Smarter Parking for a University Campus

Patrick O'Boyle, David Hiser, Saumya Karia, Guarang Goel

Objective / Background

- Problem: Campus parking
- Solution: Smart Parking
- Approach: Computer Vision

- Implementation: Demo that demonstrates the necessary features to help students and staff find parking spots
- Scope: Our project focuses on the *back-end* to stay modular
    - Can send our data to any interface that displays where open spots are

UI Demo

- **Multiplexed** screen
  - Grid of camera views
- Timestamp
- Per-view occupancy counts
  - **# of cars in spots**
  - Out of total spots
- Linux Terminal (**admin**)
  - Read-Execute-Print Loop (like fdisk, sftp, etc)
  - Updating occupancy values
  - Log access
- No screen transitions
  - Main use case: **monitoring**

Lot A **<timestamp>** <camera1 view> **Vehicles: 22/25**	Lot B <camera2 view> **Vehicles: 13/30**	Lot B <camera3 view> **Vehicles: 55/60**
Lot B <camera4 view> **Vehicles: 32/38**	Lot C <camera5 view> **Vehicles: 31/35**	Lot C <camera6 view> **Vehicles: 27/34**

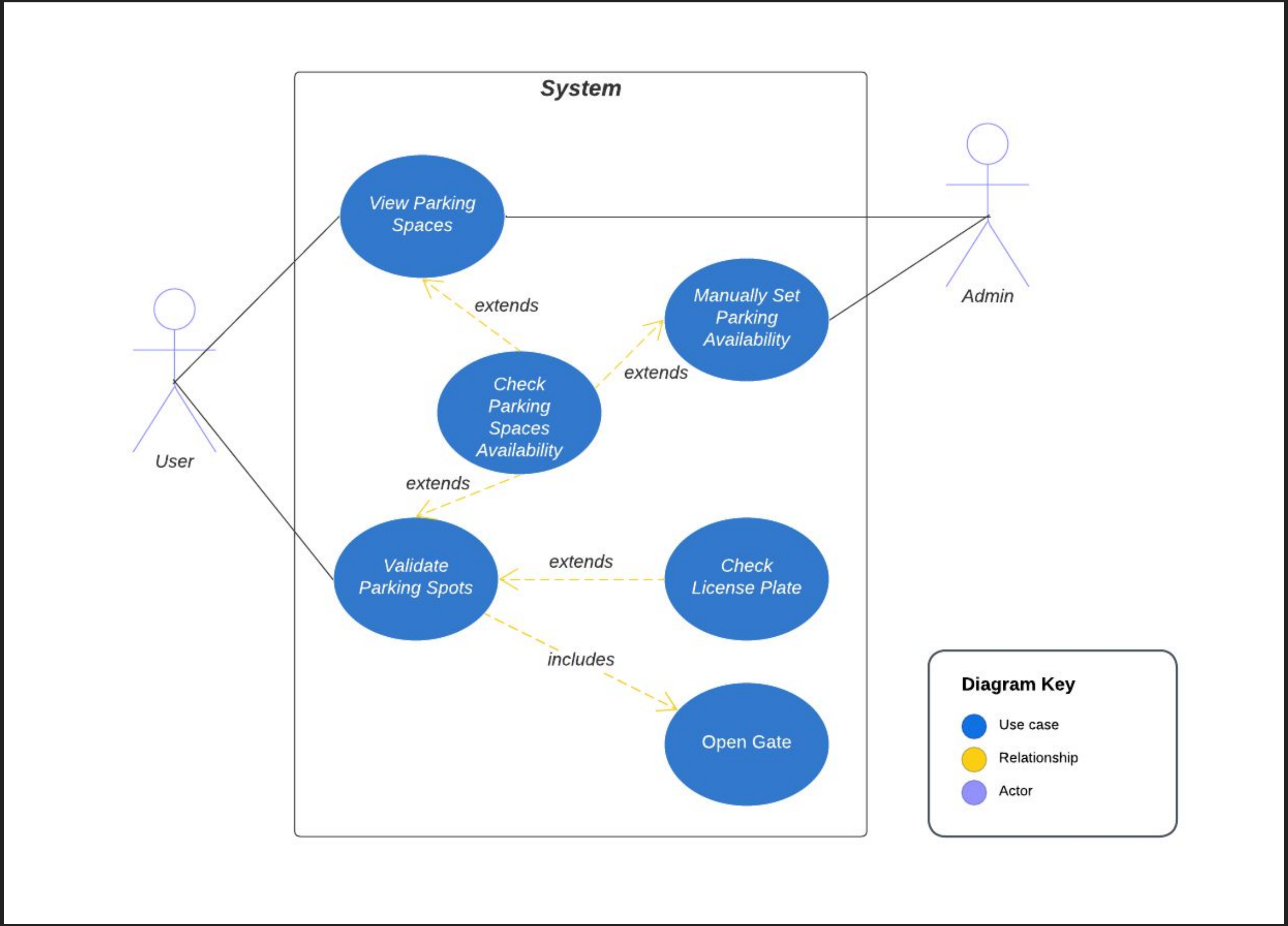
Fundamental Requirements

- Create End of Day Summarized Log Files
- Allow Admins to modify Parking Occupancy
- License Plate Validation
- Detect Available Parking Spots using Vision
- Validate Correct Parking based on Parking Permit Tier

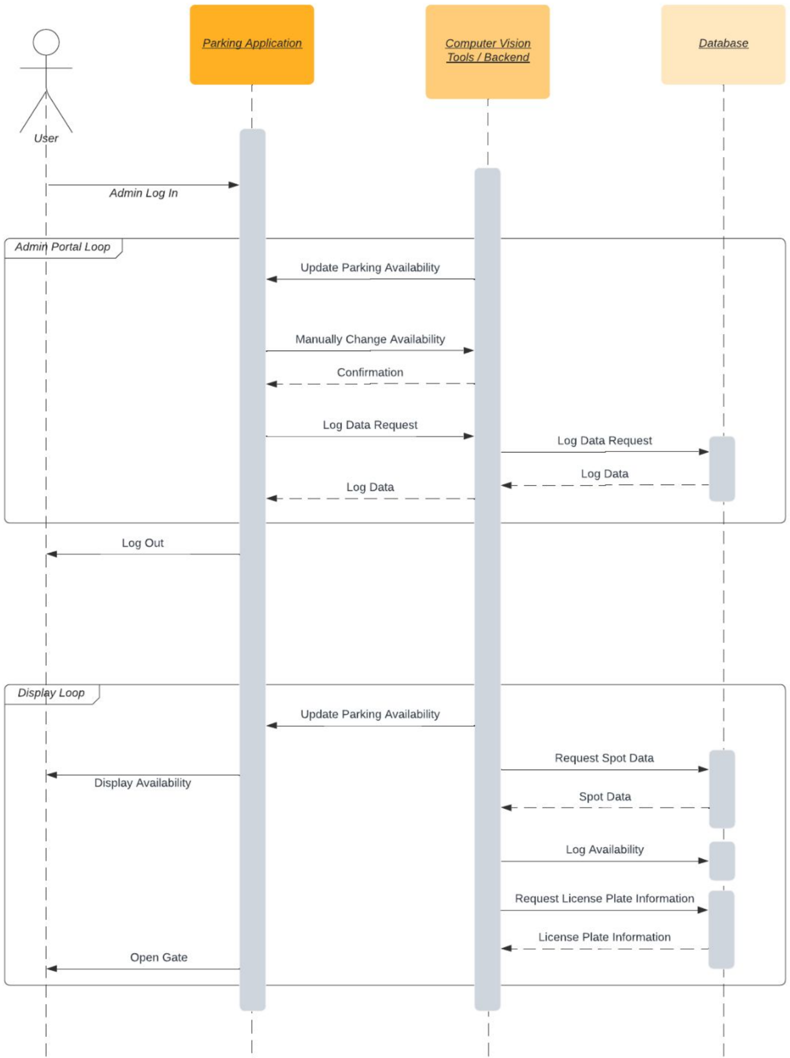
Non-Fundamental Requirements

- Most important Non-Functional Requirement
- Efficiency - System Response time > 5 seconds
- Security - Should pass OWASP Application Security Verification Standards
- Regulatory - Should Comply with EU GDPR, GLBA Act, PIPEDA, and CCPA
- Additional requirement included in the report with in-depth explanation

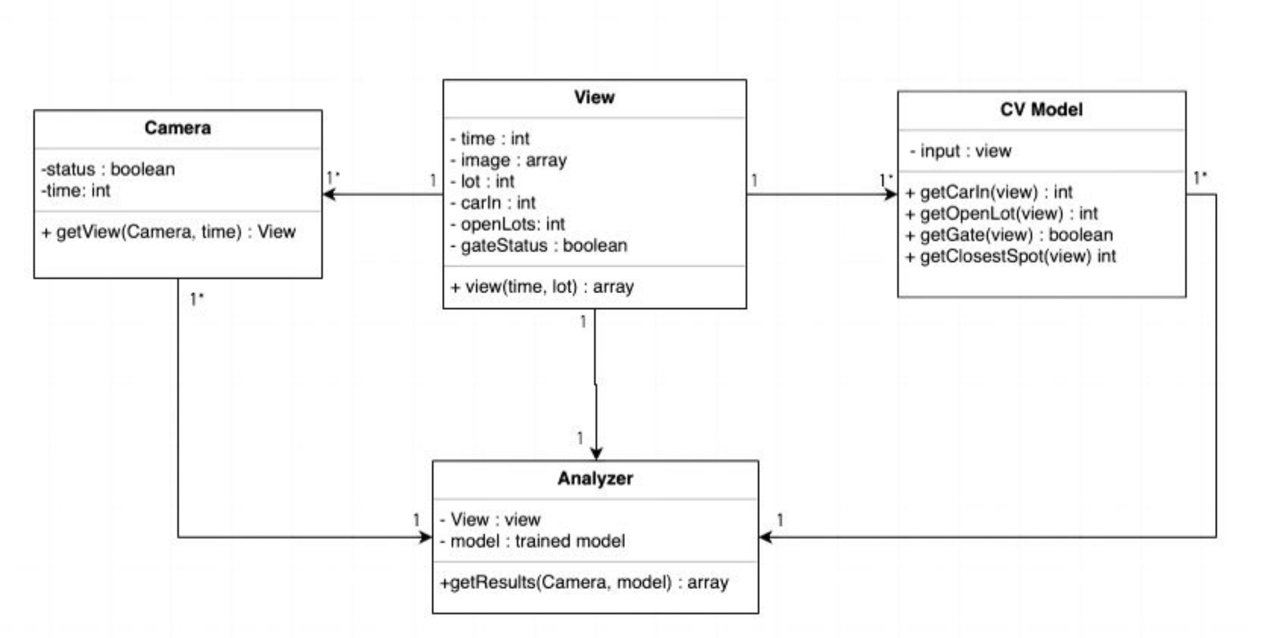
Use Case Diagram



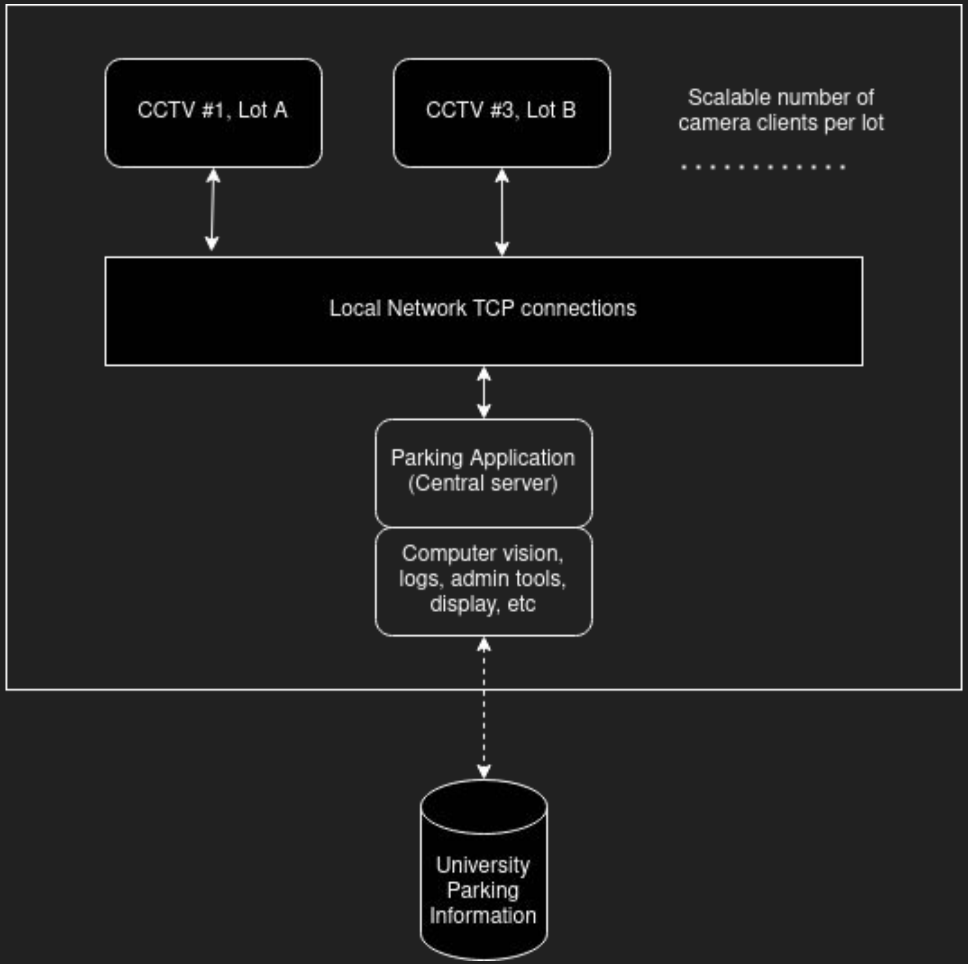
Sequence Diagram



Class Diagram



Architectural Design (Client-Server)



Project Scheduling

- Split into 2 Phases
- Phase 1 (February - May) :
- Research, On-site Evaluations, Verify Inventory, Price Quotations and Planning
- Phase 2 (May - September):
- Develop Software, Establish Database, Train Vision and ML Model
- Installation of Hardware Components, Build and Verify Prototype, Bug Fixing, Staff Training

Cost Estimation

- Calculated using Function Point Estimation
- Estimated Function Points of about 307 points
- Required time to complete - 1380 hours
- At 35 hours/week with a 4 person team can be completed in 2.5 months *

*Software Development

Cost Estimation

Products	Cost
Labor (Software Developers + training)	~$250,056
Hardware	~$285,250
Server	~$20,000
Total :	~ $560,000

Hardware and Software Cost Estimation

- One time Hardware Cost based on research and Guestimation
- Costs includes Servers, Outdoor Cameras, WIFI Access point routers and Outdoor Video Screen Panels and Labor
- Majority of Software is Open Source

Labor Cost Estimation

- Average Texas Software Developer Pay - $45.3 /hr
- Software can be integrated into University Office of Information and Technology
- University staff and Student Workers can be trained to use software at no additional cost.

Conclusions

- Comparisons
  - Many research papers that build low-cost simple smart parking systems with website and app interfaces (lacking features)
  - Industry smart parking systems (costly)
- Future Work / Changes
  - Integrate with interface(s) to display information
  - Addressing policy issues related to using a university’s cameras and infrastructure
  - Use UDP-based protocol instead of TCP