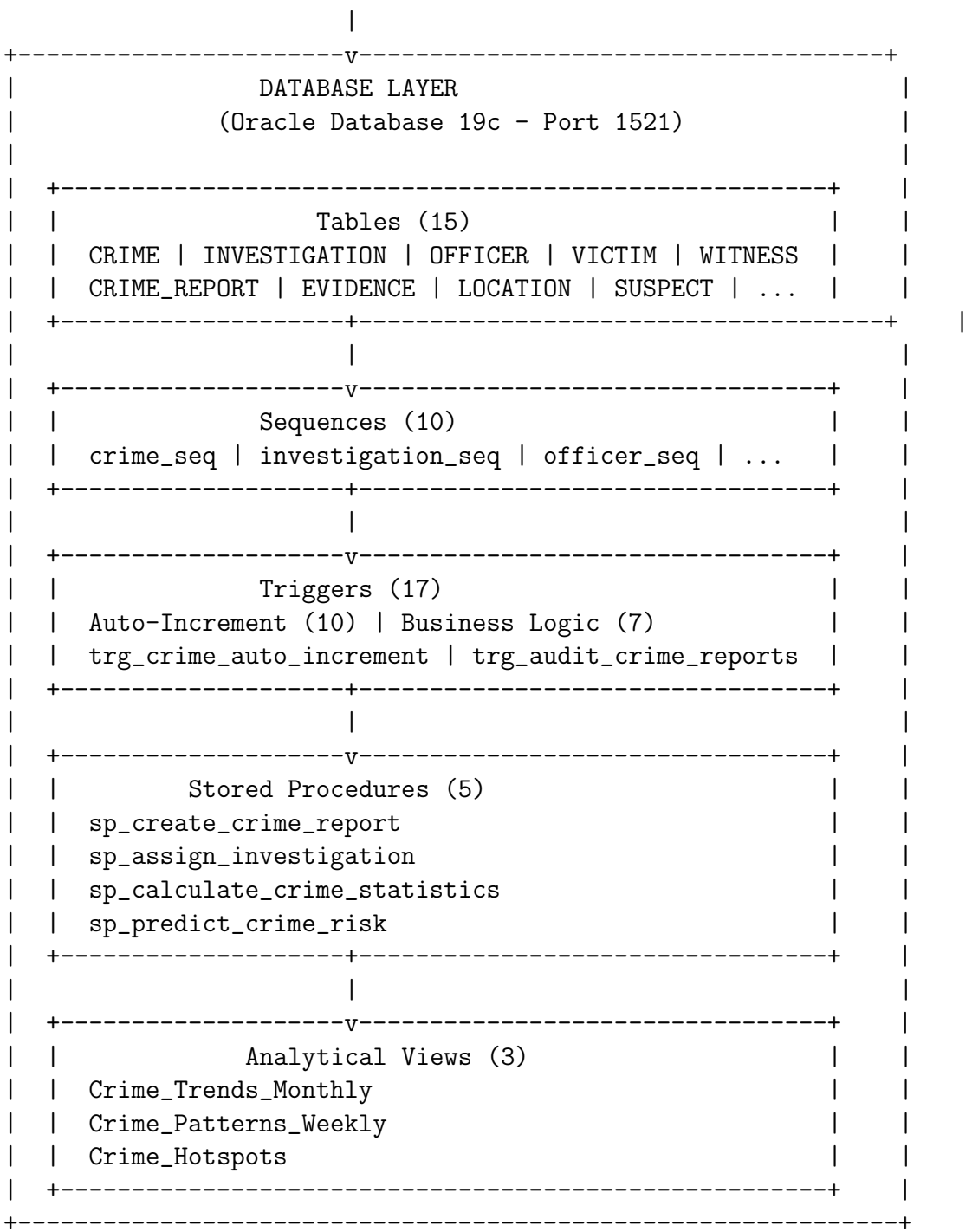
```
|  p_end,    | | (MONTH), | |  p_type  | | Area,    |
|  p_type    | | COUNT(*) | | )        | | COUNT(*) |
| )          | | GROUP BY | |          | | GROUP BY |
|            | | MONTH    | |          | | Location |
+-----+------+ +----+-----+ +----+-----+ +----+-----+
      |            |          |          |
      |            |          |          |
      v            v          v          v
+-----------------------------------------------+
| Return JSON Data:                             |
| - Total Crimes                                |
| - Crimes by Type                              |
| - Crimes by Status                            |
| - Monthly Trends                              |
| - Predictions                                 |
| - Hotspot Locations                           |
+------------------+----------------------------+
                   |
                   v
+-----------------------------------------------+
| Frontend Renders:                             |
| - Bar Charts (Crime Types)                    |
| - Line Graphs (Trends)                        |
| - Pie Charts (Status Distribution)            |
| - Heatmap (Geographic Hotspots)               |
| - Prediction Alerts                           |
+------------------+----------------------------+
                   |
                   v
+-----------------------------------------------+
| Officer Can:                                  |
| - Filter by Date Range                        |
| - Filter by Crime Type                        |
| - Filter by Location                          |
| - Export Reports (CSV/PDF)                    |
| - Drill Down into Details                     |
+-----------------------------------------------+
\end{verbatim>

\subsection{Data Flow Architecture}

This diagram shows the technical flow of data through the system layers:

\begin{verbatim}
+-------------------------------------------------------------+
|                     FRONTEND LAYER                          |
|                   (React - Port 3000)                       |
|                                                             |
```


Login Component
Dashboard Component
Forms Components
Analytics Component
api.js (Axios)
HTTP Requests (JWT Token)
BACKEND LAYER
(Node.js Express - Port 5000)
Routes Layer
authRoutes | crimeRoutes | investigationRoutes
Middleware Layer
authMiddleware.js (JWT Verification)
roleMiddleware.js (Permission Check)
Controllers Layer
authController | crimeController | analyticsCtrl
Models Layer
CrimeModel | InvestigationModel | AnalyticsModel
Database Connection (db.js)
oracledb.getConnection()
SQL Queries



4 Key Features Demonstrated

4.1 Authentication and Authorization

- **Multi-Role Support:** Three distinct user types with separate login endpoints
- **JWT Tokens:** Stateless authentication using JSON Web Tokens
- **Password Security:** Bcrypt hashing for password storage
- **Role-Based Access Control:** Middleware enforces permissions at route level

4.2 Database Automation

- **Auto-Increment:** Sequences and triggers for primary key generation
- **Business Logic Triggers:** Automatic validation, auditing, and status updates
- **Stored Procedures:** Complex multi-table operations encapsulated in database
- **Analytical Views:** Pre-computed statistics for dashboard performance

4.3 User Experience

- **Intuitive Navigation:** Dropdown menus with role-specific options
- **Real-time Feedback:** Success/error messages for all operations
- **Data Visualization:** Charts and graphs for crime analytics
- **Responsive Design:** Mobile-friendly interface using React

5 Security Considerations

The application implements multiple layers of security:

1. **Input Validation:** All user inputs validated on both frontend and backend
2. **SQL Injection Prevention:** Parameterized queries and prepared statements
3. **Authentication Tokens:** Expire after configurable time period
4. **Database Constraints:** CHECK constraints enforce data integrity
5. **Cascade Delete Protection:** Foreign key constraints prevent orphaned records
6. **Audit Logging:** Triggers record all critical database modifications

6 Conclusion

The MehfoozPakistan CPAS provides a comprehensive solution for crime pattern analysis and management. The application flow diagrams demonstrate clear separation of concerns, robust authentication mechanisms, and efficient data processing through database-level automation. The three-tier architecture ensures scalability, maintainability, and security for handling sensitive law enforcement data.