Smart Parking Meter Solution for Urban Curb-Side Parking

ECE4011 Senior Design Project

Team Name: Meterrific
Faculty Advisor: Dr. Xiaoli Ma

Shiv Chopra, Computer Engineering, schopra37@gatech.edu
Blake Heard, Computer Engineering, aheard8@gatech.edu
Madison Hester, Computer Engineering, mhester32@gatech.edu
Michael Knudson, Computer Engineering, michaelknudson@gatech.edu
Raj Patel, Computer Engineering, rpatelpj@gatech.edu
Andrew Trimper, Computer Engineering, andrew.trimper@gatech.edu

Submitted

November 18, 2019

Table of Contents

Executive Summary

Meterrific attempts to modernize the metered street parking experience. The product design will address improvements in efficiency and usability over current standard metered street parking solutions. The overall goal is to provide the user with an end-to-end parking experience that streamlines navigation, spot selection and payment in a product that is easy for drivers to use and for municipalities to incorporate into existing systems. One important design problem is the separation of technical responsibilities between the app and the parking meter such as (1) declaring a parking spot as full, empty or reserved or (2) accepting payment from the user. Technical challenges include design and fabrication of the physical parking meter and design and implementation of the accompanying mobile app. The parking meter will be designed with industry-standard CAD tools and fabricated through a mixture of machining and 3D printing. The UI/UX component of the app will be designed on Adobe XD, and produced with software engineering tools including modern version control (Git). The product will use both an app and an embedded systems parking meter to allow a user to navigate to a desired parking spot and start a metered parking session. The app will use physical sensors both on the mobile device and the meter to smartly suggest parking spaces to navigate to given a destination, when to start a session and when to stop a session. The estimated cost per unit is $156.33, though this does not include labor costs for software maintenance. The proof-of-concept test for the completed prototype will include a user's attempt to navigate to a parking spot, change his selection mid-navigation and start and end a metered session. Future work may include additional features or incorporating the product into an existing metered parking solution.

Nomenclature

Symbols/Variables/Parameters

Temperature	℃	degrees Celsius
Length	m	meter
	in	inch
Current	A	ampere
	mA	milliampere
Voltage	V	volt
	mV	millivolt
Electric Charge (Current x Time)	mAh	milliampere hour

Acronyms

ABS	Acrylonitrile Butadiene Styrene
API	Application Programming Interface
BLE	Bluetooth Low Energy
ECE	Electrical and Computer Engineering
GPS	Global Positioning System
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronics Engineers
JSON	JavaScript Object Notation
LCD	Liquid Crystal Display
LIDAR	Light Detection and Ranging
PCB	Printed Circuit Board
PERT	Program Evaluation Review Technique
QFD	Quality Function Deployment
SDK	Software Development Kit
UI/UX	User Interface/User Experience

1. Introduction

Meteriffic will design a smart parking meter to help address issues with urban street parking by allowing users to find available street parking close to their desired location in real time through an associated app. The team is requesting $164.20 in funding to develop the prototype.

The objective of the smart parking meter is to notify users of available street parking near their location to help eliminate time spent searching for street parking in busy city streets. The team will design and prototype a new parking meter with an associated app to accomplish this goal. The meter itself will perform all the normal functions of a traditional parking meter like notifying parking attendants of unpaid parking and accepting payment for parking. The unique features of the meter will be (1) the use of sensors to identify whether a car is parked in the spot or not in real time and notify app users with this information and (2) navigation to open parking spots near desired destination and rerouting as spots open and close.

The motivation behind designing the smart parking meter system is to limit time spent searching for parking and provide users a stress-free city parking experience. By the end of the project, the team aspires to develop a complete parking system that encompasses finding the closest spot to a given destination and then navigating to it. Once at the spot, the user will be able to easily pay and experience their destination without the associated parking stress. The current intended audience for the app is smart-phone users that need to navigate around a large city, but the prototype could expand to work within smaller districts. The smart parking meter will increase parking enforcement revenue, as spot detection can increase city revenue by an average of 25-50% by reporting parking violations [1]. Given the product will be comparable in price to alternatives, city governments will be incentivized to choose this product because of its superior features and design [2-3].

The smart parking meter will utilize sensors and Wi-Fi to send real-time data to the database. The database will then be utilized by an associated smart-phone app so that users can get updated parking information. A major challenge with a smart parking meter is how to make it small, energy efficient, and low cost. The team needs to design a system that can rely on solar power and thus needs to limit the power consumption across the meter. The firmware, features, and parts must be adapted to keep power consumption low. The team also has to consider the fact that they are creating a prototype, not producing on a large scale. A sacrifice associated with making a prototype is that the team will be using a Wi-Fi module instead of a 2G cellular chip. For operation in the field, the idea would be to incorporate cellular communication. Since there will not much data sent from the parking meter to the database, 2G communication will provide sufficient bandwidth for data transmission and would have widespread availability in comparison to Wi-Fi. However, for the purposes of prototyping, Wi-Fi will provide a more cost-effective solution.

The main feature of the app is to navigate users to the closest open parking spot near their destination. The challenge of integrating the database with the API will be creating the queries to gather the appropriate data from the sensors, process it, and be able to efficiently and effectively send that back to users. The app will also need to be consistent and reliable with the information it provides so that users will continue utilizing it.

In the following sections, the project proposal will address the project description and goals, technical specifications, design approach and details, project demonstration, team schedule, current project status, and team leadership roles and responsibilities.

2. Project Description, Customer Requirements, and Goals

The overall goal of the Meterrific Smart Parking Meter is to establish an efficient, easy way to find on-street parking. The project will consist of two parts: the actual parking meter and an Android app that communicates with the parking meter. The parking meter itself is made up of a housing unit and a processing unit. The housing unit encases the processing unit and protects it from the weather, while the processing unit includes a custom PCB board with an LCD screen, a LIDAR sensor, and a microprocessor with Wi-Fi capability. The processing unit will utilize the LIDAR sensor and Wi-Fi capability to send status updates to a cloud database. The Android app will navigate a user to the closest parking spot to his or her destination, and it will communicate with the actual parking meter through a cloud database to confirm occupied parking spots. Customer needs and project goals include:

Overall

- Parking meter system that utilizes hardware to track overall status (spot occupation, parking time allotted) and relay this status to a database while utilizing software to navigate a user to a parking spot
- Targeted user: car commuters in heavily populated cities
- Target price of product: $300

Parking Meter

- Protects processing unit from the elements and people
- Keeps track of overall status and sends it to a cloud database

Android App

- Efficiently navigates the user to the closest parking spot to the specified destination

- Receives status information from a cloud database and re-routes the user if necessary in real time
- User able to pay directly through the app

Customer needs and constraints will need to be further considered for the duration of the project. Appendix A displays a QFD Chart based on customer needs and engineering requirements. There are also both hardware and software constraints. For example, the processing unit will be adding additional components not in a normal parking meter, such as the microprocessor with Wi-Fi. These extra components will need to be powered at all times, so it will be vital that there be a sufficient power source. Additionally, the Android app and the physical parking meter will need to communicate through a database, so naturally there will be a slight delay in updating information. Another constraint will be design materials. The housing unit will need to be made of a weather-resistant material while still maintaining breathability to avoid the overheating of any electrical components within. Design materials will also need to fit budgetary constraints, so the size of the parking meter and the materials used will need to be monitored to stay within the budget. Finally, a key constraint will be design and city codes and standards, as the design will need to strictly follow these standards to be deemed safe and legal.

3. Technical Specifications

Table 3.1 Microcontroller Specifications for Optimal Energy Consumption

Feature	Specification
Operating Temperature	- 30℃ to + 45℃
Maximum Current Draw	5 mA
Operating Voltage	< 5 V

Table 3.2 Proximity Sensor Specifications For Maximum Accuracy

Feature	Specification
Sensing Range	> 1.3m
Operating Temperature	- 30℃ to + 45℃
Maximum Current Draw	15 mA
Operating Voltage	< 5 V

Table 3.3 Wi-Fi Specifications For Ideal Performance

Feature	Specification
Operating Temperature	- 30℃ to + 45℃
Maximum Current Draw	200 mA
Operating Voltage	< 5 V
Boot-Up and Transmit Time	< 2 ms

Table 3.4 LCD Specifications For Usability and Operation

Feature	Specification
Operating Temperature	- 30℃ to + 45℃
Maximum Current Draw	20 mA
Operating Voltage	< 5 V

Table 3.5 Battery Specifications For Ideal Lifecycle

Feature	Specification
Maximum Power Storage	1700 mAh for 300 to 500 cycles

Table 3.6 Solar Cell Specifications

Feature	Specification
Current Generation	< 1 A
Voltage Generation	< 5 V

Table 3.7 Unit Housing Specifications for Safety and Convenience

Feature	Specification
Minimum Ingress Protection Rating	IP35
Dimensions	5” x 3” x 2”

4. Design Approach and Details

4.1. Design Concept Ideation, Constraints, Alternatives, and Tradeoffs

The smart parking meter solution is composed of a physical add-on to current single-space parking meter models and a connected app. The solution will allow the user to pay from their phone prorated over real-time spent in the parking spot. The solution will navigate the user to the nearest parking spot, re-directing them as spots close and nearer ones open based on traffic. The physical add-on will have the task of wirelessly communicating with the app about the availability of the parking spot.

In order to meet the desired functionality, the physical add-on will need to provide current single-space parking meter models with additional computational capability. There are multiple options to ascertain the availability of a parking spot: Bluetooth Low Energy (BLE), range-finding sensor, or GPS. BLE would allow the smart parking meter to sense when a user is near by recognizing their phone. However, the trade-off with BLE would be a greater strain on power supply than a range-finding sensor. An alternative would be to have users' phones track the fixed GPS location of the smart parking meters and, thus, not expend more energy powering peripherals. There are also a couple methods to connect to the internet and communicate necessary information to the app: Wi-Fi or cellular. The tradeoff with cellular is the increase in cost of hardware in comparison to Wi-Fi. The physical add-on will need a microcontroller that can interface with, potentially, a proximity sensor, LCD, and wireless communication module. The LCD is needed to display information to users not using the app.

The physical add-on needs to operate at all times to realistically be implemented in cities. For operation, the physical add-on needs a consistent power-supply and sturdy casing for protection from weather and vandalism. A realistic design constraint is the power the system will consume while being in a low-power environment. Since the average parking meter is being upgraded to an IoT device, new components for IoT functionality will draw more power in addition to the power already drawn from necessary components in the average parking meter. The problem arises when there are more components in the design than are able to be reasonably powered or when all of the components are drawing power at all times. A major trade-off will be between complexity and expended power. When more computational aspects are included, the strain on the battery will be higher. In the case of how to power the meter, the supply would come from solar power, rechargeable batteries, and backup replaceable batteries. While the sun would power the parking meter during the day, it could also recharge batteries for operation during nighttime. In case of emergency power need, the backup replaceable batteries would provide power while communicating to the database the emergency.

Other realistic design constraints are the size and cost of the smart parking meter compared to the average parking meter. If the smart parking meter is bulkier and cannot fit within a shape or space similar to average parking meters, then it will be more difficult to transition between the two. If the smart parking meter is significantly more costly than average parking meters and revenue generated for example, then the public sector will have a harder time investing in this technology. Another large trade-off will be between complexity and cost. With more hardware aspects included for necessary functionality, the budget becomes more limited.

For sustainable communication between the wireless communication module in each parking meter and each user app, there needs to be an online database where the wireless communication module stores information to be read by the app. An optimal database would be Google Firebase since it is ideal, in comparison to MongoDB, for real-time systems and would utilize the latest stored data when offline. An alternative database would be MongoDB, but it is more useful for querying documents. A realistic design constraint is the amount of time communication would take between the wireless communication module in the smart parking meter, database, and app. If the average communication time is longer than a couple of seconds, then, due to its real-time nature, navigation and re-routing would become unstable and too inaccurate to be convenient to the user.

In order to accurately navigate users to open parking spots, the app must be integrated with existing map APIs. There are a couple of choices for map APIs: Google Maps, Mapbox, and OpenLayers. OpenLayers is a free service whereas Google Maps and MapBox charge based on number of transactions with the service. However, Google Maps and MapBox provide navigation features. Additionally, Google Maps includes traffic tracking as a feature. While navigating the user, the app will update directions based on information it accesses in the online database. In summary, the primary hardware-software interaction in this design will be the communication between the smart parking meter and the app through a database.

4.2. Preliminary Concept Selection and Justification

Table 4.2.1 Decision Making Matrix for Optimal Feature Selection

	Rating Metrics					
Concepts	Cost	Software Development	Hardware Development	Feasibility	Improvement To System	Total Rating
Physical Parking Meter	-4	0	-5	5	10	6
Visualize Open Parking Spaces	-1	-4	-1	5	8	7
Physical Arm to Reserve Spot	-4	-2	-7	2	5	-6
Navigation to Open Spot	-1	-4	0	5	5	5
Sensor to Check if Spot is Open	-1	-2	-2	5	6	6
Bluetooth Communication Between Meters	-2	-4	-3	3	5	-1
Wi-Fi Communication for each meter	-3	-3	-2	5	6	3
Solar Power to Power System	-4	0	-5	5	5	1
Power Grid to Power System	-10	0	-10	1	6	-13
Replaceable Batteries to Power System	-3	0	-4	4	5	2
Pay Through App	-1	-3	0	4	5	5
Pay Through Meter	-4	-1	-5	2	5	-3

There were numerous promising features that we wanted to include in our Smart Parking Meter project. To narrow down these concepts to only those that are beneficial to the project, the decision-making matrix in Table 4.2.1 was utilized. The rows represent the promising concepts to be rated. The columns of the table represent the metrics to rate those concepts. The values in the cells represent the rating for each metric. Finally, the concepts with the highest total ratings were selected. This can be seen in the table with those cells that are highlighted.

The highlighted concepts in table 4.2.1 were selected to be designed further. These concepts have varying degrees of feasibility. The physical parking meter is a requirement for the design as it will house all the electronics to provide the functionality for the other concepts.

A sensor to check if a car is in a parking spot is a feasible concept, as it requires only a simple distance sensor in the housing of the parking meter. Then only a small amount of coding has to be done to read it and send the data to a database.

Visualization of open parking spaces is a feasible concept, assuming the previous concept is complete. In terms of software, the data stored in the database has to be made accessible to the app. A block must be placed on the location of each open meter on a map that is displayed on the app. Navigating to an open spot is similar as also very feasible. The only addition to this concept is to have integration with a navigation SDK such as google maps.

The Wi-Fi communication for each meter is required to have the meter communicate with the database so that the app can perform its functions. The feasibility of Wi-Fi communication should not be concerning because there are countless examples of solutions adding Wi-Fi communication capability to microprocessors, if the microprocessor doesn’t already have that capability.

The next concepts are to power the meter with solar power and with a replaceable battery. These concepts are feasible because this is how many parking meters on the market are powered. They are also necessary to the meter because they bring reliable power to the parking meter.

The final concept is to pay for parking through the app. This fulfills the requirement of monetizing the parking meter and does this without adding any hardware components. The software will have to include the ability to receive payments from a credit card, this can be done through a pre-existing platform such as PayPal.

Referencing the PERT Chart from Section VI, it can be observed that the critical path goes through development of the app. This includes the entire feature set on the user side, such as visualizing open parking spaces and navigating to them. The main technical issue is that if the creation of the base app has complications or if the development environment is not set up for the members working on the app in a timely manner, then those members won't be able to start working on their tasks for the app. The design process will handle this issue by focusing the effort of all the members who will work on the app to work on setting up the base app in the first weeks of development. Another similar technical issue that could arise would be if the integration of the microprocessor and the physical components could not be done in the allocated time, then the communication between the meter and the app could not be tested. The design process would counter this by having the team work on the app and the database without communication with the meter and test with the completed meter later in the process and solve problems that arise then. A software test bench should be implemented to avoid the hardware dependency for the app development team.

Many aspects of the design have been finalized either with what component to use or what platform to use. The app will be developed for android devices using android studio. To navigate to spots or display open spots, the Google Maps API will be utilized. To accept payments from the user for the parking meter, the PayPal API will be utilized. Google Firebase will be used to receive data from the physical meter and to store it. The app will read the data from Google Firebase and visualize it. A full list of the electronic components and mechanical hardware has been selected and can be seen in Table 7.2. A prototype of what the app will look like has been developed and can be observed in figures 4.2.1 to 4.2.4. The figures show the proposed pages of the app.

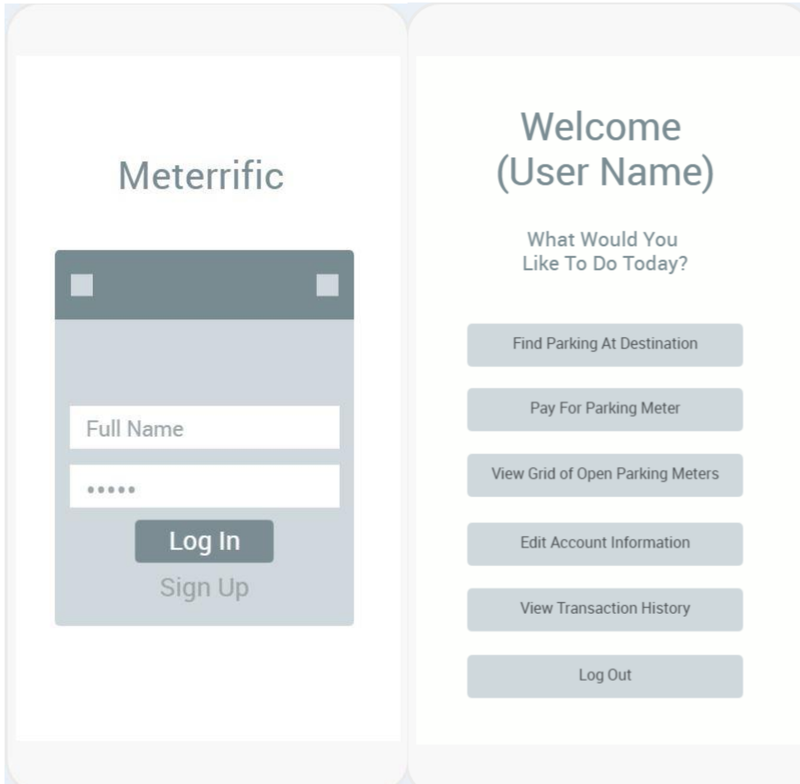


Figure 4.2.1: Prototype for the app. On the left, login page for the app. On the right, home page of app with button leading each of the other pages.

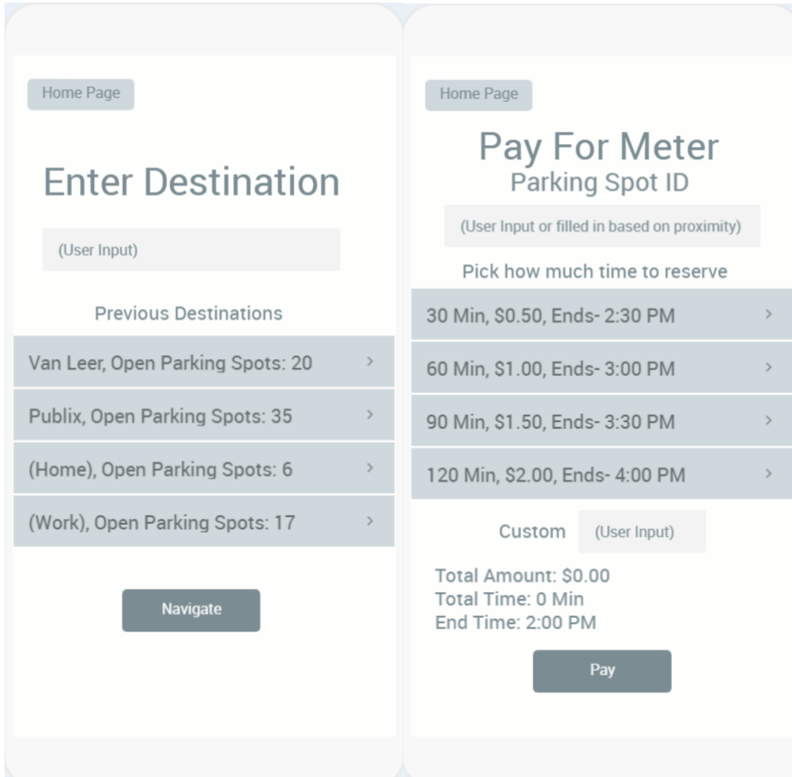


Figure 4.2.2: Prototype for the app. On the left, the page to start navigating to a destination. On the right, the page to pay for a parking meter.

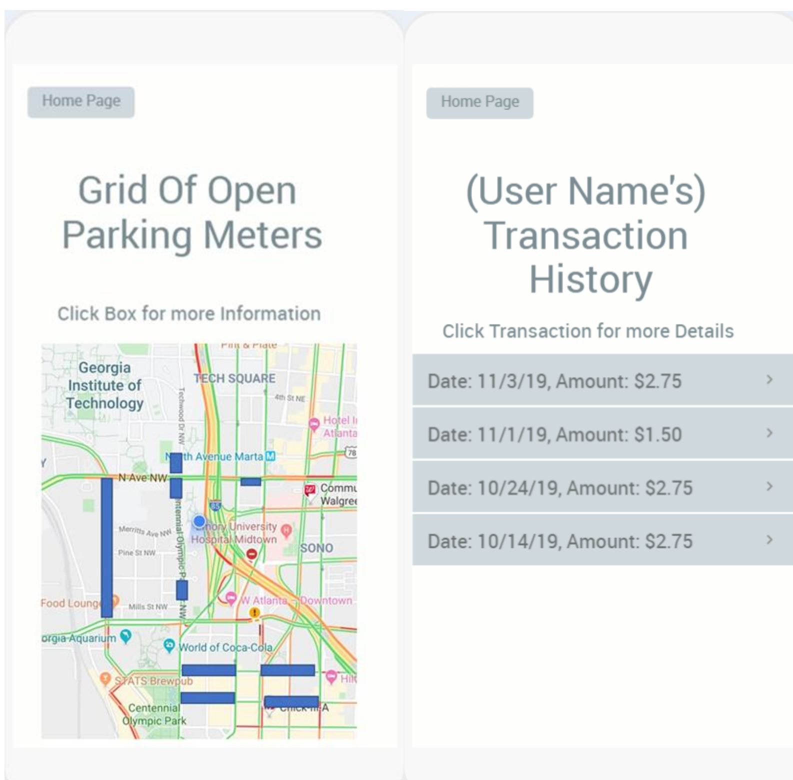


Figure 4.2.3: Prototype for the app. On the left, the page to start navigating to a destination . On the right, the page to pay for a parking meter.

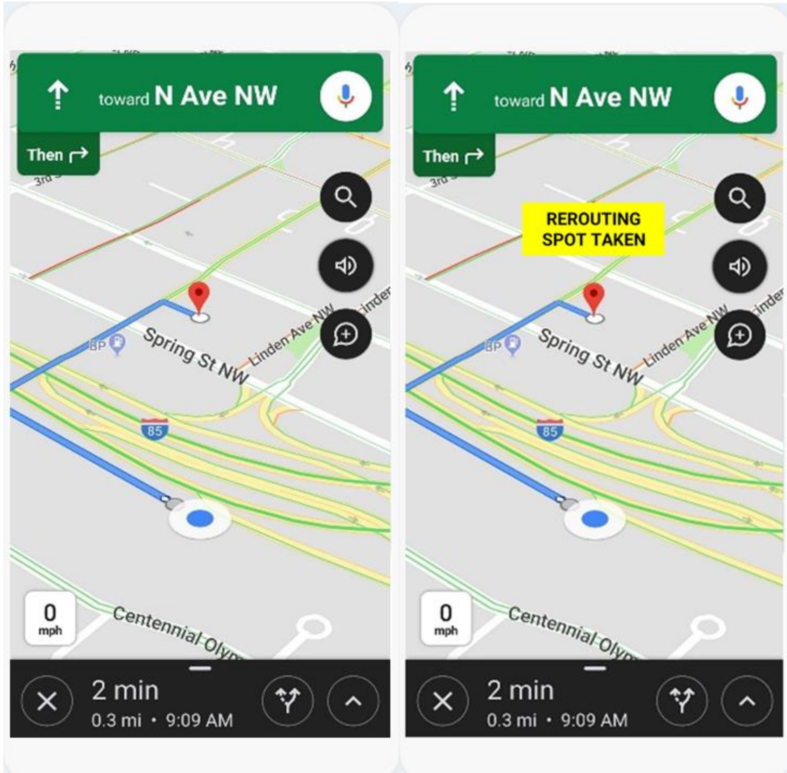


Figure 4.2.4: Prototype for the app. The left image shows the navigation page to the destination and the right image shows the navigation page when it has to reroute when the parking spot is taken.

A contingency plan must be in place for the failure of all tasks in the development process. Figure 4.2.5 shows the contingency plan for the case in which setting up the microprocessor with the allocated power doesn't work and the case in which developing the base Android environment doesn’t work. Both of these tasks have future tasks that rely on then. To reduce the delay caused by these failures, the dependent tasks are to be worked on with wire-frame code while a new environment is developed and then integrated together once the environment is set up.

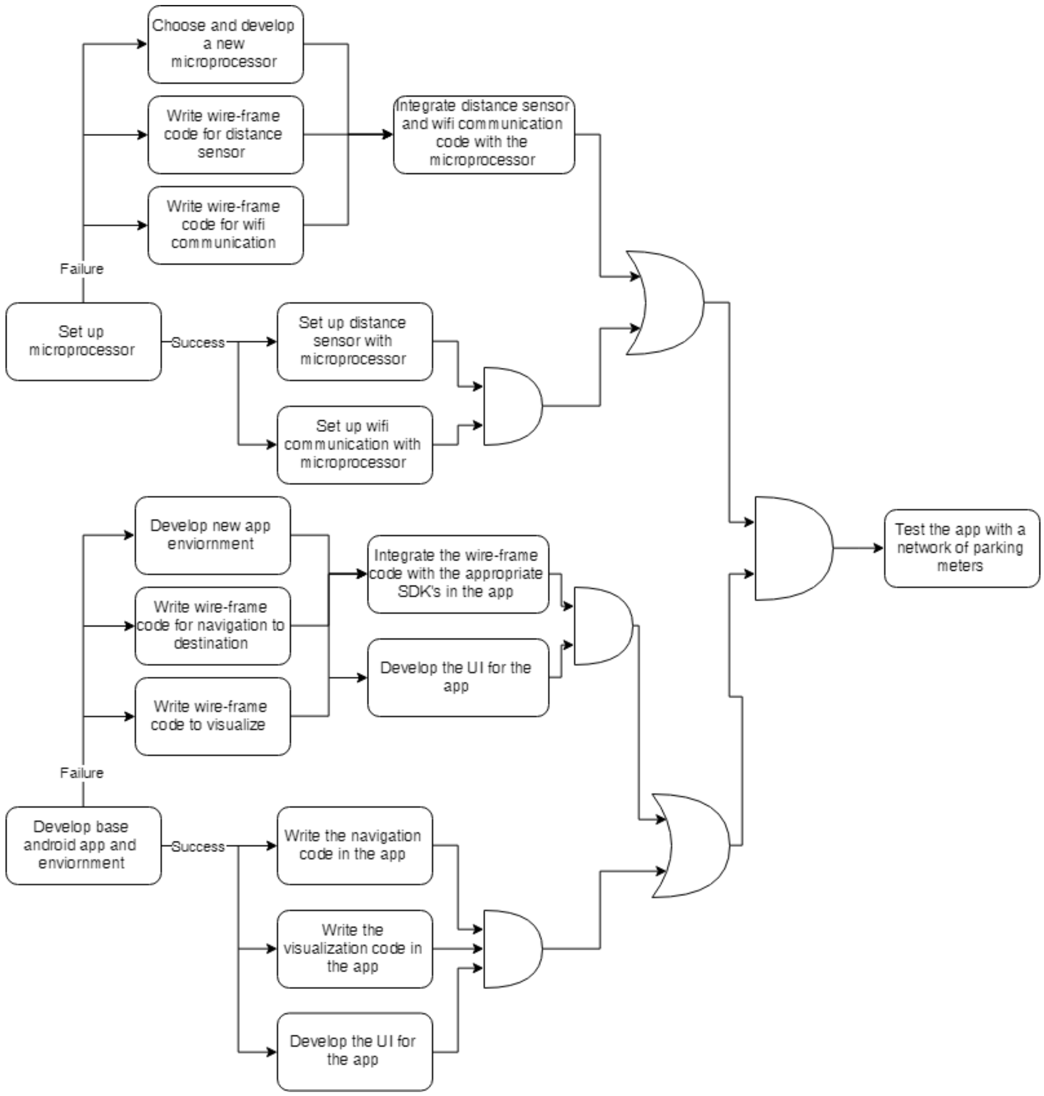


Figure 4.2.5: Block diagram showing contingency play for developing the app and the microprocessor

Some of the aspects in our design have yet to be finalized. The design of the electronics housing, the specific mounting patterns for the electronics, and the user interface of the parking meter has not been determined yet. The platform to develop the PCB has yet to be selected, although Autodesk Eagle is a strong candidate.

Many of the engineering design specifications have been determined based upon existing industry standards. Wi-Fi communication shall be used to upload data to the Google Firebase database using the firebase libraries. Data will be uploaded in a JSON format to make it easy to obtain information from. The Android app will similarly use Wi-Fi to retrieve the data. Our parking meter solution will map parking meters to locations based on the Standard Positioning Service standards. Most of the technical specifications have been defined in the tables in part 3.

4.3. Engineering Analyses and Experiment

During the design process, the electronic hardware will be prototyped using a breadboard for connections and DC power supply for consistent energy. Since the design will consist mostly of I/O devices and pull-up resistors, power consumption will mostly be dependent on the design of the firmware. As the firmware is updated for low-power consumption, the current required by the electronic hardware will be monitored through how much is drawn from the DC power supply. The goal would be to have the electronic hardware consume less power than is generated by solar power. To this end, the current drawn by the electronic hardware will be compared to the average power output of the solar cell as specified in its manual.

A surplus of energy from solar power would allow the rechargeable batteries to charge throughout the day and enable nighttime operation. As such, the surplus of energy over an hour would be monitored to determine the practicality of storing enough energy to sustain the smart parking meter overnight until dawn. In an ideal case, the surplus should follow the following relationships.

$$Power_{stored\ in\ rechargeable\ batteries} = Power_{generated\ by\ solar} - Power_{consumed\ by\ electronics}$$

$$\begin{aligned} Electric\ Charge_{consumed\ by\ electronics\ overnight} &= Power_{stored\ in\ rechargeable\ batteries}\ Time_{dusk\ to\ dawn} \\ &= (Power_{generated\ by\ solar} - Power_{consumed\ by\ electronics})\ Time_{dusk\ to\ dawn} \end{aligned}$$

4.4. Codes and Standards

The most significant codes and standards that apply to the smart parking meter design are listed below.

- Section 150-94: all-night parking. This code states that no person shall park between the hours of 1:00 AM and 6:00 AM in a particular area of Atlanta [4]. This code affects the operation and timing of the smart parking meters. The smart parking meters will not accept payments in these areas during restricted time frame.
- C57.12.28-2014 - IEEE Standard for Pad-Mounted Equipment--Enclosure Integrity. This standard specifies enclosure specifications for high voltage electronics located outdoors [5]. Although the smart parking meter design is not high voltage, these environmental protection techniques will still apply.
- GPS Standard Positioning Service (SPS) Performance Standard-4th Edition. This standard specifies the accuracy expected from GPS when integrating location data in the app [6].
- IMT-2000. This group of standards sets the frequency and data transmission protocols used for 1G, 2G, and 3G communication [7]. The smart parking meter design required Internet connectivity over wireless networks, so the standards are essential to understand the nature of the transmission the meter must use to transmit and receive data.

5. Project Demonstration

The demonstration will showcase the project's ability to:

1. Allow a user to start a parking session with mobile-based payment
2. Navigate a user to a parking spot
3. Reroute navigation when destination is taken by non-app user
4. Prompt a user to end parking session
5. Display parking session summary to user

The demonstration will accomplish the above by featuring an app user navigating to a parking spot and a non-app user competing for the same spot. The resolution of the conflict between the two users will demonstrate (1) and (2), and the successful parking session of the app-user will demonstrate (3) and (4).

6. Schedule, Tasks, and Milestones

Meterrific will be developed starting in the Spring 2020 semester from January to April. Appendix B contains a table of preliminary tasks, the leads for those tasks, and relative risk levels. Appendix C displays a GANTT chart of all relevant tasks and milestones with their expected timelines. The critical path for the project is highlighted in red. Finally, Appendix D displays a PERT chart of all relevant tasks and milestones with their worst-case, expected, and best-case timelines along with the critical path shown with red lines. Appendix E contains a table of time expectations and delays for tasks. Based on these expectations, the probability that the project will be finished before the GT Capstone Expo is about 0.376%. However, this also includes finalizing the Project Proposal and Summary before the Expo, but the final due dates for these items will be after the Expo.

7. Marketing and Cost Analysis

7.1. Marketing Analysis

The market for parking meters is different than the traditional user-based commercial market typical of IoT devices, as government entities typically choose a supplier of parking meters for either a city sector or an entire city [8]. Rather than targeting individual users, Meteriffic will target these government entities and provide evidence that the product is sufficiently durable, affordable, and has the optimal feature set for end users. Meteriffic's main competitors are traditional single-space parking meters, smart single-space parking meters, and multi-space parking systems. Traditional single-space parking meters are less expensive than Meteriffic's solution, but limited in feature set, as they do not implement internet connectivity [9]. Duncan is the main manufacturer of traditional single-space meters. Smart single-space parking meters are similar in price to Meteriffic's solution, but lack key features such as parking space navigation and usage of parking meters without an app. Meteriffic's solution would provide these features. Parkmobile implements the most widely-used smart single-space parking meter solution, which allows for space reservation and real-time pay-to-park capability [10]. Multi-space parking solutions are more expensive than Meterrific's solution, but do not track the status of independent parking spaces or provide navigation to free parking spaces. Parkeon's Strada Evolution is one of the most dominant multi-space parking solutions [11]. A complete table of costs for competing parking solutions can be found in table 7.1.

Table 7.1 Cost Comparison for Competing Parking Solutions [8]

Product	Overall Cost (Est.)	Cost Per Parking Space (Est.)
Duncan Parking Meter	$300	$300
Parkmobile	$700	$700
Parkeon	$10,000	$1,000

7.2. Cost Analysis

The materials cost for one prototype of Meterrific's smart parking meter solution is $82.10. The team plans on manufacturing two prototype designs for demonstration purposes and picked components with the goal of keeping the overall materials cost under a funding limit of $200. The predicted materials costs are illustrated in Table 7.2 below. The largest single component of the prototype materials cost is the PCB printing, which is necessary for a marketable product, as PCB electronics packages are smaller and easier to produce than proto-board designs. Many components required for a prototype design are already owned by members of Meterrific, which will help keep materials costs for prototyping below the target.

Table 7.2 Materials Cost

	Asset Name	Cost	Amount	Total Cost	Source
Electronics	LCD screen	$11.52	1	$11.52	Digikey
	LIDAR sensor	$0.00	1	$0.00	Team Supply
	Microprocessor	$3.50	1	$3.50	Digikey
	Wi-Fi Chip	$6.95	1	$6.95	Sparkfun
	Solar Panel	$5.99	2	$11.98	Xump
	LiPo Charge Controller	$6.95	1	$6.95	Digikey
	3.7V LiPo Battery	$0.00	1	$0.00	HIVE Makerspace
	PCB printing	$33.00	1	$33.00	Advanced Circuits
			Total Electronics:	$73.90	
Mechanical Hardware	ABS Filament	$0.00	1	$0.00	Interdisciplinary Design Commons
	PVC Rod (1ft)	$0.55	4	$2.20	Home Depot
	Mounting Hardware	$6.00	1	$6.00	Home Depot
			Total Hardware:	$8.20	
			Grand Total (per):	$82.10	
			Grand Total (x2):	$164.20	

In addition to materials costs for production of Meterrific's prototype smart parking meter, labor costs must be estimated. The majority of the feature set for Meterrific's solution is software, which does not have any direct materials cost. Additionally, there is time spent both in class and writing documentation that must be accounted for. Finally, time spent wiring and debugging electronics must be estimated. Table 7.3 shows estimates of time and cost for these categories, using $45/hour as a wage estimate. These estimates assume 14 weeks of work on the solution.

Table 7.3 Estimate of Labor Costs for Meterrific Prototype Development

	Weekly Labor Hours	Semester Total Labor Hours	Semester Total Labor Cost
Backend Development			
Embedded Software	3.5	49.00	$2,205.00
Data Transmission	1	14.00	$630.00
App Development			
User Interface	4	56.00	$2,520.00
Navigation	3	42.00	$1,890.00
Electronics			
Breadboard Testing	0.5	7.00	$315.00
PCB Design	2.5	35.00	$1,575.00
PCB Debugging	1	14.00	$630.00
Administrative			
Documentation	6	84.00	$3,780.00
Meeting Attendance	2	28.00	$1,260.00
		Total Cost:	$14,805.00

Finally, it is necessary to estimate a final figure for overall development costs for Meteriffic's proposed solution. The overall development cost estimate is $35,473.86. This figure is a combination of materials costs, labor expenses, fringe benefits, and overhead costs. The team has chosen a 25% fringe benefit allowance, as a percentage of labor expenses. The overhead cost estimate is 90% of overall project costs. This overhead cost is relatively low compared to industry standards discussed in class, as documentation and class meeting time have already been encapsulated into the budget. Table 7.4 shows the breakdown of overall costs.

Table 7.4 Overall Project Development Costs

Category	Cost
Materials	$164.20
Labor	$14,805.00
Fringe Benefits (as a percentage of labor cost)	$3701.25
Subtotal:	$18,670.45
Overhead (as a percentage of subtotal)	$16,803.41
Grand Total:	$35,473.86

The profit analysis for Meterrific's solution requires multiple cost assumptions in the areas of manufacturing, sales, and installation. The manufacturing of the solution will likely require technicians to perform the soldering of components to the printed circuit board. Additionally, the printed circuit board and the housing must be assembled. These assembly costs are estimated to equal 25% of the per-unit cost. Additionally, the installation of Meteriffic's solution will likely be expensive and vary depending on the location of installation. Due to the variation in installation technique, the team estimates installation costs at 50% of per-unit costs. Lastly, the sales cost will likely be modest, as single customers will purchase large quantities of the solution. Sales cost is estimated to equal 5% of per-unit costs. Assuming 1500 parking meters are sold over a 5-year period, the total number of parking meters sold is estimated at 7500. Meteriffic is targeting a $300 price for its solution, as this price is equal to traditional single space parking meters, but Meteriffic's solution has a much more diverse feature set. Table 7.5 shows the breakdown of profit. Costs are colored red and income is colored green.

Table 7.5 Per-Unit Profit Breakdown Based on 7500 Units Sold

Category	Value
Materials Costs	$82.10
Amortized Labor Costs	$1.97
Amortized Fringe Benefits	$0.49
Amortized Overhead	$2.24
Per-Unit Subtotal:	$86.80
Assembly	$21.70
Installation	$43.40
Sales	$4.43
Total Per-Unit Cost:	$156.33
Selling Price	$300.00
Per-Unit Profit:	$143.67
Total Profit:	$1,077,525.00

8. Current Status

The Meterrific product life cycle will be broken into stages and substages as follows.

1. Needs Analysis

2. Feasibility Study

3. Design

    3.1. Defining the problem

    3.2. Planning the project

    3.3. Gathering information

    3.4.. Conceptualizing alternative approaches

    3.5. Evaluating the alternatives

3.6. Selecting the preferred alternative

3.7. Communicating the design

5. Production

6. Distribution

The Meteriffic team has completed steps one and two. The team has also completed step three, but expects to iterate on the design in the future. During step four, the team expects to make usability, feasibility, and efficiency discoveries that will lead to changes in the final design of the product. As a result, steps 3.4 through 3.7 will be completed repetitively during step four. The team still needs to complete steps four, five and six.

The work completed to date is estimated to be thirty percent of the total work required for the Meteriffic product life cycle, as step four (not yet started) requires the most work.

9. Leadership Roles

The team has distributed specialized roles based on the skills discussed in the skills matrix. Shiv Chopra is the Expo Coordinator and the Co-App Development Lead with Blake Heard. As Co-App Development leads, Shiv and Blake will ensure performance, quality, and responsiveness of the applications. They will also identify bugs and build the Java code keeping in mind quality, organization, and automation. In addition to this role, Blake is the Documentation Coordinator for the team. Madison Hester is the App UI/UX Lead and the Co-Webmaster with Michael Knudson. As App UI/UX lead, Madison will ensure the quality of experience and effectiveness of the app design. As Co-Webmasters, Madison and Michael are responsible for creating an efficient and responsive web experience for users. Michael is also leading the Fabrication efforts where he will oversee the designing, modeling, printing, and construction of the electronics and hardware housing. Raj Patel is the Team Lead and Co-Firmware Lead with Andrew Trimper. Within team leadership, Raj will utilize interpersonal skills and organization to keep the team on track. As Co-Firmware Leads, Raj and Andrew are responsible for uC/uP programming and debugging, data interfaces and protocols (Wi-Fi, Bluetooth, GPS, UART, I2C, SPI), RTOS, and efficient, power-saving C/C++ code. Andrew is also the Hardware Lead where he is overseeing the assembly, wiring, and testing of hardware components including but not limited to wiring, soldering, circuit diagrams, and logic analyzers.

10. References

[1] "Vehicle Detection Sensors," *IPS Group Inc* . [Online]. Available: https://www.ipsgroupinc.com/products/vehicle-detection/. [Accessed: 16-Nov-2019].

[2] "Parking Payment and Enforcement Technologies," *Town of Concord*. [Online]. Available: https://concordma.gov/DocumentCenter/View/1495/Concord-Technology-Memo-PDF. [Accessed: 16-Nov-2019].

[3] "Parking Meter Pricing," *Metropolitan Area Planning Council*. [Online]. Available: http://www.mapc.org/wp-content/uploads/2018/03/IPS-MAPC-2018-Parking-Meter-Pricing-Matrix.pdf. [Accessed: 16-Nov-2019].

[4] "Listing of Regulations ," *ATL Plus Mobility – Atlanta Parking*. [Online]. Available: https://www.atlplusmobility.com/parking-regulations/. [Accessed: 15-Nov-2019].

[5] *Pad-Mounted Equipment - Enclosure Integrity*, IEEE C57.12.28, September 2005. [Online]. Available: https://standards.ieee.org/standard/C57_12_28-2014.html

[6] *Global Positioning System Standard Positioning Service Performance Standard*. Assistant Secretary of Defense for Command, Control, Communications, and Intelligence, 2008.

[7] "About Mobile Technology and IMT-2000," *About Mobile Technology and IMT-2000*, 04-Apr-2011. [Online]. Available: https://www.itu.int/osg/spu/imt-2000/technology.html. [Accessed: 18-Nov-2019].

[8] F. Martin, "Single-Space Meters vs. Pay-and-Display Systems," *Parking Today*, Feb-2004. [Online]. Available: https://www.parkingtoday.com/articledetails.php?id=55&t=single-space -meters-vs-pay-and-display-systems. [Accessed: 15-Nov-2019].

[9] "Single Space Smart Parking Meters: CivicSmart, Inc.," *Civic Smart*. [Online]. Available: https://www.civicsmart.com/new-liberty-next-gen-meter. [Accessed: 18-Nov-2019].

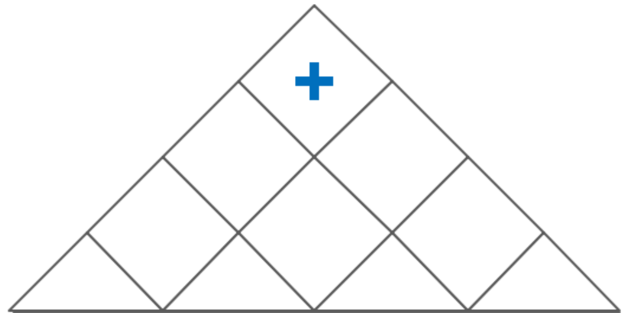
[10] "ParkMobile- A Smarter Way to Park," *ParkMobile*. [Online]. Available: https://parkmobile.io/. [Accessed: 18-Nov-2019].

[11] "Strada Evolution," *Parkeon United States*. [Online]. Available: https://www.parkeon.us/our-solutions/product-catalogue/strada-evolution-2/. [Accessed: 18-Nov-2019].

"M5™ Single-Space Meters IPS Group US," *IPS Group US*. [Online]. Available: https://www.ipsgroupinc.com/products/m5-single-space-meters/. [Accessed: 16-Nov-2019].

"Find and Pay for the Perfect Spot," *ParkMe*. [Online]. Available: https://www.parkme.com/. [Accessed: 16-Nov-2019].

Appendix A

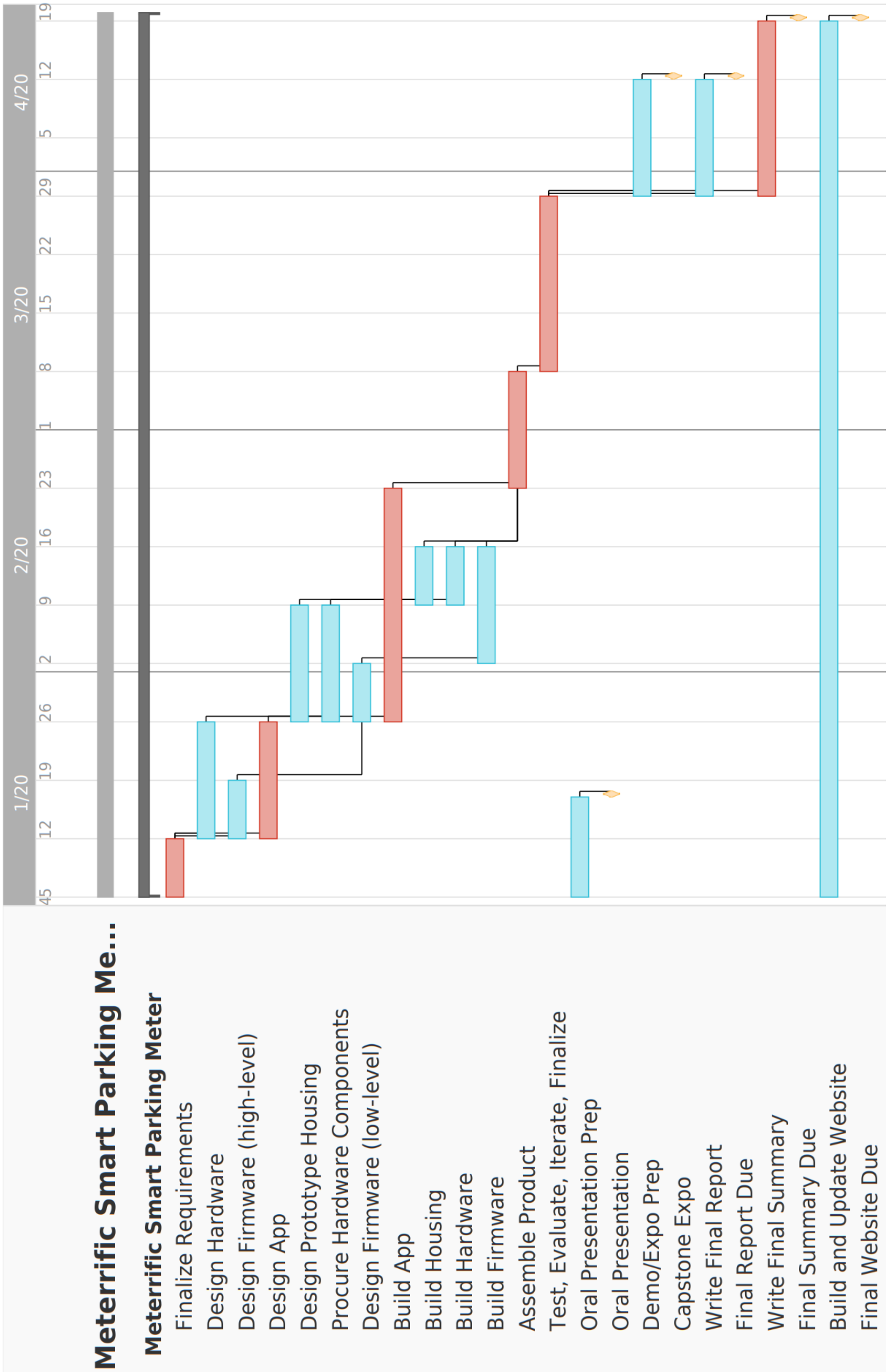


	Maps API	Store User Accounts in a Database	+ Car Detection Accuracy	+ GPS Location Accuracy
Seamless Navigation	+			
Parking Spot Close to Destination	+			+
Easy Payment		+		
Accuracy on Taken Spots			+	
Accuracy of Parking Spot from Navigation				+

Appendix B

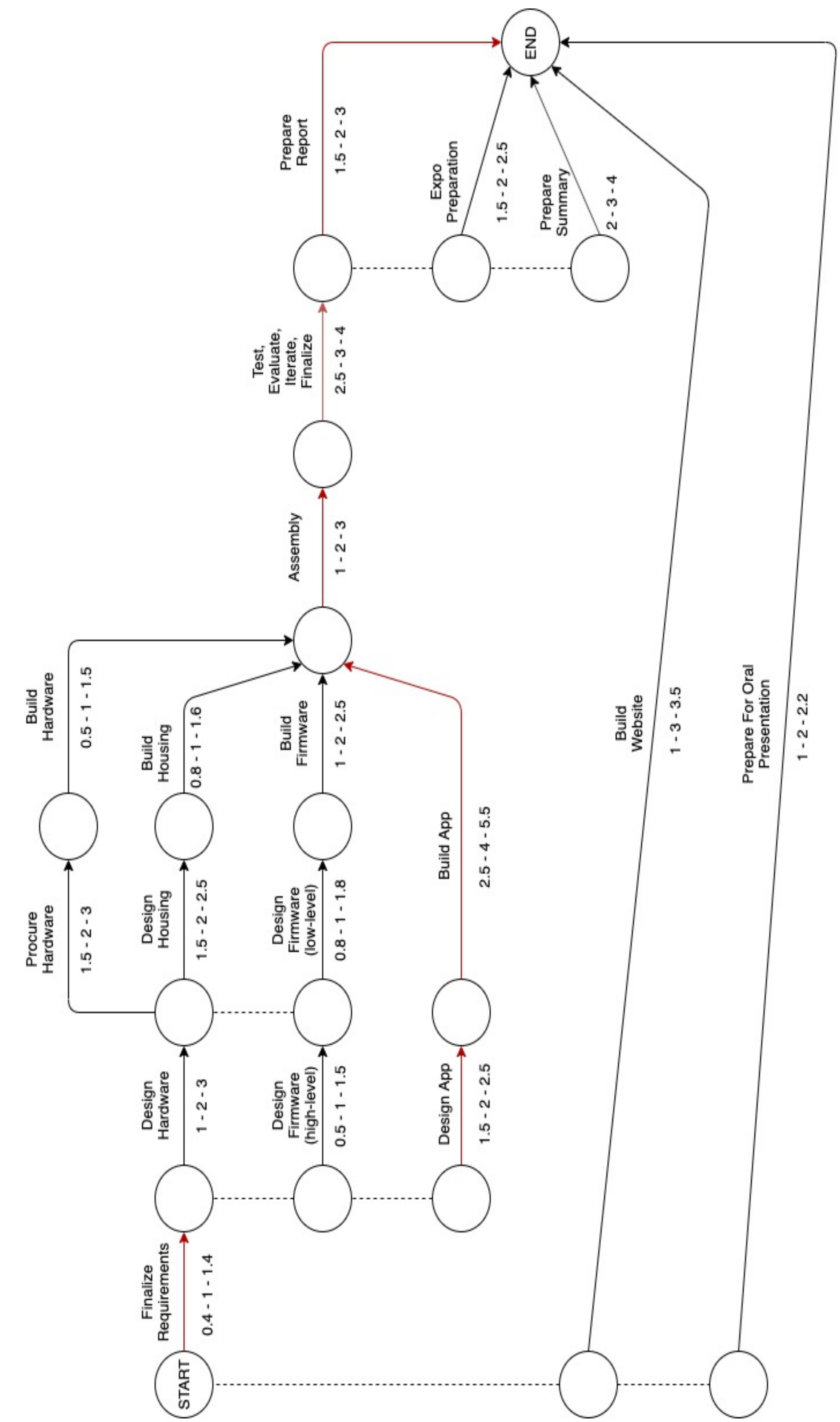
Task	Task Lead	Risk Level
Finalize Requirements	All	Low
Design Hardware	Andrew	Medium
Design Firmware (high-level)	Andrew, Raj	Low
Design App	Madison, Blake, Shiv	High
Design Prototype Housing	Michael	Low
Procure Hardware	Andrew	Low
Design Firmware (low-level)	Andrew, Raj	High
Build App	Blake, Shiv	High
Build Housing	Micheal	Medium
Build Hardware	Andrew	High
Build Firmware	Andrew, Raj	Medium
Assemble Product	All	Low
Test, Evaluate, Iterate, Finalize	All	High
Oral Presentation Prep	All	Low
Demo/Expo Prep	All	Medium
Final Report	All	Low
Final Summary	All	Low
Build/Update Website	Madison, Michael	Low

Appendix C



1/20
2/20
3/20
4/20
Meterrific Smart Parking Me...
Meterrific Smart Parking Meter
Finalize Requirements
Design Hardware
Design Firmware (high-level)
Design App
Design Prototype Housing
Procure Hardware Components
Design Firmware (low-level)
Build App
Build Housing
Build Hardware
Build Firmware
Assemble Product
Test, Evaluate, Iterate, Finalize
Oral Presentation Prep
Oral Presentation
Demo/Expo Prep
Capstone Expo
Write Final Report
Final Report Due
Write Final Summary
Final Summary Due
Build and Update Website
Final Website Due

Appendix D


START
Finalize Requirements
0.4 - 1 - 1.4
Design Hardware
1 - 2 - 3
Design Firmware (high-level)
0.5 - 1 - 1.5
Design App
1.5 - 2 - 2.5
Procure Hardware
1.5 - 2 - 3
Design Housing
1.5 - 2 - 2.5
Design Firmware (low-level)
0.8 - 1 - 1.8
Build App
2.5 - 4 - 5.5
Build Hardware
0.5 - 1 - 1.5
Build Housing
0.8 - 1 - 1.6
Build Firmware
1 - 2 - 2.5
Assembly
1 - 2 - 3
Test, Evaluate, Iterate, Finalize
2.5 - 3 - 4
Prepare Report
1.5 - 2 - 3
Expo Preparation
1.5 - 2 - 2.5
Prepare Summary
2 - 3 - 4
Build Website
1 - 3 - 3.5
Prepare For Oral Presentation
1 - 2 - 2.2
END

Appendix E

Task	Expected Time	EES	LS	FLOAT	Std. Dev.
Finalize Requirements	0.97	0.00	0.00	0.00	0.17
Design Hardware	2.00	0.97	1.89	0.92	0.33
Design Firmware (high-level)	1.00	0.97	2.90	1.93	0.17
Design App	2.00	0.97	0.97	0.00	0.17
Design Housing	2.00	2.97	3.90	0.93	0.17
Procure Hardware	2.08	2.97	3.89	0.92	0.25
Design Firmware (low-level)	1.10	1.97	3.90	1.93	0.17
Build App	4.00	2.97	2.97	0.00	0.50
Build Housing	1.07	4.97	5.90	0.93	0.13
Build Firmware	1.92	3.07	5.00	1.93	0.25
Build Hardware	1.00	5.05	5.97	0.92	0.17
Assembly	2.00	6.97	6.97	0.00	0.33
Test, Evaluate, Iterate, Finalize	3.08	8.97	8.97	0.00	0.25
Oral Presentation	1.87	0.00	13.80	13.80	0.20
Demo/Expo	2.00	12.05	13.05	1.00	0.17
Final Report	2.08	12.05	12.97	0.92	0.25
Summary	3.00	12.05	12.05	0.00	0.33
Website	2.75	12.05	12.30	0.25	0.42