Swift Park
A Prototype of the Internet of Things

Brien Elwood Washburn, Kohl Riekhof, Jia Jun Yu
University of Utah

***Abstract*— The internet is currently on the doorstep of the next stage in its evolution—the internet of things. The internet of things is a vision of the future in which things in the world can talk to each other. A current problem that can be addressed by implementing the internet of things is parking, particularly at universities. This proposal focuses on development of an internet-of-things protocol, and on a parking-lot-monitoring system that will use the internet-of-things protocol to convey information about each spot in the parking lot. The lot-monitoring system will use ultra-high-frequency radio-frequency-identification tags buried in each parking spot and an on-board computer in each student's car to send information about each parking space to a server. This information will then be displayed on a website.**

I. Introduction

The internet of things is a vision of the future where objects throughout the world will contain and transmit data. As the name implies, common appliances and structures in our environment will be capable of connecting to the internet in a meaningful way. The goal of this vision is to connect the ordinary objects in life to eliminate the mundane, such as shopping for food or watering plants. The internet of things is a vision of productivity.

Parking, especially at universities which sell more passes than spaces, can be time-consuming. Lots are often far apart, or have special rules about which passes can be used during certain times of the day. It is inconvenient to drive each row of the lot only to realize that there aren't any spaces left. Navigating multiple lots in search of a place to park can take up to 10 minutes.

Sometimes the best option is to park in a paid stall when there is no time to wander the lot looking for a spot. Current systems require the user to be carrying change or manually enter information into an app each time they park in a paid space. While paying with an app is a simple process, it suffers from the same problems that traditional meters have—it is necessary to correctly estimate how long you will need to park. Oftentimes these services require a surcharge for processing the payment, making it even more costly to extend the time on a paid stall.

Both of these problems can be mitigated with the Swift Park lot-monitoring system. The Swift Park system involves placing ultra-high-frequency radio-frequency-identification (UHF RFID) tags into each parking space in the lot. Each person that frequents the lot will be assigned a Swift Park hub to connect to the on-board diagnostics-II (OBD-II) port of the car. (Each car past '96 is required to have an OBD-II port that is no further than three feet from the driver and is accessible without tools.) When the car is turned on or off, the hub in the car will engage the tag and relay the information to a server that will use it to modify the Swift Park web page. Time will be tracked, and the user's card will be charged accordingly once the user exits the stall.

Swift Park is a vision of the future. In this future, every car will come straight from the manufacturer with hardware that supports communication through the internet-of-things (IoT) protocol. In reality, Swift Park is a solution that best fits universities. The patrons of a university are consistent, which means the university can require new students to obtain a Swift Park hub. This solution allows for easier parking for the students and faculty of the school, and will increase revenue for paid parking given that users are held accountable for every second that the stall is in use.

The proposed research and implementation of Swift Park involves four main sections: development of passive and active IoT protocols, parking-space sensing, intermediate collection/transmission modules (hubs) and transmission, and the server complex.

The early phase of development will include research into and a definition of the passive and active IoT protocols.

Sensing of the each parking space will be done with a UHF RFID tag that is placed an inch into the concrete of the space. The tag will hold information about the parking space and the URL to send the information.

The active IoT hub will be made of a Raspberry Pi computer that communicates with a UHF RFID reader through the serial peripheral interface (SPI) protocol and with a WiFi module through the universal serial bus (USB) protocol. The hub will activate the reader to scan for a tag when the care is turned on or off. If it receives information, it will transmit the information using WiFi to the address provided through the passive IoT protocol.

An Apache server will receive the information over WiFi and store it in a database. A web server that holds the application will request information from a translator program that will pull information from the database and format it for the website.

Users will be able to access the website from a browser on their phone through the hypertext transfer protocol (HTTP).

The current timeline involves finishing the project by the beginning of the fall semester so there is ample time to thoroughly test the project and attempt to complete a stretch goal.

II. Background

A. Evolution of the Internet

The internet began as a way to connect computers to other computers. It was created in universities as a tool of academia. With the advent of the personal computer, the internet became a tool for people to connect with other people. Social networks see widespread use, as well as sites used for business and education.

As technology has improved over the last decade, the size of the hardware has become drastically smaller and more powerful. The internet has become common in nearly every aspect of our lives, and is now used to keep in constant contact with people through messaging and email.

Power is now so common throughout our world that devices can be placed nearly anywhere. Now that computers can be embedded into just about anything, the internet is going through another stage in its evolution: one in which objects can communicate with one another in the hopes of greater efficiency with less human effort. This next step in the evolution of the internet is the internet of things.

B. Internet of Things Application

A study was conducted about the use of smart objects as the building blocks for the internet of things [1]. This study involved designing smart objects with different design categories such as awareness and interactions. The three types of objects were activity-aware objects, policy-aware objects, and process-aware objects.

Activity-aware objects are capable of recording information about how the tool was used. This data includes metrics such as time, state, and vibration of the object.

Policy-aware objects are activity-aware objects that are capable of evaluating the data stored from activity-aware objects and taking certain action based upon that data.

One example used was a smart barrel that was capable of informing workers in the vicinity of the barrel whether the current state of the chemicals within the barrel posed a risk to workers based on its current condition.

This is similar to the paid-parking functionality in Swift Park that is capable of taking certain actions based upon the interactions between a passive and active IoT component. If a person is in paid parking, the system is able to use information from the passive and active IoT components to charge the user's credit card according to the time that they have been parked.

C. Lot Monitoring

Smart Lot is a senior project completed at the University of Utah in 2010. The Smart Lot system uses a camera to locate empty parking spaces and then relays that information to traffic lights stationed at the end of each row. The challenges this system faces stem primarily from the parking-space detection algorithm. It encounters issues with different lighting, snow, and potentially from the color of cars.

Swarco Traffic Systems is a lot-monitoring system that uses ultrasonic technology to determine where cars are parked [2]. The ultrasonic sensors are capable of measuring the distance of an object to the ground. If the distance traveled by the wave is determined to be reduced by the presence of car, the sensor sends an "Occupied" status message to a higher-level area controller. These controllers then communicate with hardware that displays information about parking on signage inside of the parking lot. Streetline uses a mesh network to relay information about free parking spaces and volume and speed of passing traffic [3]. Each node in the network has a magnetic sensor that detects cars parked above it. The sensor is capable of determining if a car is above it by searching for disturbances in the earth's magnetic field. This information is then transmitted wirelessly through the mesh network. A mesh network allows for each mesh node to collect and transfer information through other nodes and eventually to servers that can display free parking spots through street signage or smartphone maps.

The Swift Park system circumvents the detection problems encountered by Smart Lot by monitoring each spot with its own sensor. This will allow accurate sensing of individual parking spaces despite environmental factors such as weather or the time of day.

The primary differences seen between Swarco Traffic Systems and Streetline and Swift Park is the cost. The sensors for these two systems are more complex due to their sensing methodologies and the ability to communicate wirelessly. The sensors in the Swift Park system can be made at a fraction of the cost, with the cost of the hub being distributed to students in the form of a parking pass. Swift Park also utilizes a dedicated website to display lot information instead of hardware in the lot or existing maps for smartphones.

III. Proposed Work

As computer engineers, it is important to understand current technology while having an eye on the future. Computer engineering is an innovative field and competitive field, and it is a useful skill to be able see the trends in technology.

The goal of this project is to predict a possible future of the internet, and to create a hardware implementation that can facilitate that future internet. The prediction for this proposal comes in the form of the internet of things.

A small portion of the project will revolve around developing a simple protocol that can be used for the internet of things, or utilizing existing protocols [4]. This is discussed below in Research.

The main body of the project is the implementation that supports the internet of things, which is the lot-monitoring system, Swift Park. The work required to implement Swift Park is described in Implementation.

A. Team Responsibilities

Jia Jun Yu is responsible for all hardware. He will research parts for all aspects of the project, and will be in charge of procurement as well.

Brien Washburn is responsible official documentation. This includes both rough and final drafts of the project proposal. Brien is also responsible for managing the repository.

Kohl Riekhof is responsible for documenting team progress, coordinating meetings, and setting up and managing the team web page.

B. Project Responsibilities

The diagram shown in Fig. 1 provides an overview of the communication seen in the lot-monitoring system. In the left side of the diagram, the active IoT hub in the car is communicating with the UHF RFID tag via passive IoT and then transmitting that information to the Apache lot server in the Cade lab via WiFi. The web server is then able to query the lot-server database through the translator. The information it receives from the translator can be used to generate the web application. Users can request the web page that exists on the Cade machine through an HTTP request.

Jia Jun will handle all communication on the link layer of the protocol stack. This includes passive IoT communication between the active IoT hub in the car and the passive IoT tag in the parking spot.

Brien is heading up development of the active IoT hub that will connect to the OBD-II port of the car. This requires using the Raspberry Pi to interface with the UHF RFID reader through SPI and the WiFi module through USB. It also requires development of infrastructure to power the Raspberry Pi through the OBD-II port (Jia Jun will work on powering the board with Brien).

Kohl is responsible for developing the software suite pictured in Fig. 1. This includes the web application, the translator, and the server that will communicate with the active IoT hub. (Brien will assist the with lot server and the translator.)

C. Research

The research portion of the project is centered on defining two protocols to communicate with active and passive IoT hardware. The passive IoT protocol will be used to communicate with hardware that cannot process information. When two active IoT items such as a server and Swift Park hub are communicating, the active IoT protocol will be used.

When innovating it is unnecessary to reinvent the wheel, so we will be making use of the current protocol stack utilized in networking. The protocol stack consists of five layers. From bottom to top, these layers are: physical (or hardware), link, network, transport, and application layers. The protocol stack can be seen in Fig. 2, with common protocols to the right of the stack.

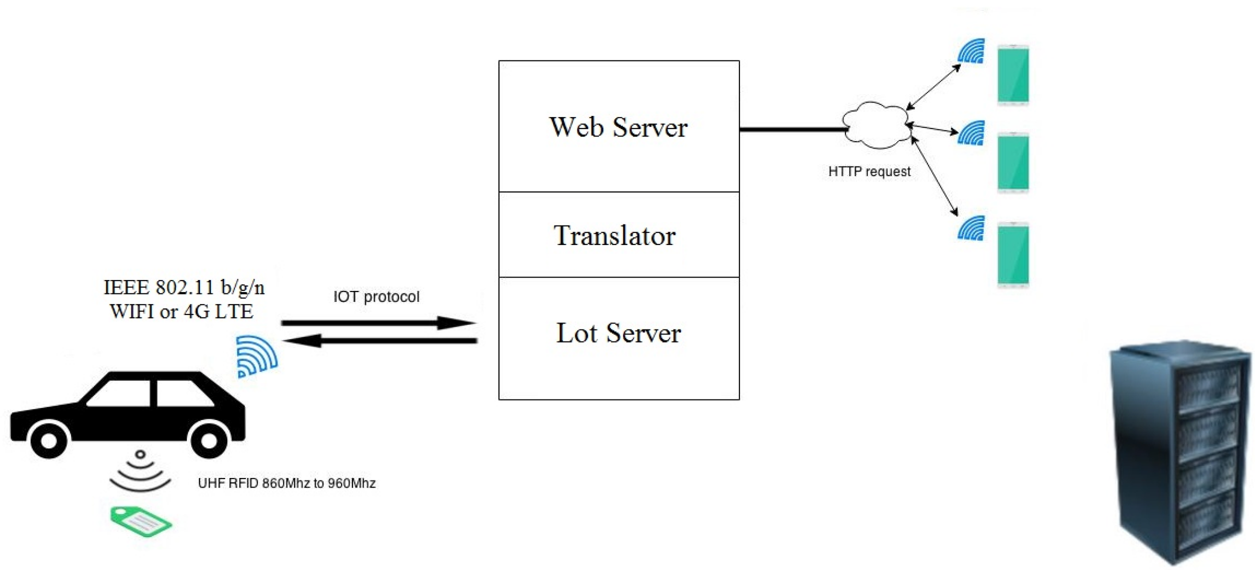

Fig. 1. Overview of communication from a parking spot to the user.

(The User/Application layer is not a part of the five-layer protocol stack—it is used for the express purpose of showing a typical network from top to bottom.)

The physical layer describes the medium over which the information travels. In this case, information will travel on electromagnetic waves.

The link layer details the protocol that controls the physical layer. In the Swift Park project, the link layer will support two protocols: passive IoT using the UHF RFID reader, and WiFi (IEEE 802.11), which will communicate information to the Apache lot server.

The network and transport will mantain their nearly shared functionality of making certain that messages are accurately passed from the user, down through the protocol stack, and back up to web page.

The application layer is traditionally used for the HTTP protocol for requesting information about web pages on the internet. In this project, the application layer will be used to facilitate communication over the active IoT protocol, which will be using an HTTP framework.

When communicating with UHF RFID tags, it is necessary to minimize the complexity of the protocol. This means that the UHF RFID tags need to carry a small amount of information, namely only a uniform resource locator (URL) and a value. The intent is to send the value to the URL provided, at which point the server can store the information or return instructions back to the hub.

The active IoT protocol will be used between active IoT components which are able to process information. The active protocol will be capable of communicating a variety of actions which will be specified in the handshake between two active components. Each active IoT component will then be able to provide requests to the other IoT component, as well as the parameters required to enact the request.

The first avenue of research into the active IoT protocol involves using a protocol for IoT that has already been defined [4]. The secondary option is a novel implementation, and the third is to use HTTP as a framework. This would mean that the protocol would simply specify the formatting of text that is transferred using HTTP. The third option is the most likely given that it is the simplest, but it will be educational to first try and define a clean, simple protocol or find an existing one that is suitable for the project.

D. Implementation

Once a suitable IoT protocol has been found or developed, the protocol will be utilized in the internet-of-things implementation, Swift Park. The goal of Swift Park is to make it simple for users to find optimal parking, avoid tickets through automatic pay, and potentially find their car if it is stolen and parked in a Swift Park-capable lot.

The sensing segment of the project involves placing a UHF RFID tag in each parking stall. Each tag will contain a URL and a value. (The value is a unique number that corresponds to the parking stall in which it is placed.) The tag will communicate with the reader using the passive IoT protocol. The tag can be read by a UHF RFID reader that will generate a burst of radio waves at the frequency of the tag. This will power the tag, which will emit the URL and value stored on the tag.

The hardware that initiates reading of the tag is the hub. The hub is made up of a Raspberry Pi, a UHF RFID reader, and a WiFi module that connects to the Pi via USB. The RFID reader will scan for tags each time the car is turned on or off. This will allow the car to identify when it pulls into or out of a parking spot.

After receiving information from a tag, the reader will transmit the information to the Pi through SPI. The Pi will create a TCP socket and will send the value from the UHF RFID tag to the URL specified through the WiFi module. The IoT Active protocol, which exists on an HTTP framework, will be used to format the data. This data will then descend through the network stack, making use of TCP on the Transport layer, IP on the Network layer, and WiFi on the Link layer.

The server complex that will receive the information consists of three main components: a server that will accept the parking-space information over WiFi and store it in a database; a translator program that can transform the information stored in the database into an optimal form for the web application; the web server that will receive the output of the translator and use it to modify the Swift Park web application. The server and the translator will created in Apache, and will reside on a machine in the Cade laboratory at the University of Utah. The web application will consist of a server programmed in PHP, a database created using MySQL, and a web page constructed using Mobile jQuery (this will make it simpler to use on smartphones), cascading style sheets (CSS), and hypertext markup language (HTML).

Users can request information about the parking lot through the HTTP protocol. The URL of the server hosting the Swift Park web application can be entered into a browser to navigate to the web page.

This system will allow a user to check any monitored lot at any time by simply entering the appropriate URL in a browser and navigating to the relevant lot.

E. Stretch Goals

1) Custom Printed Circuit Boards: A significant stretch goal that we have is to create custom printed circuit boards (PCBs) for the hardware in the car. This would involve a single board with a processor that

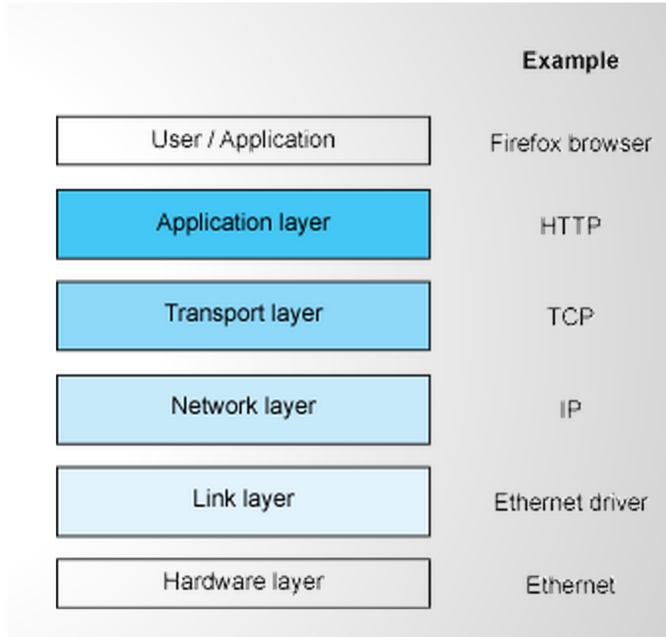

Fig. 2. The common five-layer protocol stack used in networking [5].

would play the role of the Raspberry Pi. The processor would interface with the UHF RFID reader, as well as the WiFi module.

Making custom PCBs would be a significant investment, and would require extensive custom software for the processor to be able to interface simply with the WiFi module. (The Raspberry Pi already has these capabilities built into it with the Raspbian operating system.)

2) Smartphone App: The current plan involves making a website that is smartphone-friendly. It would be great there was time to make either an iOS or Android app that could be downloaded once and used for all facets of the Swift Park experience.

3) Developer Tools: It is important to treat any project as a professional. While it is unlikely that Swift Park will be purchased and implemented for wide-scale use, it would be useful to approach the project from this point of view.

Creating development tools for easy implementation of new parking lots would be a great step to making the project commercially viable. It would be quite difficult for new users (or even those familiar with the product) to design and add new parking lots. A generic set of tools that can accurately represent the layout of the lot and important information about parking spaces in the parking lot would make the setup for new lots much simpler.

F. Design and Implementation Risks

1) Protocol Definition: Defining protocols can take a substantial amount of time. It is almost certainly beyond the scope of this project to define a full protocol, so it will be important to not spend too much time researching and defining a custom protocol. The best option is to use either an existing protocol specifically for the internet of things, or define a very basic protocol that specifies the format of text which can be transmitted using HTTP.

2) Reader Interference: UHF RFID tags have a great deal of range, and, unobstructed, the tags we are using would have no problem reading at the one to two meter range necessary for Swift Park to work. However, RFID technology in general can be quite sensitive to environmental factors, and UHF systems are particularly prone to issues with reflection or re-radiation of power signals [6]. It may be necessary to place an antenna on the underside of the car to allow for clear, reliable communication between the reader in the hub and the tags in the parking stalls.

G. Product Risks

1) Phantom Cars: There are currently a few serious risks posed by installation of the Swift Park lot-monitoring system. The most significant comes from the fact that a car without the hub is undetectable by the system. This means that it will not appear on the website, even if the car parks in a spot with a UHF RFID tag. This makes the Swift Park system extremely error prone when dealing with "phantom cars".

The remedy to the problem of phantom cars comes from consistent policing. Lots are currently monitored by staff to see if people are parking without passes. This effort would have to continue to ticket people without the appropriate hardware.

2) Loss of Internet Access: WiFi is crucial to Swift Park. It is impossible to transmit data about the lot if the hub in the car cannot transmit data over WiFi. This risk is significant if there is no infrastructure on each hub that keeps track of data to make sure that any information collected is guaranteed to be transmitted. This is especially important to paid parking, and to maintain the convenience of the Swift Park system. It may be possible to use smartphones to transmit information, which would be a great boon.

3) Broken RFID Tags: Broken UHF RFID tags can pose a problem in that it would be difficult to know if a tag is not working. It would be necessary for staff to identify a car that is parked on a stall that does not appear to be occupied but is known to contain a hub.

In the case that a broken tag is identified, the caulk covering the tag can be removed and the tag replaced and recovered without too much effort.

IV. Testing and Integration

Swift Park has many complex pieces that will require extensive testing and slow integration. This section will talk about each aspect of the project in regards to testing The testing will then be tied together in a section about Integration.

A. Testing Hardware and Software

1) Car Hardware and Software: Testing of the car hardware will be conducted on the Raspberry Pi Model B+. The Raspberry Pi is a computer the size of a credit card that is capable of connecting to a monitor and keyboard. The Model B+ has the Raspbian operating system on it. (Raspbian is a Linux distribution, called Debian, that is optimized for the Raspberry Pi.) Python, which the Raspbian operating system is capable of compiling and running, will be the language used for testing.

A UHF RFID reader will be connected to the Raspberry Pi through SPI. This reader will enable communication between the car and the spot by engaging with the UHF RFID tag embedded in the parking space. (While testing, tags will not be buried.) Tags will be programmed and read using the RFID reader.

Messages will be transmitted from the car to the server through a WiFi module that engages with the Raspberry Pi via USB2.0/1.1. To test the WiFi module, messages will be generated and transmitted to the Apache server in the Cade lab. The messages that will be transmitted by the WiFi module will be in the IoT-Active-protocol format, and can be compared to the generated messages to see if the format was maintained through WiFi communication.

The system will be powered through the OBD-II diagnostic port in the car. This port is the general-purpose port used for all diagnostics of the car's on-board computers, and is capable of supplying power to the device.

2) Stall Monitoring: Each stall will be equipped with a UHF RFID tag that will hold information about the stall. Specifically, each RFID tag will contain the URL of the website as well as a value that identifies the specific parking spot.

To test the range of the UHF RFID tags, a tag will be placed on the ground at distances ranging from .5m to 2m away and then read using the UHF RFID reader. If the reading distance is significant (exceeding 2m), multiple tags will be lined up at a width of one parking stall and then the tags will be read again. If multiple tags can be read by a single reader, it will be necessary to handle multiple tags.

The two ways to handle the reading of multiple tags are: attempt to tune the distance of the RFID reader; transmit all tags read and come up with software for the website that will correctly pick out the correct tag. (If the reading distance is consistent, it will always be possible to identify the correct tag given a list of tags.)

3) Servers, Translation Software, and Web Application: A server, coded in Python, will be created to receive data. This server will be receiving information from the car hub, which will be transmitting over WiFi. The information will be a parking spot and the unique identifier of the car that transmitted the data.

The server will reside on a machine in the Cade laboratory. To test this server, simple messages will be transmitted over WiFi and then stored in the MySQL database. This will be done using a Raspberry Pi and a WiFi module that connects to the Pi through USB.

The database can then be queried to see if the information sent was received and stored appropriately.

The translator module will be coded in Python, and can be tested by stimulating the program with prompts similar to those that will be generated by the web page. The program will then access the database, retrieve any new information, and format it so the Swift Park website can easily integrate it. The translator can be tested by manually setting database entries, pulling from the database, and then checking the data and expected format to see if they are congruent with the design of the software

The web page will be generated using PHP and CSS and can be tested by requested information over HTTP. This information can be displayed on a computer or smartphone and checked for discrepancies.

B. Integration

Integration will begin with the hub. UHF RFID tags will be programmed and read by the UHF RFID reader. The reader will be integrated with the Raspberry Pi, along with the WiFi module. The information stored on the UHF RFID tag will be read and transmitted through WiFi to the Apache server and then checked against the data stored on the tag.

Once this process is completed, the data sent from the hub and received by the Apache server will be stored in a database. This database will be queried, and once again the information will be checked against the data stored on the tags.

The translator will be integrated next by generating inputs to the program and allowing it to retrieve and format data from the database.

Finally, the requests can be generated by the web page and then transmitted over HTTP when requested through smartphones.

This modular approach will make it easier to identify and isolate issues at each stage of development.

V. Schedule

The current wrap-up date for the project is scheduled to be the end of August. This will leave plenty of time for extended testing or in case more time is needed to complete the project. The Gantt chart seen in Fig. 3 provides an overview of the major milestones of the project.

The first major milestone includes the reading and writing of UHF RFID tags. This milestone is set to be completed by the end of May.

The protocol-definition milestone is being pursued concurrently to reading the tags, and will finished near the start of June.

The next step is to get the RFID reader to communicate through SPI with the Raspberry Pi. The Raspberry Pi will then process the information from the passive-IoT protocol, and generate a message to transmit to the WiFi module via USB. The projected finishing date for this milestone is the middle of June.

This time will also be used to develop the database that the Apache server will be storing information in. The database will be completed in the last quarter of June.

The next milestone is comprised of reliable communication between the WiFi module and the parking-lot server. This will entail development in Python on the Raspberry Pi module that will be placed in the car, as well as the server for the lot. This milestone, as well as the translator between the website and the database, will be completed by the middle of July.

Development of the website and the content for the page will start once WiFi is handled and the translator has been finished.

As each milestone displayed on the Gantt chart concludes, extensive testing for that milestone will ensue.

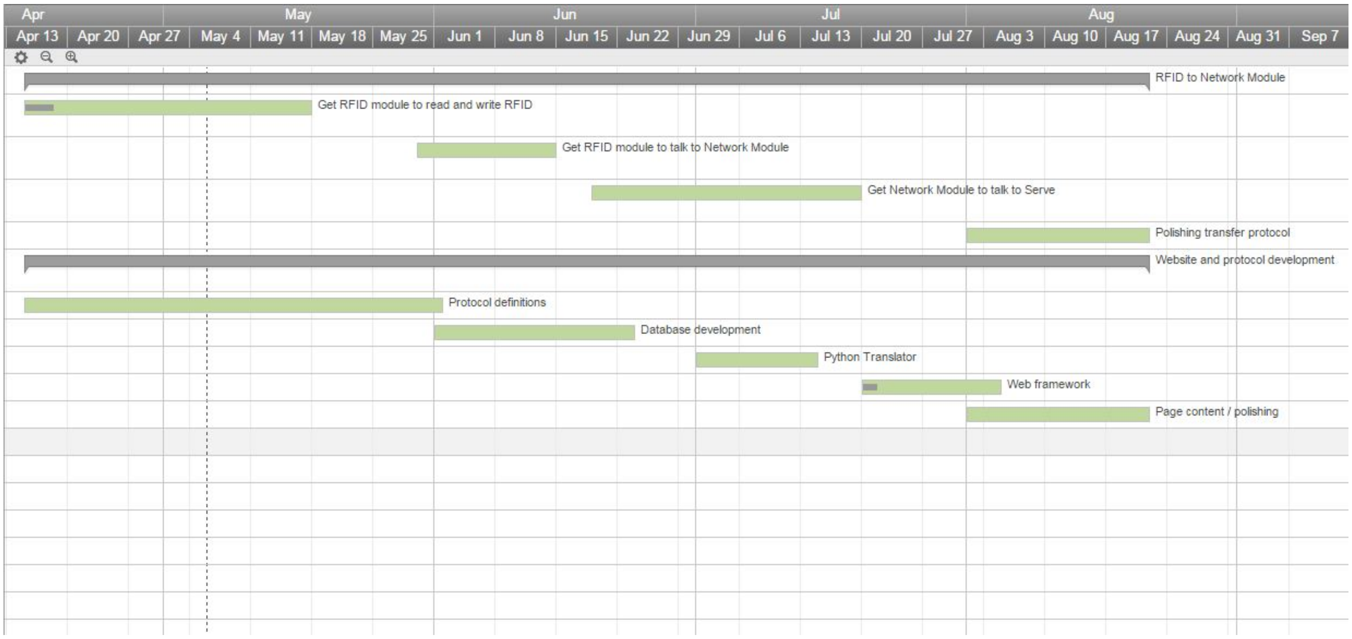

Fig. 3. Timeline of the project in a Gantt chart.

A. *Team Meetings*

Team planning has taken place weekly throughout the semester to be certain that project development is an ongoing process. Meetings have typically been scheduled for at least one hour, and occur on either Mondays or Wednesdays. Recently, meetings have last approximately two hours in length to flesh out of the details of the project.

Development during the summer will be accelerated in an attempt to finish the bulk of the project before the start of Fall semester. Meetings are currently scheduled to occur twice a week, on Mondays and Thursdays. Team meetings will begin at 10:00am, and will adjourn no earlier than 11:00am.

VI. Required Resources

There are many resources that are available through the University of Utah or have been previously acquired for other projects. However, it will be necessary to procure other supplies and software.

This section will list the resources that have been previously purchased or are attainable through the University of Utah. It will then itemize all items that will require purchase, and will conclude with a bill of materials.

A. *Freely Available Resources*

Jia Jun will need access to a Raspberry Pi and the WiFi module used for wireless communication to the server in the Cade lab. Brien will also make use of this hardware when integrating the Raspberry Pi and the wireless module into a single hub.

The server suite can be implemented by Kohl and Brien on a machine in the Cade lab. All of the software, including MySQL, Python, PHP, Mobile jQuery, and CSS are accessible through the cade machines.

Kohl will also need access to Adobe Dreamweaver to design the web page, a smartphone to test the web page, and a computer to host the server suite. The server suite can be hosted on a Cade machine and a smartphone has already been acquired, while Adobe Dreamweaver can be acquired through the University.

B. *Items to Procure*

Jia Jun will require UHF RFID tags and readers to handle communication between the active and passive modules in the Swift Park system. These will need to be purchased. Brien will need a UHF RFID reader when making the hub.

C. *Bill of Materials*

UHF RFID tags will have to be purchased for testing and implementation. The tags that will be used are the Alien H3 UHF 860Mhz to 960Mhz waterproof tag. These UHF RFID tags can be purchased from Alibaba. The maximum cost is $.2 per unit, with a minimum order of 100 pieces. This would necessitate an order costing approximately $20.

The UHF RFID reader used to read tags from the hub is the cc1101-Q1 Low-Power Sub-1Ghz RF Transceiver. The cost of this reader/writer module is $6.00. It will not be necessary to acquire more than two for testing and implementation.

It is likely that miscellaneous parts will be required to interface the devices for the hub during testing, which will likely raise the cost by $10-$15.

This will put the total budget for all testing and implementation (without consideration any stretch goals) up to approximately $40.

VII. Summary

The world is becoming increasingly connected, and The internet of things is the next logical evolution of the internet. Searching for convenient parking is an unnecessary activity no one wants to engage in, and an intelligent lot-monitoring system can use the principles of The internet of things to decrease the time and effort required to find and pay for parking. Other lot-monitoring systems oftentimes require the users to drive into the lot before it is possible to determine if there is any available parking. Swift Park removes this inconvenience through a free website that allows users to see if the lot has available parking long before they are on location. It also provides users a way to pay for parking, and, with widespread implementation use, locate stolen cars.

Swift Park is an interesting and useful senior project for the creators because it requires an understanding of modern networking infrastructure, the design and implementation of custom protocols and software, and the ability to use existing hardware and software to make a new product. Implementing RFID technology fortifies electrical-engineering fundamentals, application development and networking expand computer science expertise, and the interface between hardware and software unify the fields of electrical engineering and computer science into a perfect project for computer engineers.

References

[1] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy, "Smart objects as building blocks for the internet of things," *Internet Computing, IEEE*, vol. 14, no. 1, pp. 44–51, Jan. 2010. [Online]. Available: http://dx.doi.org/10.1109/mic.2009.143

[2] "Individual parking space monitoring - sensors." [Online]. Available: http://www.swarco.com/sts-en/Products-Services/Parking/Detection/Individual-Parking-Space-Monitoring-Sensors

[3] K. Grifantini, "Find a parking space online," Jul. 2008. [Online]. Available: http://www.technologyreview.com/news/410505/find-a-parking-space-online/

[4] S. Schneiger. Untitled. [Online]. Available: http://electronicdesign.com/embedded/understanding-protocols-behind-internet-things

[5] "Untitled." [Online]. Available: http://www.cyberciti.biz/tips/wp-content/uploads/2007/08/linux-internet-model-network-stack.gif

[6] RFID & interference. [Online]. Available: http://www.corerfid.com/technology/TechnologyIssues/IssuesInterference.aspx