Design Document

Senior Design

Group May 13-17

Group Members:

Eric Cheatham, Michael Flagg, Intae Kim, James Sampica, David Turner

First Draft

Table of Contents:

Introduction

Executive Summary

Monitoring parking lots at universities like Iowa State is a time and resource intensive task. Parking Division Vehicles are sent out to monitor parking lots and catch parking violations. This costs man hours through staffing as well as vehicle upkeep and maintenance. Many violations do not get ticketed because there is no enforcement present to monitor the lot.

Statement of Purpose

The purpose of this project is to develop a system consisting of a series of centrally monitored image capturing platforms capable of identifying vehicles as they utilize campus parking lots. Our client has three main purposes for this project:

1. Introduce timing and ticketing uniformity into currently existing ticketing system
2. Reduce Department of Public Safety's (DPS's) operating cost by reducing payroll overhead
3. Eventual elimination of parking decals for registered vehicles

Definitions

JDBC – Java framework for database connectivity
MySQL – Database management system to be used for persistent storage
Berkeley Socket - C based API used to establish both TCP and UDP server/client communication
Java Socket – Java class used to establish connectivity between machines
OpenCV – Open source library of functions for real time image processing
Tesseract-OCR – Open source Optical Character Recognition engine
Back-illuminated CMOS – An image sensor design that increases low-light performance
GPU – Graphics Processing Unit. Accelerates image processing functions

Operation Environment

The software portion of our project will developed in C, Java, and OpenCV. The C and OpenCV code will be developed specifically to run on the Broadcom BCM2835 System on Chip (SoC) found the Raspberry Pi. All Java code will be used on a centralized server to provide both a data warehousing solution and a web-based client monitoring system.

Intended Users

There are two intended users for our system: Iowa State DPS and anyone parking any vehicle in an Iowa State University controlled parking lot. The vast majority DPS's interaction with the system will be through a Java based web-portal to monitor lot use across campus. Use of the system by vehicle operators will be limited to having their license plates imaged by our remote devices located at the entrances to the parking lots.

Intended Uses

The vehicle sensor will be placed at the entrances and exits of parking lots on the Iowa State University campus. They will be used to photograph the license plates of passing vehicles and pass license data to a database server. The end user will then be able to access data via the web portal.

Requirements

Business Requirements

- The system will have a total cost of less than $20,000 USD
- Ability to retrieve data collected through the front end interface
- Ability to determine who gets a ticket and display that information

Technical Requirements

- Solution will be able to make use of existing campus-wide wireless networks
- Solution will have some system for persistent data warehousing

Objectives and Approaches

Objectives

- Capture an image of a license plate using motion detection
- Locate license plate and convert characters to text
- Send information over network to a server
- Check license plate against database
- Determine if car should be ticketed
- Store images and time data
- Display information in user interface

Design Approach

Determine the necessary functions of the device and separate them into modules.

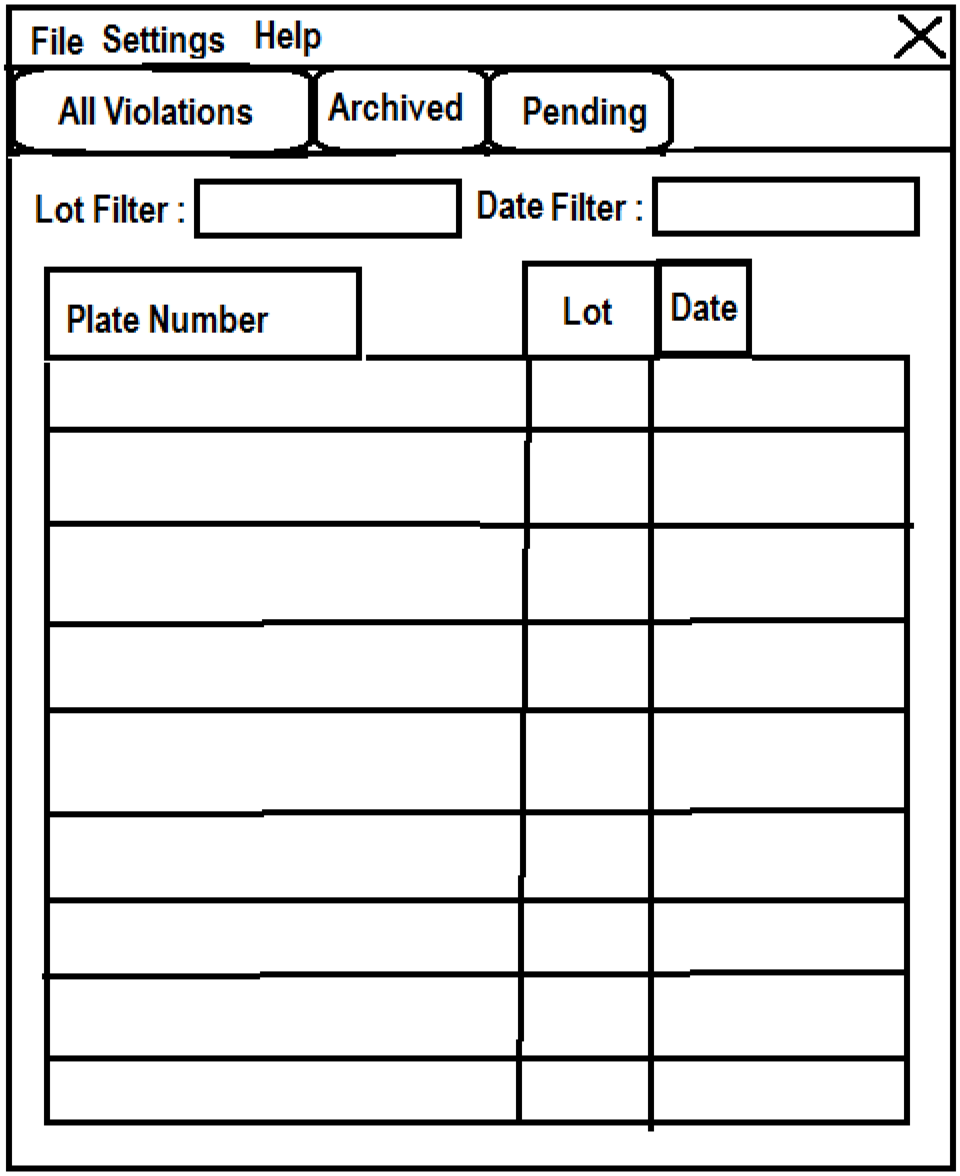
Development Approach

The project modules will be implemented independently but with focus on integration with other relevant modules.

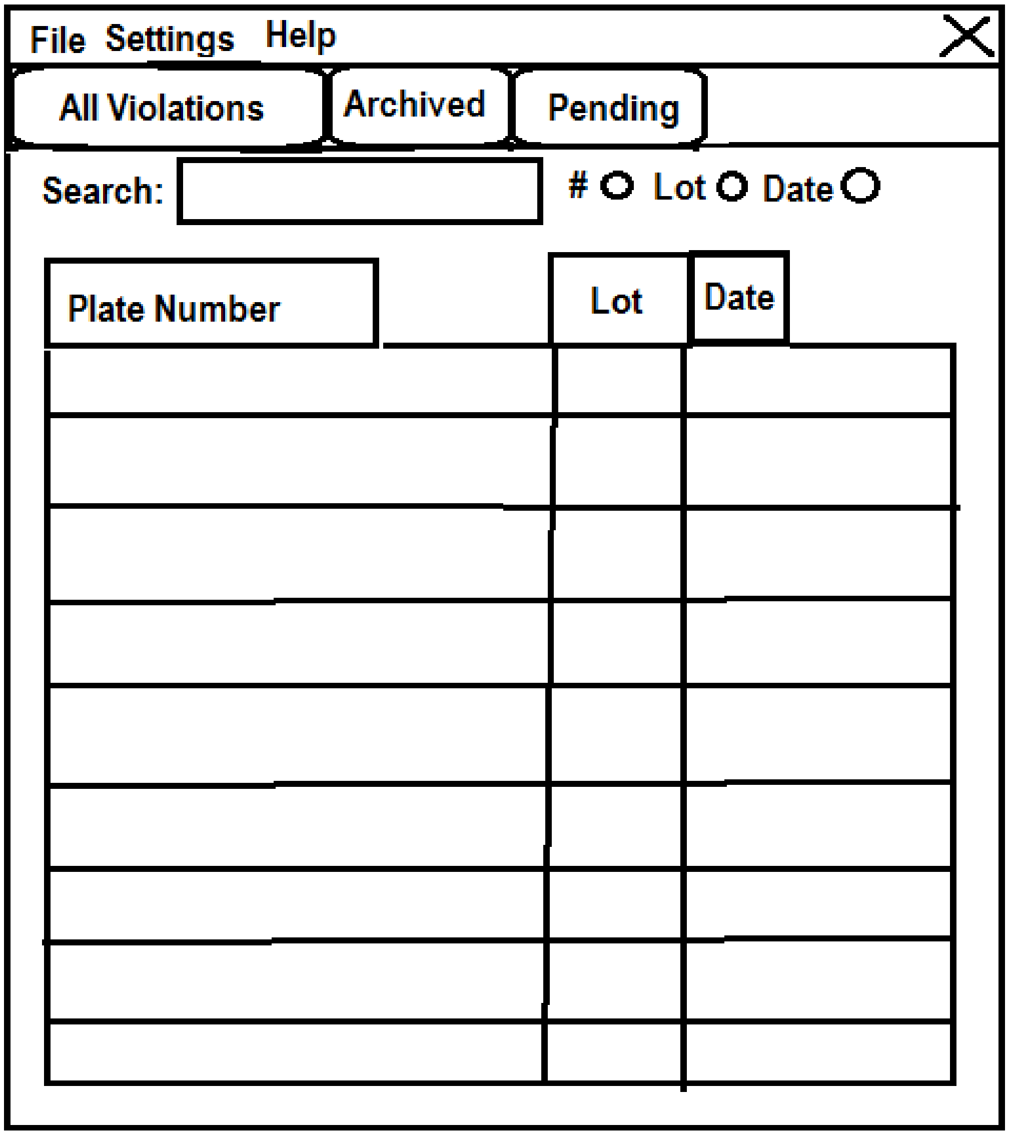
Validation Approach

All major design decisions are to be verified with the client before proceeding. All team members will test their own modules and final testing will be done collaboratively. This final testing will be thorough and attempt to cover edge cases where reasonable and possible.

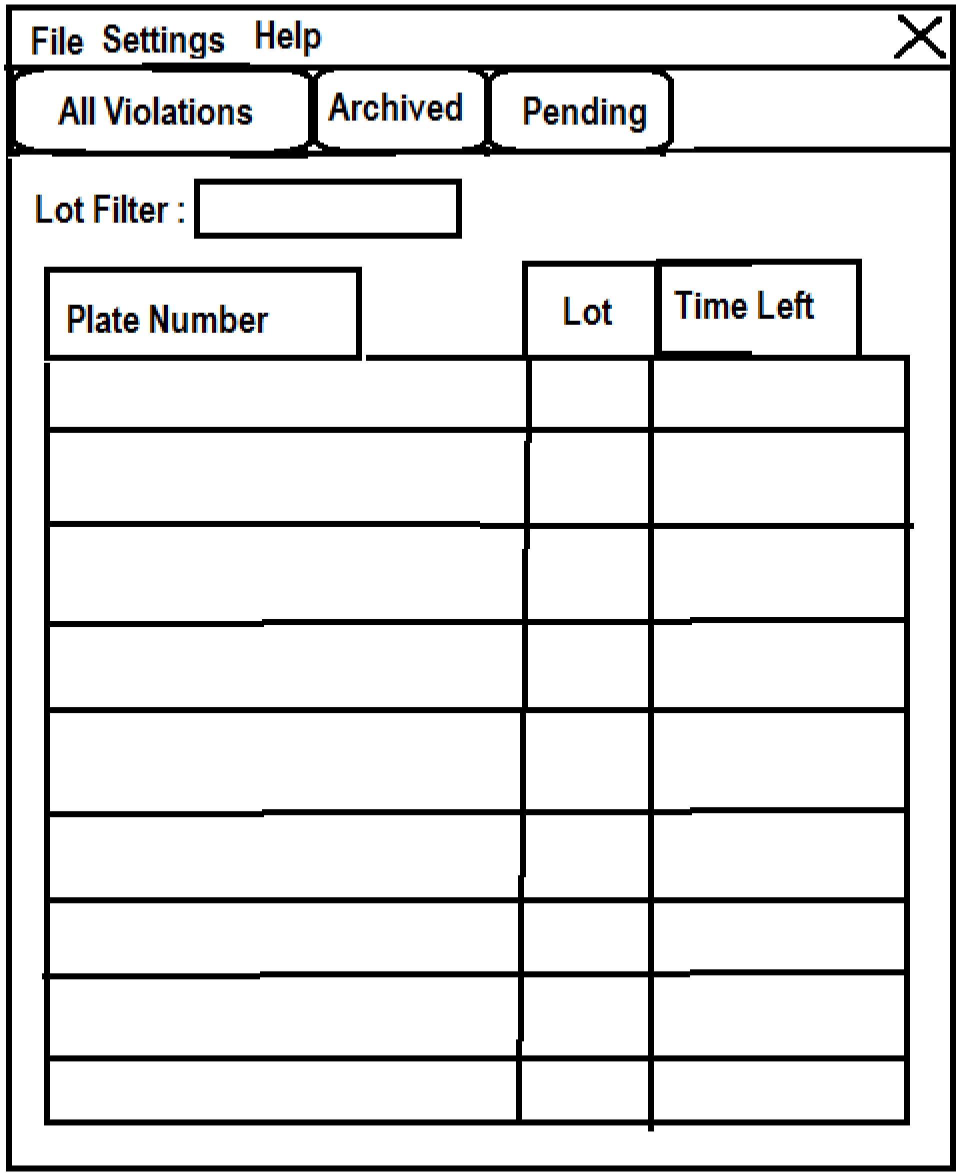
Non-Technical Design



This is the main view tab, “All Violations”. This will contain a list of the most recent violations and display information about that violation such as the plate number, lot, and date. The user can also filter by date and by lot to drill down details. The list size is static and when it fills up the least recent violation is dropped off the list.



The “Archived” tab allows the user to query the database to get a certain set of violations based upon what they have searched. The list is initially empty and populates when they have entered information and selected a radio



The Pending tab would show the user what license plates are pending for a ticket after their grace period has ended. It would contain a list of plates with corresponding lot and time amounts that count down. If the vehicle leaves or the timer counts down to zero, the vehicle is either removed entirely or removed and moved to “All violations”.

Subsystem Overview

Backend Modules: Front End client (user interface), Server, Database

User Interface module

The user interface serves as an information and data outlet for the data collected by the sensors. It makes requests to the server module for information.

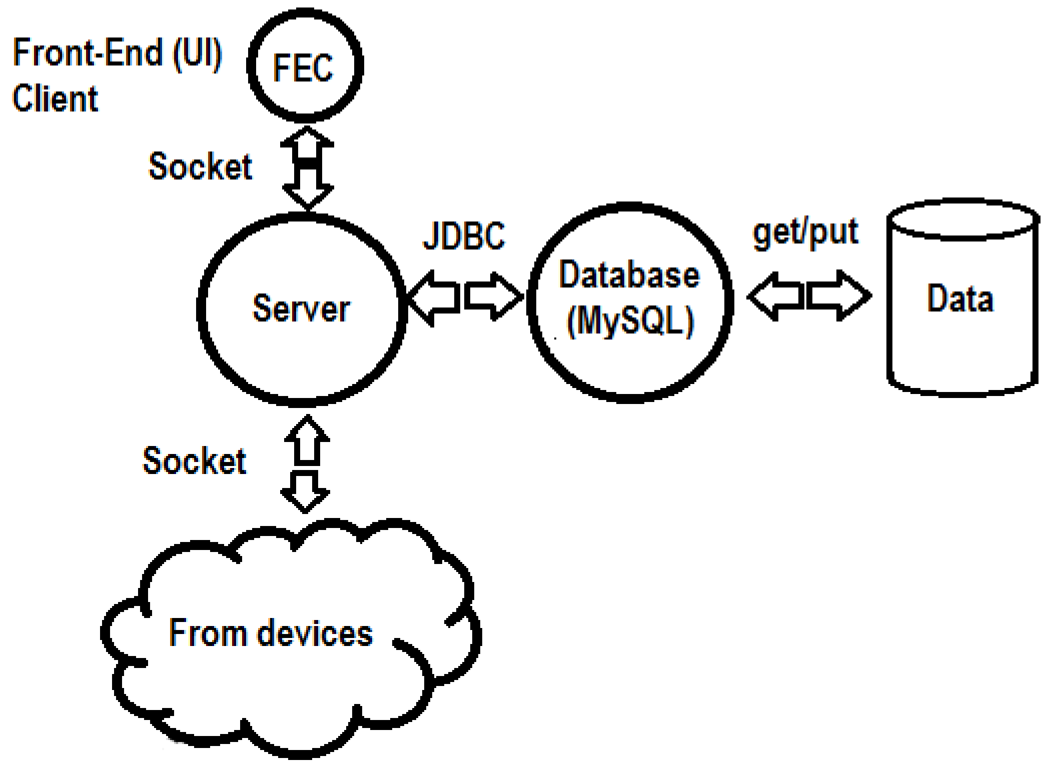
Database module

The database module stores and retrieves data stored by the sensors. It will send and retrieve its data to the server.

Server module

The server module acts as a delegate between the user interface module and the database, retrieving data from the database module and sending it to the interface to be displayed. It also acts as a delegate between the database and the sensors, collecting data from the sensors and giving it to the database module.

Architectural Diagram



Module Decomposition

This section of the document will provide an overview for each major module of the project. Each module will be discussed in further details in their respective sections of the document.

Image Capture Module

This module consists of a 5MP Back-illuminated CMOS camera sensor. It uses GPU accelerated scaling to resize the captured images to a size adequate for processing. If motion is detected in the low-resolution video stream, the camera will take a high quality still image and forward it to the image processing module.

Image Processing Module

The image processing module has two main functions. First, it takes a license plate image from the image capture module and uses OpenCV to apply filters to the image. It also applies and algorithm to localize the plate and the characters in the image. Second, these characters are sent to Tesseract, which uses OCR to translate these characters into plain text. The license plate text is then sent to the networking module.

Networking and Communication Module

The networking and communications module will provide secure communication between the central server and the devices located at various parking lots. This module will be utilized by the OCR module to send processed images and associated metadata to the central JDBC server for warehousing. Along with sending information pertinent to the image, the networking and communication module also identify the sending device to the receiving server.

Sever and Database Module

The database portion of the SDM module stores and retrieves data stored by the sensors. It will send and retrieve its data to the server portion. The server portion of the SDM acts as a delegate between the user interface and the database, retrieving data from the database module and sending it to the interface to be displayed.

Standards that Apply

Portable Operating System Interface
(http://www.opengroup.org/austin/papers/backgrounder.html)

The Portable Operating System Interface (POSIX) standard was created to ensure portability of code between operating systems. Code portability is achieved by defining a clear set of services that every POSIX compliant systems are expected to provide. POSIX acts as an umbrella for a large number of IEEE standards that define operability across multiple platforms.

Java SE Application Design with MVC
(http://www.oracle.com/technetwork/java/codeconv-138413.html)

(http://www.oracle.com/technetwork/articles/javase/mvc-136693.html)

The Java coding standard will mean it would be easy to read and for somebody else to pick it up and modify it. The MVC standard means that we can easily change views or models depending upon what need arises in the future. MVC also makes the code much more modular so it’s more maintainable.