**See Spot Park**

by:*ParKings*

Tyler Hurson

David Rodriguez

Koltan Weaks

Thomas Wurdinger

**Version 1.0**

**10/18/2016**

**Table of Contents**

|  |  |
| --- | --- |
| 1. Introduction | 2 |
| 1.1 Background  1.2 Purpose  1.3 Overview  1.4 Contextualization |  |
| 2. Project Objectives | 6 |
| 2.1 Scope  2.2 Goals |  |
| 3. Design Strategy | 8 |
| 3.1 Technical Overview  3.2 General Approach  3.3 Alternative Designs |  |
| 4. Project Plan | 12 |
| 4.1 Overview  4.2 Project Task Set  4.3 Server Task Set  4.4 Camera System Task Set  4.5 Laser Break System Task Set  4.6 Mobile Application Task Set  4.7 Task Prerequisites Chart |  |
| 5. Evaluation Criteria | 21 |
| 6. Schedule | 22 |
| 7. References | 27 |
| 8. Glossary | 28 |

**1. Introduction**

1.1 Background

Students at Texas A&M University-Kingsville face a great number of challenges before they may graduate. They have grades to maintain, deadlines to meet, extracurricular activities, and vast amounts of studying to contend with on a daily basis. Unfortunately, parking can be added to the list of headaches that students and, indeed, faculty must deal with on a daily basis. When asked, many students relayed to our group their dissatisfaction and, sometimes, downright loathing for the parking situation on campus. For now, there exists very few viable options to each student and faculty member at TAMUK:

* Arrive to campus extremely early and hope that there is an available parking space, but probably still have to drive around searching for said space.
* Park off-campus and walk to wherever it is they were headed. This option might make them late depending on how far away they have parked. This might be a few blocks because it is illegal to park in the surrounding neighborhoods on the weekdays.
* Arrive on time to campus with everyone else and definitely have to drive around looking for an open space. This option will, most likely, also cause the student or faculty member to be late. There is an inherent flaw in this method as well. A person can only see the spaces they are driving past. If a space becomes available after they had left the area, they would not know until they had completed a full circuit around campus. By then that space would be occupied by any number of people that were also searching for a place to park.
* Ride a bike.

With the right planning, of course, anybody might avoid being late. But why should someone have to leave their home an half hour to an hour early just to find a parking space and then sit and wait until their class or meeting starts?

Our team, ParKings, believes we can provide a better option. It would be an option that allows the students and faculty of TAMUK to monitor the parking situation on campus in real time and avoid the “drive-around”. By using vehicle recognition software and a network of I.P. security camera modules, this system would be able to monitor the designated parking lots across the campus, then provide live information on the availability of parking in that area. All this information would be accessible via our free mobile app. To avoid any damage or liabilities, this app would include G.P.S. recognition and a speech component to relay the location of the parking spot to the user without them having to look at their phone.

Image recognition software is becoming more and more prevalent in today’s connected world. The most visible aspect being the facial recognition aspect of the software. On the high end of things, there is IBM’s deep learning and its massive neural net. Deep learning uses the entirety of the internet as its base for information retrieval. This allows for a much broader spectrum of objects that can be identified. This option, however, has its time limitations. Any information would have to be sent to IBM, placed in a queue, processed in turn, then returned to us. To provide the most recent data to our users, the vehicle recognition software will be created by our team and housed on our server. Using a method called Viola-Jones feature detection, our software would be able to accurately recognize and count vehicles as they entered or exited a parking area. The software would recognize the features of vehicles only, ensuring that students on foot or cyclists do not render a false positive.

Once we have succeeded, a student may choose the new option of leaving early if they need to, or just in time because they are confident in their ability to find a parking spot thanks to their new mobile app.

1.2 Purpose

The purpose of this project is to alleviate the parking problem on the TAMUK campus by providing students and faculty with live information on the availability of designated parking spaces around campus via an easy-to-use mobile application.

1.3 Overview

Several parking lots around campus will be chosen as designated parking lots. Each designated parking lot will have a total capacity and a counter. The capacity will represent the maximum amount of vehicles that the parking lot can hold. The counter will represent the total amount of vehicles in the parking lot at any given time.

Each parking lot will include two cameras mounted at each entrance placed about three feet off the ground. Several laser break devices will also be mounted to each entrance. When the lasers are tripped, the cameras will capture a video of the passing object. These images will be transmitted to a central server and processed through an image recognition algorithm to determine whether or not the candidate object is a valid vehicle based upon some predetermined criteria. If the passing object is a valid vehicle, the counter associated with the parking lot will be incremented or decremented, depending on the direction of the vehicle.

Customers access the system through a free, downloadable mobile application. The application consists of a map of the TAMUK campus containing each designated parking lot (Fig. 1.1, 1.2). A large number above each parking lot will represent the total number of free parking spaces available per lot.

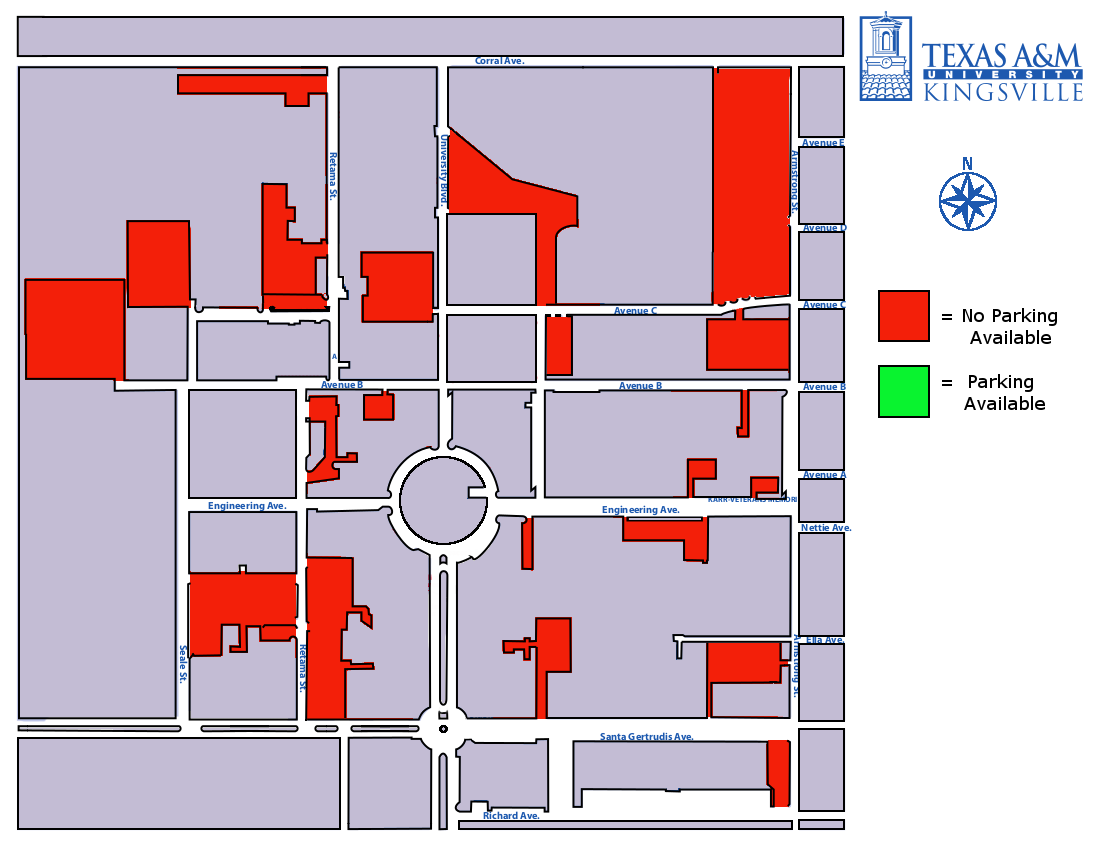


Fig 1.1 - There is no parking available.

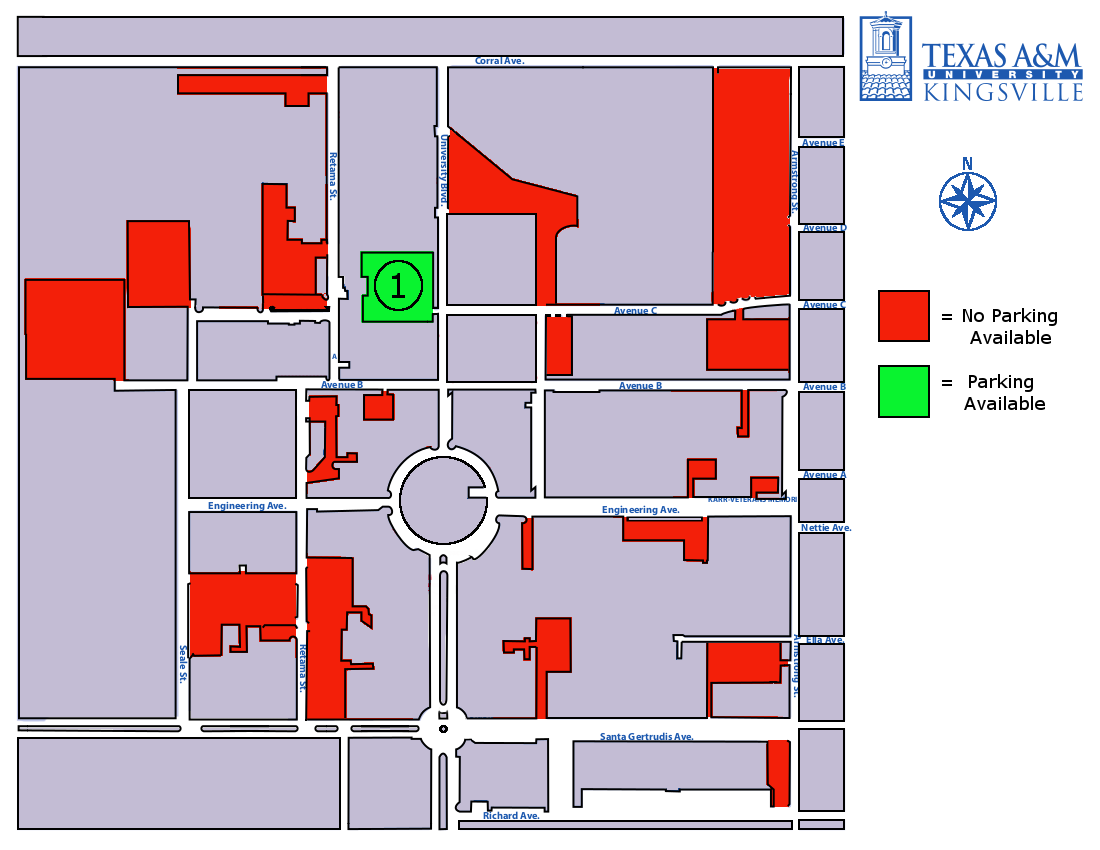


Fig 1.2 - A spot has opened up.

Due to inherent problems with long-term accuracy, the counter of each parking lot will have to be periodically reset. During the after-school hours (midnight - 6 AM), a set of overhead cameras will capture an image of the entire parking lot and use this image to determine the actual number of vehicles in the parking lot. The counter will be reset to this value.

1.4 Contextualization

This technology is currently most used in parking garages. There is no single solution, though. For each different garage and business, a new solution must be found to ensure an accurate count of the vehicles.

RedStorm has created a system that keeps a count of open spots in a parking garage using infrared sensors to detect and keep track of vehicles entering and leaving the parking area. One thing that differs from other developed systems, other than the different hardware and system used, is the street parking spot counter. There hasn’t been any other open projects that demonstrate any sort of systems that keep track of parking spots that are alongside the street.

**2. Project Objectives**

2.1 Scope

This project will be completed over a period of approximately eight months.

The physical scope of the project is restricted to the TAMUK campus. Initially, the system will target enclosed parking areas only. Furthermore, the system will target only student, commuter, and faculty parking, since these are the most contested parking areas. Residential parking lots will be omitted due to their size.

The prototype system will be the full implementation the design, restricted to a single parking lot. It will be constructed on the Engineering Complex parking lot (the parking lot across the street from the Engineering Complex, next to the Physical Education Center).

If the prototype system is completed earlier than scheduled, the system may be expanded to more than one parking lot. Furthermore, street parking (i.e. non-enclosed parking) may be targeted. However, this will require a significantly different approach.

For the mobile application, only Apple and/or Android devices will be targeted, since these platforms command 25% and 69% of the total market share respectively.

At the completion of the project, the prototype system will be complete and fully functioning. Students and faculty will be able to download the accompanying mobile application and view the live status of designated parking lots.

2.2 Goals

We have defined six system quality attributes that will help shape the design of our project. Our primary goal is to ensure that these attributes are properly reflected in our design with respect to their priority. Here the attributes are listed in descending order, from most important to least important:

1. **Precision** - The system must be as accurate as possible. Otherwise, it is useless to the customer.
2. **Availability** - The system must always be available during school hours.
3. **Usability** - The application must be easy to use and not overly complex for the customer.
4. **Scalability** - The system must be able to be implemented in any enclosed parking lot with a discrete number of entrances.
5. **Durability** - The physical components of the system must be able to withstand extreme weather conditions.
6. **Maintainability** - The system must be able to run even after the original developers have graduated.

In short, we plan to create accurate vehicle recognition software. Concurrently, we will be designing, building, and distributing weather/tamper resistant camera modules to designated parking areas around campus. Once we have modules at every entrance and exit, we will be able to use the data we gather to provide an easy-to-use application for students and faculty that has an accuracy of 1-2 vehicles.

**3. Design Strategy**

3.1 Technical Overview

The overall system will consist of four components: a central server, the camera system, the laser-brake system, a variable number of mobile applications.

The server contains the database and the main application. It is responsible for storing system data, receiving data from the camera system, and sending data to mobile applications.

Static data on the server will include programming source files and libraries. Dynamic data will include the capacity and counter for each parking lot. Dynamic data will be stored on a database.

The main application implements the domain logic and interfaces with the database. It handles data requests and responses and updates the database accordingly.

The camera system consists of a set of cameras physically mounted around designated parking lots. The cameras will be outdoor cameras with a built-in WiFi capabilities. The camera system consists of two subsystems: the entrance camera system and the overhead camera system.

The entrance cameras are responsible for capturing images of vehicles as they enter and exit each parking lot and sending the images to the server. The cameras will be placed 10 cm-2 m from the curb of the parking lot entrance on either side of the entrance, facing each other, centered between the laser break devices. They will be attached to a metal pole that will rise 0.75 m from the ground. As an object trips a laser-break device, the entrance cameras capture the passing object as a series of images and send the images to the server to be processed (Fig 3.1).

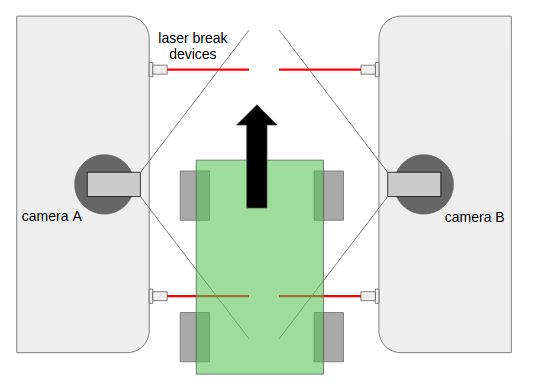


Fig 3.1 - As a vehicle passes through the parking lot entrance, the lasers are tripped and the cameras capture the video of the passing vehicle.

Overhead cameras are responsible for capturing images of entire parking lots and sending the images to the central server. The cameras will be placed several meters above the parking lot on a nearby pole or building. The number of overhead cameras will differ per parking lot depending on the geometry of the lot and the placement of nearby poles and buildings. Larger parking lots and non-rectangular shaped parking lots may require more than one overhead camera.

The laser break system will consist of several laser break devices mounted at the entrance of designated parking lots. Each entrance will contain two rows of laser break devices. When a device is tripped, it will send a signal to the entrance cameras indicating that they should begin to record and transmit images to the central server for processing.

The mobile application is the customer-facing portion of the system. The mobile application consists of a map of the TAMUK campus containing each designated parking lot. It is responsible for sending data requests to, and receiving data responses from the central server, and updating the map with the appropriate information.

3.2 General Approach

For this project, we will be following a traditional Iterative process model. Due to our relative inexperience with the technologies involved in this project, we expect some of our requirements to be inaccurate. Therefore, we may need to revisit our requirements once we begin to implement the project.

To maximize work output, we divided the project into several semi-independent sets of tasks. Each task set will be undertaken by 1-2 members of the team. The team will meet twice a week- once on Tuesday, and once on Thursday. The purpose of these meetings is to report on the progress of each individual team member and determine if the schedule needs to be revised. Communication is key.

3.3 Alternative Designs

Several alternative approaches to counting vehicles were considered for the project. These methods were rejected because they did not meet our project goals.

One method of counting vehicles involves an induction loop. A wire is placed at the entrance of each parking lot. A current is sent through the wire, creating a magnetic field. As a vehicle passes over the wire, the magnetic field changes. The system would detect the change in magnetic field and update the counter to the appropriate value. This design was rejected due to its cost and complexity.

Another method of counting vehicles involves a set of overhead cameras. Cameras are placed at locations above each parking lot. Each camera captures a set of images which, combined together, model a 3D space. Vehicles would be counted using object recognition technology. This design was rejected due to its complexity and non-scalability.

A final method of counting vehicles involves a laser break system. In the event that our camera system fails, or the image recognition algorithm does not meet our accuracy requirements, we will fall back upon this alternative counting system.

In this system, the entrance cameras are done away with entirely, and the counting of vehicles is done exclusively by the laser break system. There are three rows of laser break devices instead of two. The counter for each parking lot is incremented or decremented depending on the order in which the rows of devices are tripped. Figure 3.1 illustrates the system.

Figure 3.2 is a state diagram modeling the logic of the system. The system begins at state 0, and waits for either row A, B, or C to be tripped. As the lasers are tripped, the system enters a new state depending on which row was tripped. For example, a vehicle tripping C, followed by B, followed by A would indicate that a vehicle has left the parking lot (prompting the system to decrement the counter by one). Likewise, a vehicle tripping A, followed by B, followed by C would indicate that a vehicle has entered the parking lot.

The logic also handles two vehicles entering and exiting at the same time. However, a major flaw of the system is that one vehicle may “mask” another vehicle by blocking the lasers that would detect it. Due to this problem, we decided to settle upon a camera-based system. Though a camera-based system is more expensive and sophisticated, it will prove much more reliable, and therefore much more accurate.

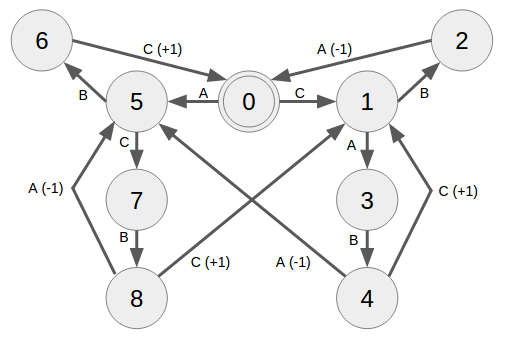


Fig 3.2 - State diagram for laser break counting

**4. Project Plan**

4.1 Overview

The project consists of four major task sets. Each task set encapsulates the development of one of the four major components of the system (the central server, the camera system, the laser-break system, the mobile application). In general, each task set may be completed independently, though there are some hard dependencies between each task set.

Each task set contains several tasks to be completed in sequential order. Each task builds upon the previous task, until a final product is realized. Each task is assigned a complexity level (1-10). This value is an estimation of how much time the task will take. A task with a complexity level of 10 may take several weeks to complete, while a task with a complexity level of 1 will take only a few hours to complete at maximum. The complexity level of each task will be used to determine an accurate schedule for the project.

The following sections list the tasks in no particular order. The Task Prerequisites Chart in section 4.7 provides a visual display of the relationships between tasks.

4.2 Project Task Set

|  |  |
| --- | --- |
| **Task** | **Complexity** |
| **Establish Problem/Solution**  Establish the problem to be solved. Determine if the problem is a real problem. Brainstorm potential solutions to the problem and the technologies that would be involved in solving the problem. Determine the required features of the system and determine if they fit within the time constraints of the project. Determine if the solution can be implemented on a reasonable budget.  *prerequisites: none* | 4 |
| **Determine Project Goals**  Determine the software quality attributes (reliability, maintainability, etc.) that are most important to the project.  *prerequisites: Establish Problem/Solution* | 2 |
| **Gather Project Requirements**  Establish the overall project budget. Determine a typical user’s expectations of the application. What are the functional requirements? What are the non-functional requirements?  *prerequisites: Determine Project Goals* | 3 |
| **Prioritize Features**  Prioritize product features from high priority to low priority. High priority features are a “must have” whereas low priority features a “nice to have.”  *prerequisites: Gather Project Requirements* | 2 |
| **Define Prototype**  Define the prototype system for the project. What is the scope of the prototype system? Where will it be located? What components will be involved?  *prerequisites: Prioritize Features* | 2 |
| **Gain Permission**  Gain permission from the parking authority or any other required authority on campus to ensure that the project may legally be completed.  *prerequisites: Define Prototypes* | 1 |
| **Develop Prototype A**  Develop the initial prototype of the system. Prototype A combines the camera system, the laser break system, and the central server to create a functioning (non-visual) application.  *prerequisites: Mount Laser Break Devices, Communicate with Laser Break System, Test Server Application, Gain Permission* | 5 |

|  |  |
| --- | --- |
| **Test Prototype A**  Test prototype A against a series of test cases similar to the ones used in testing the server application. Ensure that the system is reading vehicles correctly and retains its accuracy over at least 24 hours.  *prerequisites: Develop Prototype A* | 9 |
| **Develop Prototype B**  Develop the final prototype of the system. Prototype B combines the camera system, the laser break system,the central server, and the mobile application to create a complete, functioning visual application. This is the product that the customer will be using.  *prerequisites: Test Prototype A* | 3 |
| **Test Prototype B**  Test prototype B against a series of test cases similar to the ones used in testing prototype A. Ensure that the system is reading vehicles correctly and retains its accuracy over at least 24 hours. Ensure the mobile application is displaying the right data and there are no visual glitches.  *prerequisites: Develop Prototype B* | 6 |
| **Deploy Finished Product**  After the team is satisfied that protobyte B is in a working state, high quality, and well polished, launch the finished product.  *prerequisites: Test Prototype B* | 4 |

4.3 Server Task Set

|  |  |
| --- | --- |
| **Task** | **Complexity** |
| **Gather Server Requirements**  Establish budget, determine hardware, environment, and technology requirements, as well as any other requirements (reliability of server, customer support, etc.). Server requirements should be informed by project requirements.  *prerequisites: Define Prototypes* | 3 |
| **Explore Available Servers**  Investigate available academic and commercial servers and compare them against the established requirements. Calculate the monthly cost of each server and determine if it fits within the budget.  *prerequisites: Gather Server Requirements* | 2 |
| **Acquire Server**  Purchase/rent the chosen server. Document the server specs and the cost. Login to the server to confirm that we have access to it. Confirm that we can read/write to the server and that the server is connected to the internet.  *prerequisites: Explore Available Servers* | 1 |
| **Setup Server**  Put server files under version control. Install required technologies. Create database and configure schema. Establish static IP for server. Open up ports for remote access.  *prerequisites: Acquire Sever* | 3 |
| **Communicate with Camera**  Establish data communication between the camera system and the server. Ensure that the server is receiving images from the camera and can interpret them in the appropriate format.  *prerequisites: Setup Server, Acquire Cameras/Mount* | 4 |
| **Develop Image Recognition**  Develop the image recognition portion of the server application. Develop a program that will take in a series of images and determine if a valid vehicle exists in the images, and also determine the direction of the vehicle.  *prerequisites: Define Prototypes* | 8 |
| **Test Image Recognition**  Test the image recognition algorithm against a series of test cases using many different types of vehicles and non-vehicles. Adjust the algorithm if any inaccuracies or bugs are found.  *prerequisites: Develop Image Recognition* | 6 |

|  |  |
| --- | --- |
| **Develop Server Application**  Develop the application that will run on top of the web server that will handle data requests/responses, implement the logic of the system, and interface with the database.  *prerequisites: Test Image Recognition* | 6 |
| **Test Server Application**  Test the server application. Ensure that it is receiving input from the camera system and correctly identifying valid/invalid vehicles.  *prerequisites: Develop Server Application, Communicate with Camera* | 3 |

4.4 Camera System Task Set

|  |  |
| --- | --- |
| **Task** | **Complexity** |
| **Gather Camera System Requirements**  Establish the budget of the camera system. Determine the hardware requirements and communication requirements. Also consider physical environment requirements: Can the camera system withstand heavy winds? How much physical abuse should it be able to withstand? Determine the material that will be used to mount the camera and how the mount will be physically placed in the ground. Visual inspection of parking lot entrances may be necessary.  *prerequisites: Define Prototypes* | 3 |
| **Explore Available Cameras/Mounts**  Investigate available cameras and compare them against the established requirements. Investigate available camera mounts and compare them against the established requirements.  *prerequisites: Gather Camera System Requirements* | 2 |
| **Acquire Cameras/Mounts**  Acquire 1-2 cameras and 1-2 mounts for the prototype. Ensure that the cameras work as intended.  *prerequisites: Explore Available Camera Mounts* | 2 |
| **Mount Cameras**  Physically place the cameras on their mounts and place the mounts in the ground. Ensure that the cameras are facing the right direction and are properly configured.  *prerequisites: Acquire Camera/Mounts* | 7 |
| **Communicate with Laser Break System**  Establish data communication between the camera system and the laser break system. Ensure that the camera is activated when the lasers are tripped.  *prerequisites: Acquire Laser Break Devices, Mount Cameras* | 4 |

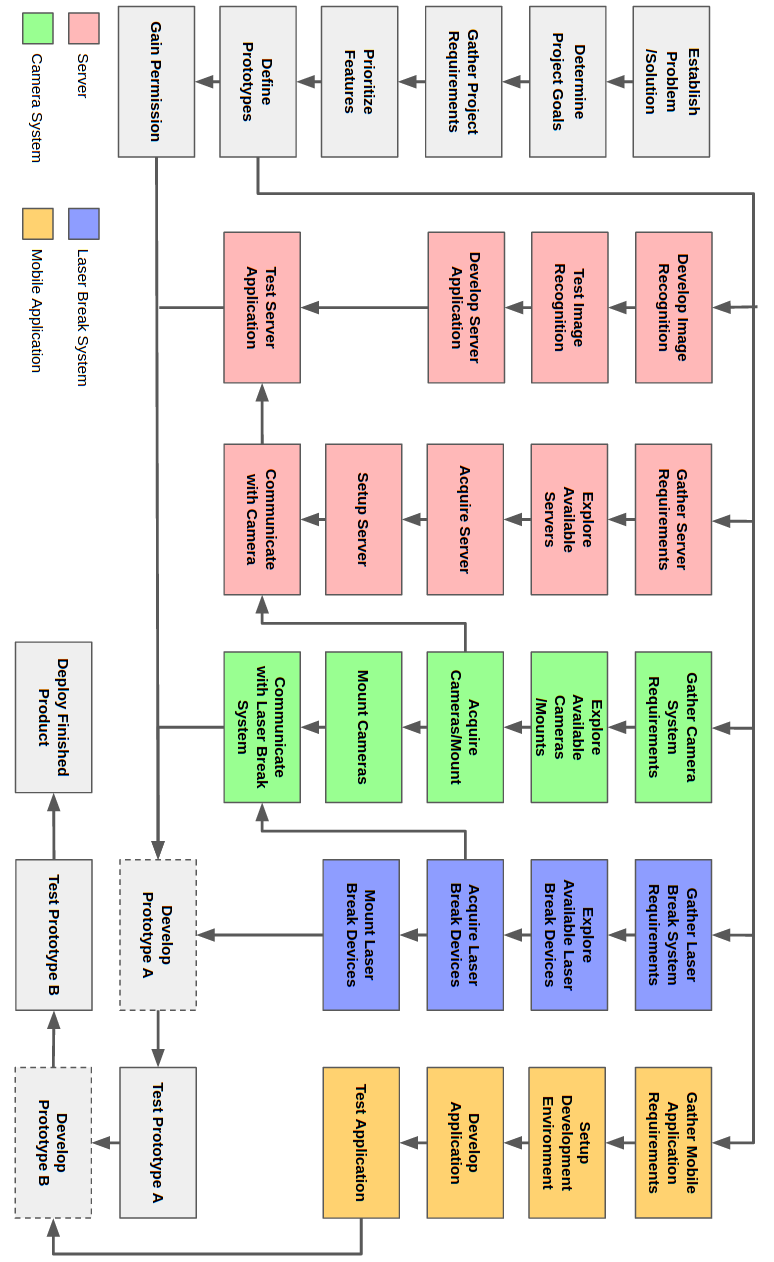
4.5 Laser Break System Task Set

|  |  |
| --- | --- |
| **Task** | **Complexity** |
| **Gather Laser Break System Requirements**  Establish the budget of the laser break system. Determine the hardware requirements and communication requirements. Also consider physical environment requirements: Can the laser break system withstand heavy winds? How much physical abuse should it be able to withstand? Determine how the laser break devices will be mounted to the curb.  *prerequisites: Define Prototypes* | 2 |
| **Explore Available Laser Break Devices**  Investigate available laser break devices and compare them against the established requirements.  *prerequisites: Gather Laser Break System Requirements* | 2 |
| **Acquire Laser Break Devices**  Acquire the appropriate number of devices as determined in the requirements.  *prerequisites: Explore Available Laser Break Devices* | 2 |
| **Mount Laser Break Devices**  Physically mount the laser break devices to the entrance of the parking lot. Ensure that the devices are facing the right direction and are properly configured.  *prerequisites: Acquire Laser Break Devices* | 4 |

4.6 Mobile Application Task Set

|  |  |
| --- | --- |
| **Task** | **Complexity** |
| **Gather Mobile Application Requirements**  Establish budget, determine hardware, environment, and technology requirements, as well as any other requirements for the mobile application. Determine what platforms we will be targetting (Android, Apple, etc.). Determine the format of the data.  *prerequisites: Define Prototypes* | 4 |
| **Setup Development Environment**  Create the application project. Include dependencies. Put it under version control. Setup virtual device.  *prerequisites: Gather Mobile Application Requirements* | 2 |
| **Develop Application**  Develop the mobile application according to the requirements.  *prerequisites: Setup Development Environment* | 9 |
| **Test Application**  Test the mobile application independent from the rest of the system. Use unit tests as well as integration tests, feeding it expected data from a fake server.  *prerequisites: Develop Application* | 5 |

4.7 Task Prerequisites Chart



**5. Evaluation Criteria**

The success of this project can easily be measured by a student or faculty member’s ability to accurately monitor the parking situation on campus. Right now that ability is pretty much non-existent. By the time we are finished, there will be an interconnected network of I.P. security cameras that will track cars as they enter and exit the parking lots. That info will be displayed to users through an easy to use interface on a mobile application.

There will be three major benchmarks to our success:

* The vehicle recognition software will be accurate to within a two car margin of error.
* .There will be a network of internet-connected camera modules installed at every entrance and exit to all designated parking lots.
* The data gathered by the cameras will be immediately available via an easy to use mobile app.

Once these benchmarks are met, TAMUK students will have that ability to accurately monitor the parking situation on campus.

**6. Schedule**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| September 2016 | | | | | | |
| Sun | Mon | Tues | Wed | Thurs | Fri | Sat |
|  |  |  |  | **1** | **2** | **3** |
| **4** | **5** | **6**  Gather Laser Break Requirements | **7** | **8** | **9** | **10** |
| **11** | **12** | **13**  Establish Problem/Solution  Determine Project Goals | **14** | **15** | **16** | **17** |
| **18** | **19** | **20**  Research Devices | **21** | **22** | **23** | **24** |
| **25** | **26** | **27**  Gather Project Requirements | **28** | **29** | **30** | **1** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| October 2016 | | | | | | |
| Sun | Mon | Tues | Wed | Thurs | Fri | Sat |
|  |  |  |  |  |  | **1** |
| **2** | **3** | **4**  Gather Project Requirements  Determine Camera System Requirements | **5** | **6** | **7** | **8** |
| **9** | **10** | **11** | **12** | **13** | **14** | **15** |
| **16** | **17** | **18**  Obtain Server | **19** | **20** | **21** | **22** |
| **23** | **24** | **25**  Initial Server Setup | **26** | **27** | **28** | **29** |
| **30** | **31** |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| November 2016 | | | | | | |
| Sun | Mon | Tues | Wed | Thurs | Fri | Sat |
|  |  | **1**  Establish Data Connection Wtih Camera | **2** | **3** | **4** | **5** |
| **6** | **7** | **8** | **9** | **10** | **11** | **12** |
| **13** | **14** | **15** | **16** | **17** | **18** | **19** |
| **20** | **21** | **22** | **23** | **24** | **25** | **26** |
| **27** | **28** | **29** | **30** |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| December 2016 | | | | | | |
| Sun | Mon | Tues | Wed | Thurs | Fri | Sat |
|  |  |  |  | **1** | **2** | **3** |
| **4** | **5** | **6**  Build Laser Break System  Deploy and Enable Camera Modules  Initial Mobile Application Setup | **7** | **8** | **9** | **10** |
| **11** | **12** | **13** | **14** | **15** | **16** | **17** |
| **18** | **19** | **20** | **21** | **22** | **23** | **24** |
| **25** | **26** | **27** | **28** | **29** | **30** | **31** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| January 2017 | | | | | | |
| Sun | Mon | Tues | Wed | Thurs | Fri | Sat |
| **1** | **2** | **3**  Mobile Application Prototype | **4** | **5** | **6** | **7** |
| **8** | **9** | **10** | **11** | **12** | **13** | **14** |
| **15** | **16** | **17**  Program Vehicle Recognition Algorithm | **18** | **19** | **20** | **21** |
| **22** | **23** | **24** | **25** | **26** | **27** | **28** |
| **29** | **30** | **31** |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| February 2017 | | | | | | |
| Sun | Mon | Tues | Wed | Thurs | Fri | Sat |
|  |  |  | **1** | **2** | **3** | **4** |
| **5** | **6** | **7**  Finished Application | **8** | **9** | **10** | **11** |
| **12** | **13** | **14**  Establish Data Communication With User Application | **15** | **16** | **17** | **18** |
| **19** | **20** | **21** | **22** | **23** | **24** | **25** |
| **26** | **27** | **28** |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| March 2017 | | | | | | |
| Sun | Mon | Tues | Wed | Thurs | Fri | Sat |
|  |  |  | **1** | **2** | **3** | **4** |
| **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| **12** | **13** | **14**  Full Working System Due | **15** | **16** | **17** | **18** |
| **19** | **20** | **21** | **22** | **23** | **24** | **25** |
| **26** | **27** | **28**  Finished Project Poster | **29** | **30** | **31** |  |

**References**

"RedStorm™ Parking Guidance System | Parking Vehicle Counting System | Parking Car Count System | Signal-Tech." *Signal-Tech*. N.p., n.d. Web. 17 Oct. 2016.

Organization, By. "Operating System Market Share." *Operating System Market Share*. Net Applications.com, n.d. Web. 18 Oct. 2016.

**Glossary**

**Designated parking lot** – A parking lot on the TAMUK campus that has been chosen as a target of the system.

**GPS** - Global Positioning System.

**IP** - Internet protocol.

**Infrared sensor** - A type of sensing equipment that detects subtle differences in the infrared spectrum of light.

**Laser break** **sensor** - A type of sensing equipment that relays data only when a laser beam is interrupted.

**Mobile app** - A computer program designed to run on a mobile device such as a cell phone, tablet, or other internet-enabled device.

**Server** - A computer or computer program that manages access to a centralized resource or service in a network.

**TAMUK** - Texas A&M University-Kingsville.