

**Final Report**

**Team Tech Ops**

**Team Members**

**Jeswin Abraham, Alex bates, Phillip Bouie, Naga Gattupalli**

**Abstract**

Parking Monitoring system is a system which is used to monitor parking lot

spaces from a high elevated camera. It means that the system will take a few of

images and compare them to see if there is any parking slot occupied or not. The

system makes use of the images to keep track of the available spaces and display the

output of the parking space availability for the potential drivers who intend to park

in a selected parking lot. When a parking space is occupied, the system updates the

parking lot by removing the occupied space. This helps users in reducing their time

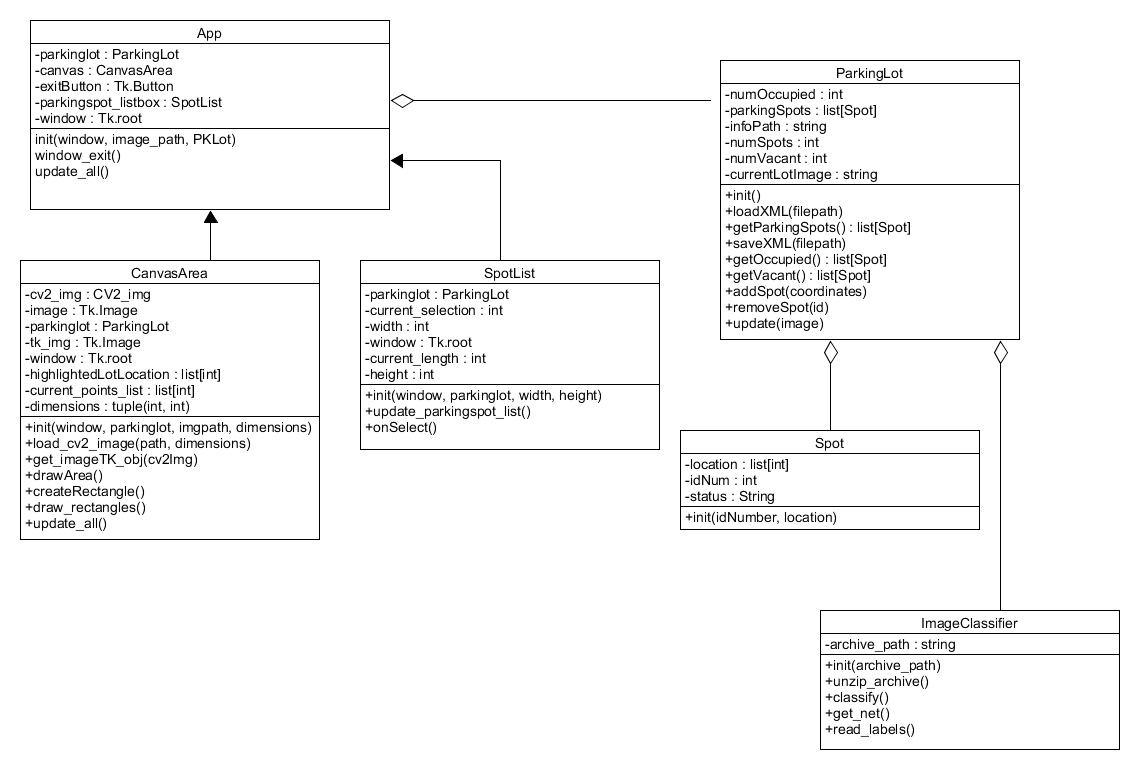
to search for an available parking space.

**Table of Contents**

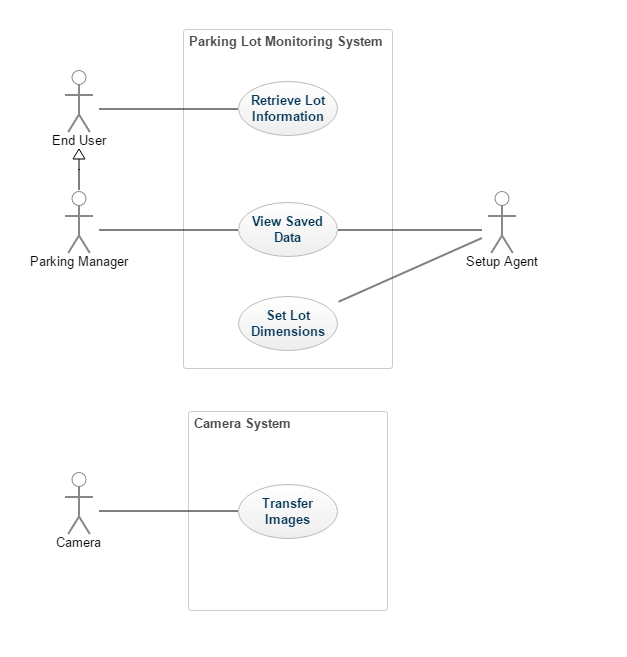
* Introduction
  + Motivation/Background
    - parking lots suck
    - hard to find spaces (kennesaw is crowded)
  + Planned end product
    - thing
  + Current end product
    - thing
  + Explanation of differences (here you can explain issues with technologies along the way (i.e., lessons learned) or, perhaps, strategies that would better serve someone working on this project in the future
    - technical challenges
    - time limits

Technical Documentation

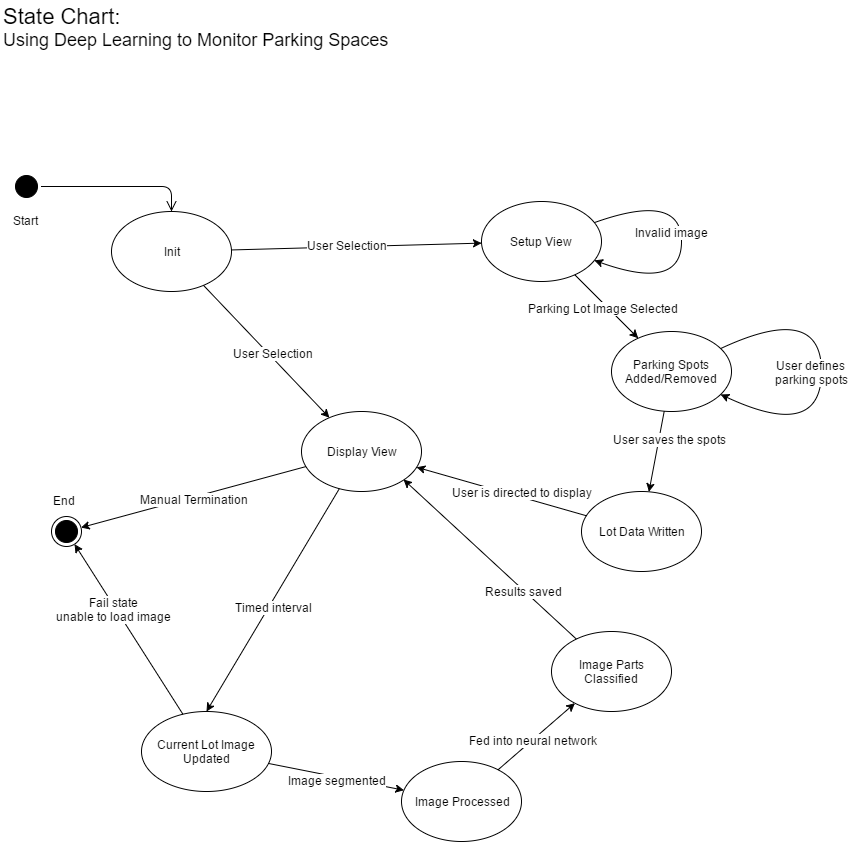
* + Software Design
    - Below are relevant documents used to shape the final software product, and a screenshot of the main interface.
    - **Class Diagram**

****

* + - **Use Case Diagram**

****

* + - **State Diagram**



* + Tools used
    - Tools used during this project include the following:
      * Python 2
      * Python standard libraries
      * Caffe (Berkeley)
      * OpenCV
      * Ubuntu
      * DIGITS (Nvidia)
  + Dependencies/Assumptions
    - The software developed during this project assumes a proper operating environment to function. This includes a capable computer, with a CUDA capable graphics card. All dependencies listed in the user manual are correct at the time of writing, but updates by third parties may change this in the future.
    - It is also assumed that the system will be provided with the appropriate quality and type of images. Very low resolutions or very shallow angled images of parking lots will affect accuracy negatively.
* Testing
  + Software Functionality
    - Testing of software was done as each piece of functionality was added. The tests were primarily to check for general errors and completeness of behavior. As features were added, this included testing of file formatting in the case of saved data, such as parking lot layouts and usage information.
  + Results and Accuracy
    - Testing the accuracy of results was very limited, partly due to time constraints. Generally, it can be said that accuracy can be greatly affected by the angle of the image being used, as well as light levels within the image. Low levels of light and heavy shadows on the ground cause inaccurate results, and low angles cause cars in one parking spot to block the view of parking spaces behind it.
  + Results of testing
    - Testing revealed that improvements could be made on both fronts. Limitations in the software exist that lead to a less than ideal user experience, and accuracy from images could be improved by retraining the classification network, as well as performing preprocessing to make the image clearer in the case of lighting problems. However, overall the end software does accomplish the tasks that were laid out in the proposal and SRS.
* Future work
  + Where can this project go in the future?
    - Better stuff with statistics tracking
    - increased accuracy/counting cars instead of laying out parking lot
    - external monitoring for vehicle operators
    - mobile application
    - improved UI/UX
  + What are the next steps for your project if you were to continue working on it?
    - Statistical analysis
    - increase accuracy of network model
    - speed improvements (multi threading)