|  |  |
| --- | --- |
| **Name:** Erik Meurrens, Benjamin Simonson, Ryan Jalloul, Evan Tobon, Samer Khatib  **Date:** 22 October 2024  **Team:** Parking Availability System (PAS) - ParKings | **Upcoming Milestone:** Pre-Alpha Build  **Stakeholder:** Raven Kim Ramos |

|  |  |
| --- | --- |
| **Goals for upcoming milestone** | Produce a list of tangible goals for the upcoming sprint. Include a title and brief description of the goal, an estimated allotment of time, and a description of how it relates to the upcoming milestone specification. |

* Attach camera hardware to board (1 story point)
  + Simple task to attach camera module to board. Since the camera module is intended to interface with the RPi, this should be super simple, but frequently things that should be super simple don’t turn out so simple . . .
  + Necessary before a video stream or image capture can be established
* Prepare app repository (2 story points)
  + Put mobile app prototype code into GitHub repository and ensure the code still works (it’s been a while)
  + Necessary to allow multiple members to work on the app
* Establish connection from App to SQL database (5 story points)
  + We need to prepare code in the app prototype that will interface with the AWS backend to demo communication between system layers for the pre-alpha build report
  + Necessary to have this communication established since the application is how users will access the real-time parking counter
* Establish connection from RPi to SQL database (5 story points)
  + Similar as above
* Get image capture frame from camera. Group selfie? 😊 (3 story points)
  + Ensures that we can interface with the camera module and obtain and store data from the camera module.
  + Necessary step before we begin processing image frames through the object detection model

|  |  |
| --- | --- |
| **Achievements since former milestone** | Report on the status of your progress since the last worksheet was filled out. Include a description and explanation for why each target was or was not accomplished. Estimate the time spent on each. |

* Set up RPi environment to begin programming remotely (Installed and set up OS, set up connection to UF network, installed some relevant libraries (numpy, matplotlib, opencv, . . . ), installed jupyter notebook, documented process for accessing the board remotely
  + Accomplished (8 story points, time ~ 10 hours)
  + Multiple tasks in the design plan were combined into one achievement. These include:
    - “Configure software backbone to RPi”
    - “Connect RPi to UF WiFi”
    - “Configure remote access to RPi for development”
  + This task was necessary to be completed before next steps so that members responsible for programming the board can program the board remotely without needing to be in possession of the board. Sets up a software backbone to test Python code through a Jupyter Notebook server that can be accessed remotely
* Acquired hardware needed to begin setting up edge device (RPi, RPi camera module)
  + Partially accomplished (3 story points, time ~ at least 1 hour)
  + The most important hardware is purchased and obtained: the board itself and the camera module
  + The sensors are not immediately necessary, but will be needed later.
  + May also need a heat sink. Just downloading and installing packages makes the processor get toasty.