



CS4961

Senior Design

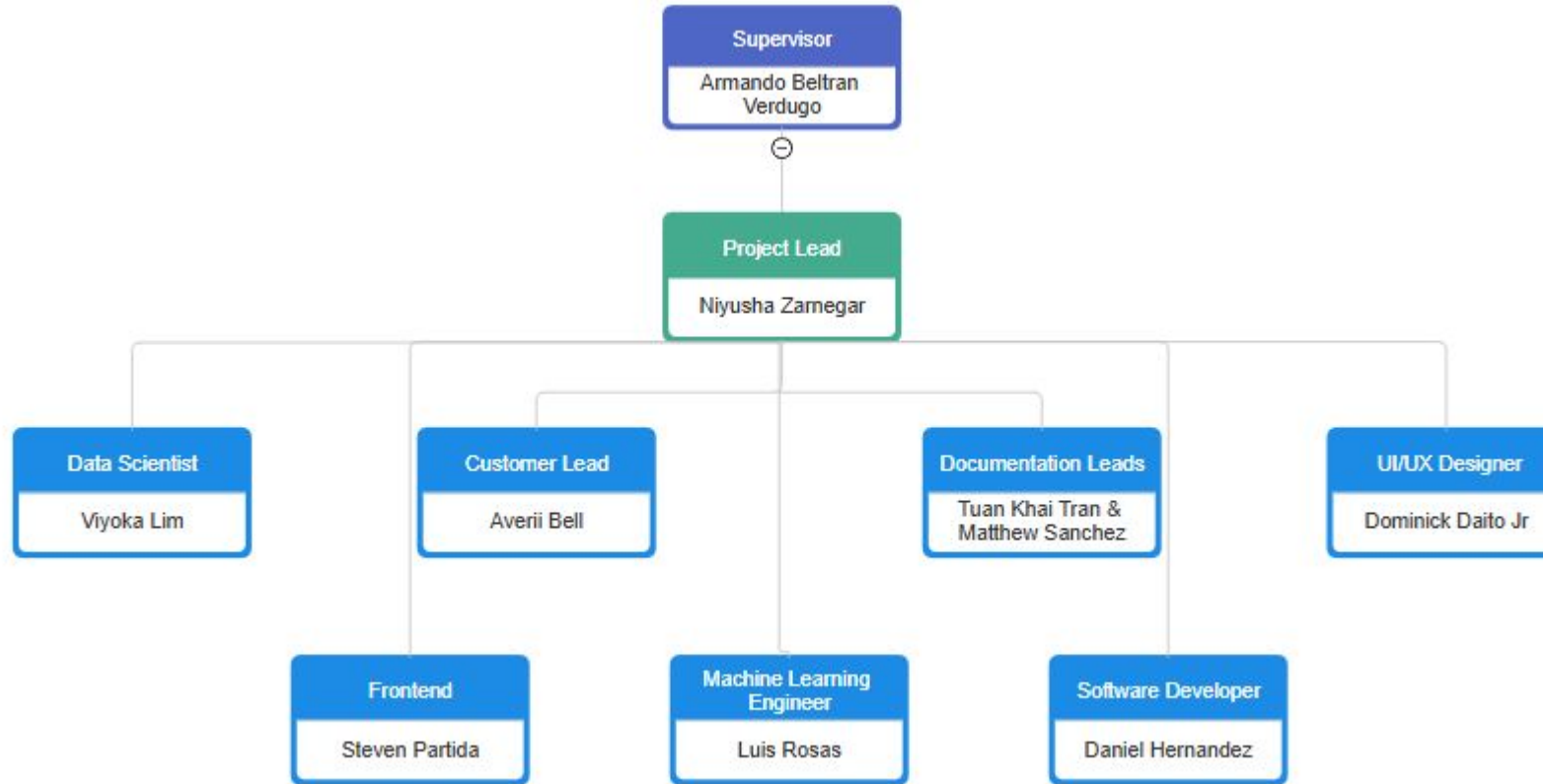
Project: AWS: AI-Powered Legal
Analysis of Police Misconduct

Team

**Niyusha Zarnegar, Dominick Daito Jr, Tuan Khai Tran, Averii Bell,
Steven Partida, Daniel Hernandez, Viyoka Lim, Matthew Sanchez,
Luis Rosas**

Department of Computer Science,
College of Engineering, Computer Science, and Technology
California State University, Los Angeles

Team Organization



Sponsored by:



In collaboration with



Agenda

- Software Requirements
- Software Design Document
- Next Steps
- Closing Thoughts and Questions

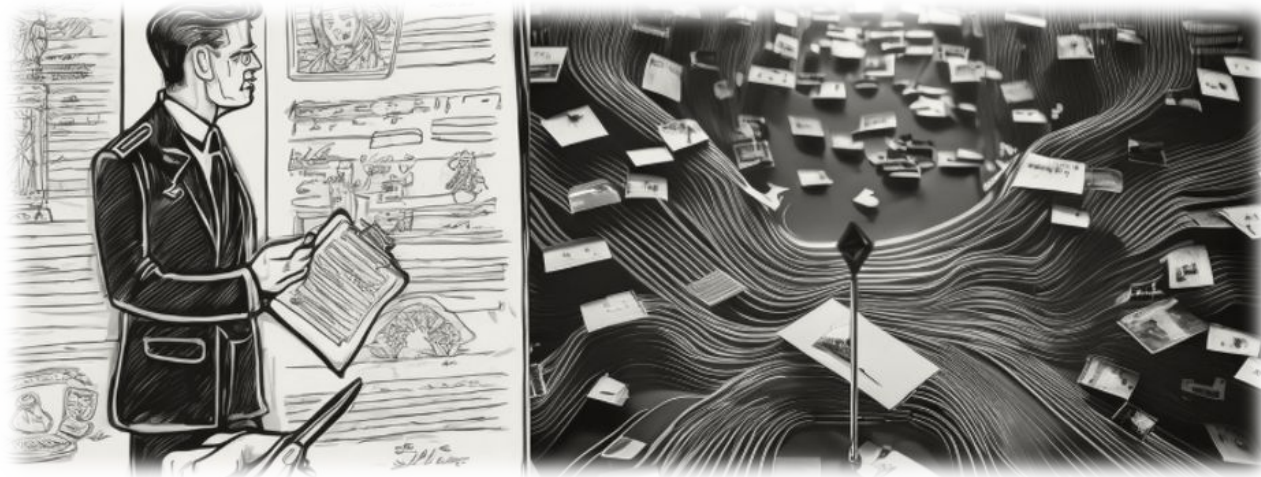
Software Requirement Document

Project Overview

- Problem Statement
- Goals and Objectives
- Target Audience/Users

Problem Statement

- The Los Angeles County Public Defender's office faces challenges in efficiently reviewing court transcripts for police misconduct.
- The current manual process is time-consuming and is an inefficient use of man hours, risking overlooked crucial details.
- We aim to develop an **automated transcript analysis** using AWS to enhance accuracy, efficiency, and accountability in identifying police misconduct.



Goal and Objectives

Goal

- Improve the efficiency and accuracy of legal workflows by enhancing the analysis of court transcripts.

Objectives

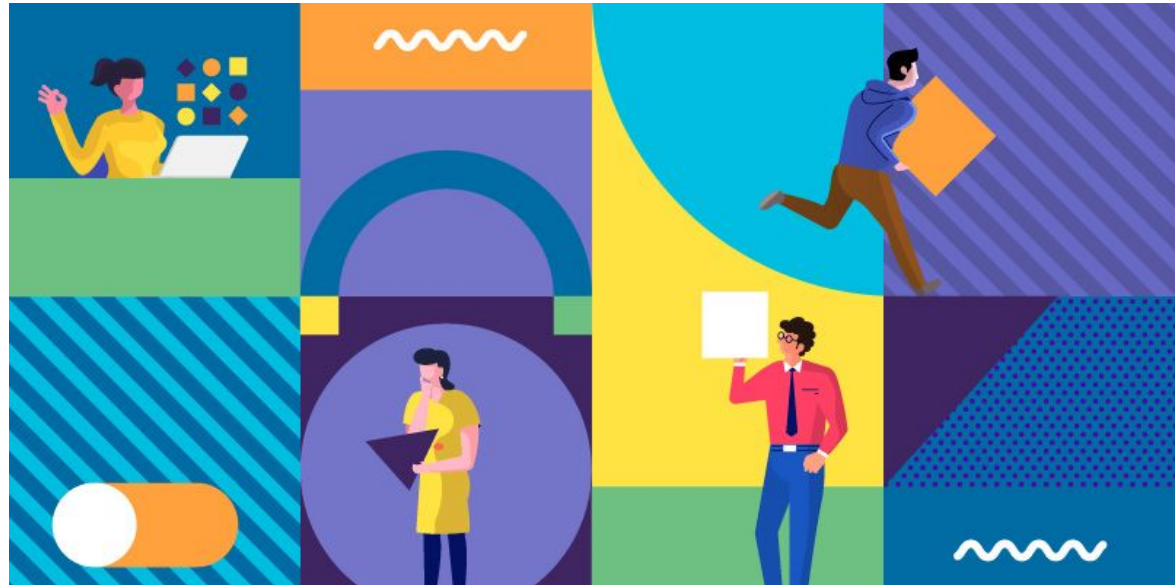
- Analyze court transcripts to detect patterns of police misconduct.
- Integrate advanced technologies:
 - AWS cloud services
 - Natural Language Processing (NLP)
 - Generative AI
- Identify critical insights:
 - Detect potential misconduct.
 - Flag inconsistencies in testimonies.
 - Extract and highlight key entities for legal use.
- Provide real-time processing to support legal teams in decision-making.

The Target Audience/Users

- **LA County Public Defender's Office**

- a. Legal Professionals:**

- Streamline case preparation, identify key insights quickly.



Functional Requirements

Transcript Upload and Storage:

- **Description:** Upload court transcripts (PDF, DOCX). OCR converts to searchable text.
- **Requirement:** Store in Box.com with encryption for secure access.

Text Analysis, Key Entity Extraction, and Misconduct Detection:

- **Description:** NLP extracts entities (e.g., officer names, locations) and detects misconduct, flagging for review.
- **Requirement:** Use NLP model hosted on EC2. Store results in a searchable format.

Case Linking and Relationship Detection:

- **Description:** Identify and link related cases via shared attributes such as individual officers and recurring locations
- **Requirement:** Use clustering/similarity algorithms for presenting relationships.

Multi-Level Flagging System:

- **Description:** Generate color-coded flags indicating confidence levels in detected issues.
- **Requirement:** Automatic flag generation with export options (PDF, CSV).

Non-Functional Requirements

Performance:

- Supports up to 50 simultaneous uploads with API response times under 2 seconds, ensuring efficient data processing.

Scalability:

- Accommodates future increases in data volume and user base through microservices architecture and auto-scaling.

Security:

- Ensures encrypted data storage and transmission (e.g., HTTPS, AES-256) with role-based access and access code validation.

Reliability:

- Guarantees 99.9% system uptime and data redundancy using solutions like Box.com and AWS S3.

Usability:

- Provides an intuitive, user-friendly interface with clear feedback and support for handling large, complex data sets.

User Interface

- **Login:** Secure authentication ensures that only authorized users can access sensitive investigation data.
- **Upload Transcripts:** Users can upload various transcripts such as audio, text documents, and videos.
- **Transcript Analysis:** The system will analyze the uploaded transcript for important information such as names, locations, and dates while highlighting relations between patterns.
- **Keyword and Entity Search:** Users can search for keywords and tags to find relevant information.
- **Annotation and Tagging:** Users can annotate transcripts with comments or tag specific segments of the transcript for further investigation which could ultimately benefit patterns and relations.
- **Pattern Recognition:** The system will analyze the transcript for patterns like frequent mentions of events, names, locations, and dates. Visualization will be presented through graphs or summaries.
- **Reports:** Users can generate custom reports summarizing findings and identified keywords.
- **Searchable Database:** A searchable database consisting of uploaded transcripts for easy access to past investigations and annotations.
- **Alerts:** Notifications for key findings.

Assumptions and Constraints

- **Key Assumptions**

- Training Data Availability
- Integration with Box, which is LACPD's main document store
- AWS Infrastructure and Service Availability
- Data Security and Compliance Requirements

- **Constraints that impact development**

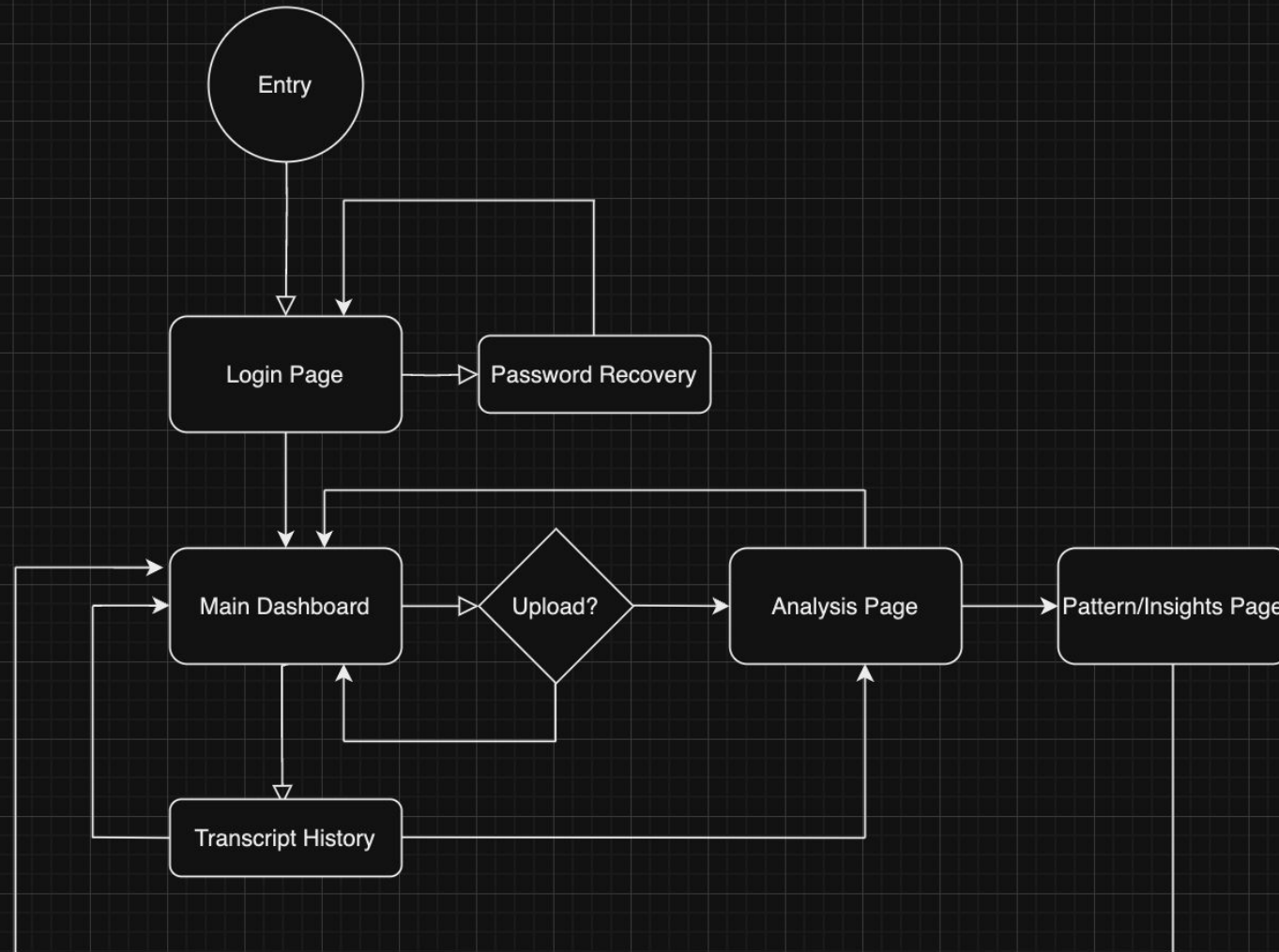
- Compliance with LACPD's Data Privacy and Regulations
- AWS cloud services is the primary platform
- Must support multi-user access with role-based permissions
- 24-hr RPO and ≤ 6 -hr RTO

Requirements Validation

- **Stakeholder Reviews:**
 - Regular meetings with the LA County Public Defender's Office to gather feedback on functionality and user experience.
- **Use Case Testing:**
 - Validation of user workflows, such as uploading documents, viewing flagged sections, and exporting reports.
- **Pilot Deployment:**
 - A limited rollout of the system to a small group of attorneys and paralegals to gather real-world feedback.
- **Acceptance Testing:**
 - Formal testing by stakeholders against acceptance criteria defined in the SRS.

Software Design Document

System Architecture



Key Components

Frontend Module (Node.js):

- Provides the user interface for uploading transcripts, viewing results, and managing accounts.
- Includes features like login, registration (with access codes), and data visualization.

Backend Module (Python):

- Handles API requests, NLP tasks (e.g., text analysis, entity extraction, keyword summarization), and Box API interactions.
- Manages logic for transcript analysis and links results to cases.

File Management (Box API):

- Facilitates secure file uploads, downloads, and storage of transcripts.
- Integrates metadata for indexing and quick retrieval.

NLP Processing (LLaMA):

- Processes text data to extract insights such as misconduct detection, entity recognition, and summarization.
- Operates as a background task triggered after file upload.

Database/Storage (AWS S3 and Box):

- Stores structured data (case information, analysis results) and unstructured data (transcripts, audio, videos).
- Ensures data redundancy and security.

Monitoring and Logging:

- Tracks user actions, file uploads, and system performance for audit and debugging purposes.
- Uses tools like AWS CloudWatch or Datadog.

Technology

- **Framework and tool that we will use**
 - **AWS Cloud Services:** Scalable and secure infrastructure
 - **NLP Libraries:** For text analysis and entity recognition
 - **Generative AI Models:** Advanced text processing and pattern detection
 - **Amazon S3:** Storage for court transcripts
 - **Amazon EC2:** Flexible compute resources for analysis
 - **Amazon RDS:** Management of relational databases
 - **Amazon ECS:** VMs for application hosting
- **Rationale for technology**
 - **AWS Cloud Services:** Ensures scalability, security, and compliance for sensitive legal data.
 - **NLP & Generative AI:** Automates complex text analysis to identify misconduct patterns.
 - **Amazon S3:** Durable storage with easy data retrieval.
 - **Amazon EC2:** Scalable compute resources based on processing needs.
 - **Amazon RDS:** Reliable management of case-related structured data.
 - **Amazon ECS:** Simplifies deployment and scalability of applications.

Challenge and Solutions

General Constraints and Solutions

- **Compliance with LACPD's Data Privacy and Security Regulations**
Solution: Implement encryption (at rest and in transit), role-based access control, and regular audits to ensure adherence to policies.
- **AWS Cloud Services as Primary Platform**
Solution: Leverage AWS's built-in security features, scalability, and compliance certifications (e.g., SOC 2, HIPAA).
- **Multi-User Access with Role-Based Permissions**
Solution: Use AWS Identity and Access Management (IAM) to define roles and enforce permissions at a granular level.
- **24-Hour RPO and ≤ 6 -Hour RTO**
Solution: Implement automated backups, disaster recovery plans, and replication across AWS regions for high availability.

Next Steps

- **Development Phase:**
 - Implement the automated transcript analysis and case management system
- **Testing Phase**
- **Deployment Phase:**
 - Deploy a pilot version on AWS.
 - Enable real-time system monitoring.





CS4961
Senior Design

Project: AWS: AI-Powered Legal
Analysis of Police Misconduct

Team

**Niyusha Zarnegar, Dominick Daito Jr, Tuan Khai Tran, Averii Bell,
Steven Partida, Daniel Hernandez, Viyoka Lim, Matthew Sanchez,
Luis Rosas**

Department of Computer Science,
College of Engineering, Computer Science, and Technology
California State University, Los Angeles

Questions?

Thank you