Surveillance Tracking System

Internet of Talents
Truong Xuan Nguyen
Fiona Lin
Christopher Beckett

Content

Project Outline

Purpose & Background

The use of private security has been exploding in Australia, with increased demand comes increased cost. Conventional alarm and surveillance systems are fixed in place, and are not able to adapt to changing environments. This requires the use of private security officers. With the majority of private security officers paid between $600-799/week[1]. Increasing the effective coverage area of each officer stands to drive major savings.

Traditional surveillance systems, suffer from problems:

- They are fixed in place.
- They are resource intensive, requiring:
    - Cameras, including expensive low light sensors
    - Cabling

A surveillance tracking system using IoT aims to be:

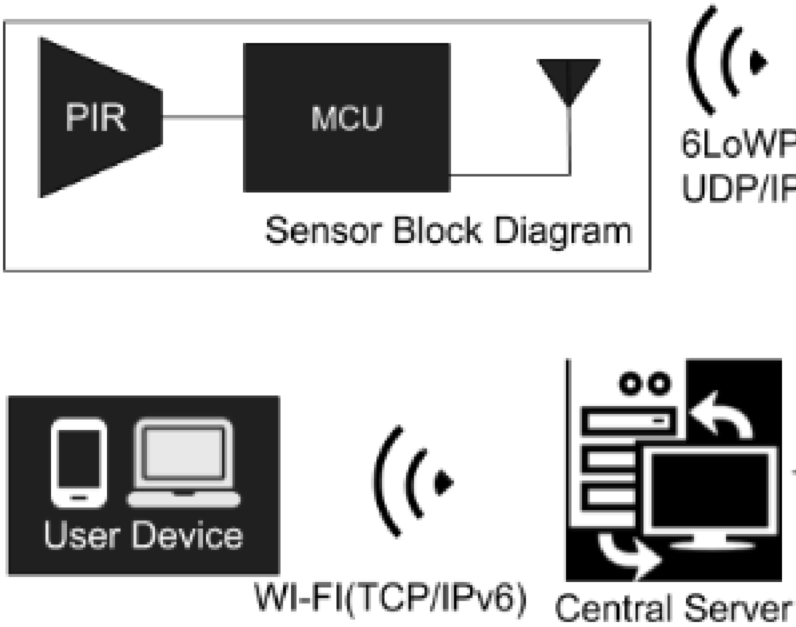
- Highly accurate, allowing less human oversight
- Portable, allowing it to monitor changing locations
- Without the need for expensive vision systems

Objectives

- Evaluate the pros and cons in comparison with other traditional fixed active surveillance systems.
- Design and prototype a surveillance system passively track a single human moving through its monitored space (a classroom sized space).
- Notify and present the tracking movements vividly to end users.

[1] Prenzler T, Earle K & Sarre R. 2009. *Private security in Australia: trends and key characteristics*. Trends & issues in crime and criminal justice No. 374. Canberra: Australian Institute of Criminology. https://aic.gov.au/publications/tandi/tandi374

System Architecture



Hardware Requirement

Sensor block

Each sensor block contains a passive infrared sensor, connected to a yet to be chosen microcontroller system, and wireless networking hardware implementing the 6LoWPAN network stack. Where will be a number of sensor blocks deployed into the monitored area.

Local server

The local server consists of a sensortag running the 6LoWPAN border router, and connected to a raspberry pi via serial connection. It validates and collects signal data from all the sensors block and provides additional information about the signal location.

Central server

A virtual machine running Linux Mint OS hosted in a cloud provider or physical hardware is used to consume data sent by local server about a sensor cluster, and provide the HTTP interface for the end user devices.

User Device

Any modern device running a web browser can be used to view the output from the sensor network.

Software Requirement

Sensor block

Passive InfraRed sensors detect object movement though changes in IR radiation. The connected MCU will poll the PIR waiting for movement to be detected, at which point it will notify the local server of its sensor ID via 6LoWPAN. As energy usage is a major concern in a portable system, experiments will be conducted to find the lowest polling timing that can effectively detect a human sized object moving through the monitored space.

Local server

A raspberry pi will serve as a local data collection server, where it will collect sensor data from the sensor nodes in the local cluster, and transmit that to the central server.
It will be connected to a border router via serial to collect the reported data over 6LoWPAN for energy efficiency.

Central server

This component mainly deal with the localization and solving object detection algorithm to identify the exact spot of the object in the monitored space. The more accurate the tracking needs, the more computation complexity this mapping requires, therefore it is isolated out of local server. By doing this, it allows the flexibility to cater different accuracy needs

User Device

A well designed UI web app written in React framework will be developed to present the notifications and location of object in the monitored space

Localisation (detection algorithms)

Once the sensor nodes are set up, the user will configure where they are located in relation to each other, this will allow the software to detect the location of the intruding object by finding the only intersection of sensor coverage and displaying that similar to a Venn diagram on a map of the area.

Testing system

A testing system has proposed to accompany the implementation of project plan and achieve all milestones as required. It is made of five unit tests to ensure each component works as expected, and three integration tests to assure related components exchange and collaborate information and data integrity as designed. During surveillance system is developing, the testing system will vary to suit the needs of system developments.

Unit tests will break up according to the workflow process listing below:

Sensor block detecting objects

It is required to detect the changing signal in the monitored space while the object is presented, as well as report detected signal and location information via wifi module.

Local server receive, validate and store sensor data

It is required to receive to reported data and information, and validate incoming data and store sensor data for batch forwarding.

Central server object detection/location algorithm

Since all the reported data and information has collected, the detection/localization algorithm is needed to be as effective and accurate as the system needs.

Central server push notification of object detection

Notification is needed to present to end user as the location of object has computed within certain allowed time legacy

Central server showing object location to end user via web app

An acceptable user interface in web app shows the detected object in the monitored space

Three mainly integration tests will develop to check the coordination of related components:

Sensor blocks and local server integration

It ensures the sensor data reporting from sensor blocks to local server and the fluency of each sensor block traffic flow to local server.

Local server and central server integration

It assures the integrity and accuracy of data flow from local server to central server.

Central server push notification and user devices integration

It makes sure the end result will come to end user to track the unexpected object in the monitored space.

Project Plan

To ensure a smooth and efficient implementation of the project, the overall plan is divided into 3 smaller stages:

1. Stage 1: System architecture proposal and project planning.
2. Stage 2: Product development.
3. Stage 3: Prototyping, testing and debug.

In stage 1, the primary goal is to achieve a clear understanding of the proposed system amongst the team members and identify the milestones that can help keep track of the progress. This is really important because it sets out a solid foundation for future development, allowing the team members to parallelize the subsequent tasks while maintaining a single focus on the end-goal of the project. The activities involved in stage 1 includes refining the project scope, proposing a potential framework to evaluate any future implementations, experimenting and selecting the required hardware (microcontroller, wireless connection modules, associated equipment, etc.) and software (front-end and back-end technologies, communications protocol, programming language, etc.).

Stage 2 is the main phase of the project. In this stage, the team will focus on developing and implementing the solution proposed in stage 1. Because of the limited timeframe of the project, the team shall be splitted into 2 branches in which one branch accounts for the back-end development and the other branch is in charge of the front-end development. Within the context of this project, the back-end refers to the design, features and inter-communications of the sensor blocks and the front-end is anything related to the central server and the user interface.

Stage 3 is when the final prototype of the proposed system is delivered. All the sub-systems developed in the earlier stages shall be aggregated at this stage to provide a fully functional prototype. It is also a requirement of this stage that the team shall prepare a test environment where the prototype can be deployed and demonstrated. The testing and debugging of the proposed system are also the two main activities of this stage.

Key Milestones

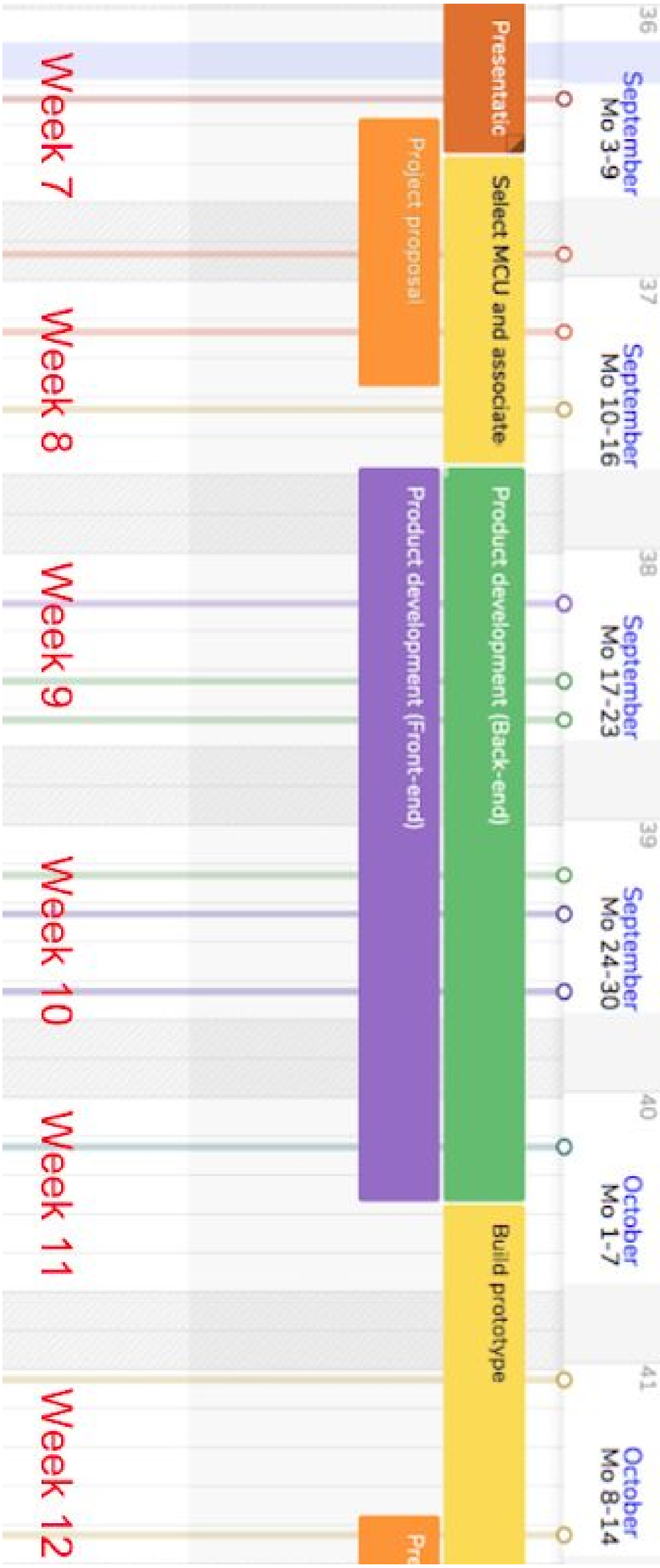
The table below summarises the key milestones of the project:

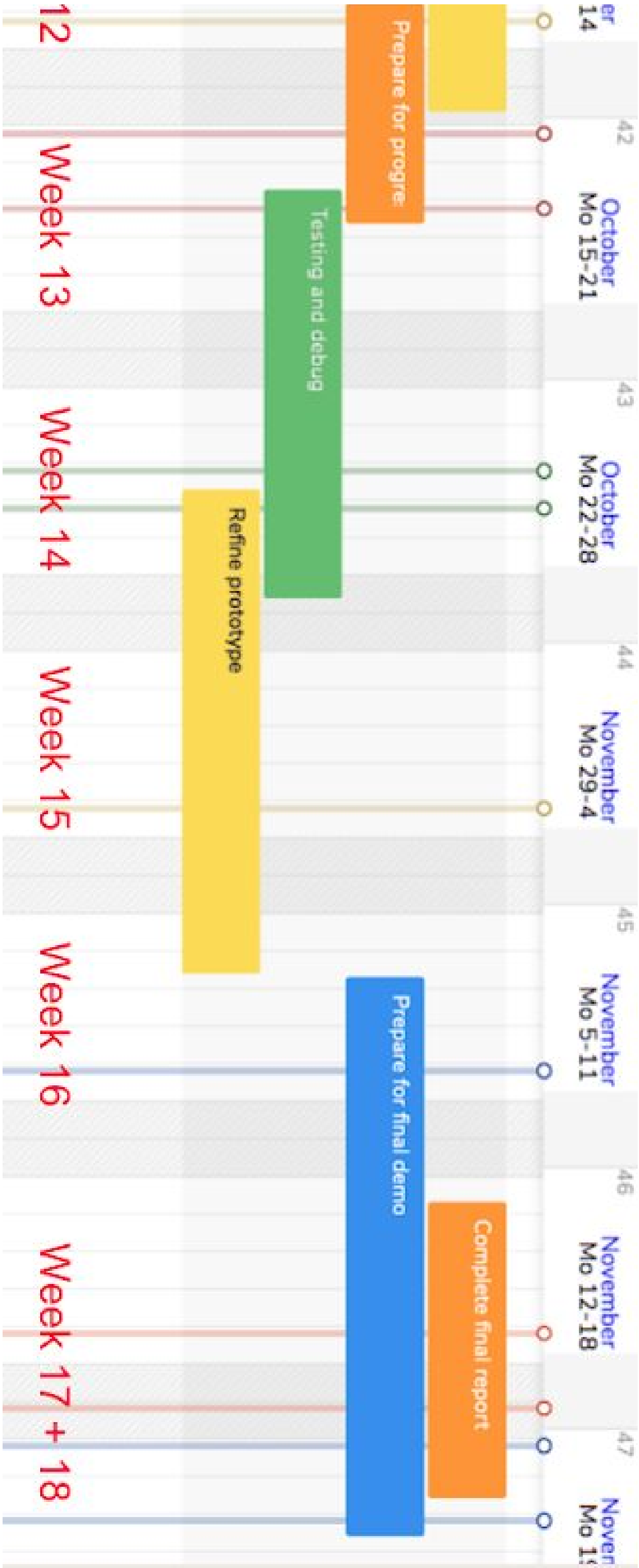
Milestones	Due	Responsibility
Select and obtain all hardware	Week 08	TN
Finalise wireless communications protocol	Week 09	CB + FL
Build first sensor block	Week 09	TN
Server ready to receive, aggregate and present data	Week 10	CB + FL
Backend - Frontend integration	Week 10	TN + CB
First prototype ready for initial deployment	Week 12	TN + FL

Final prototype ready for deployment	Week 15	TN + FL
Deploy final prototype to classroom environment	Week 16	CB
In-class showcase and final report due	Week 18	CB

Gantt Chart

The provisional project plan can be found in the gantt-chart below:


36
September
Mo 3-9
37
September
Mo 10-16
38
September
Mo 17-23
39
September
Mo 24-30
40
October
Mo 1-7
41
October
Mo 8-14
Presentatic
Select MCU and associate
Product development (Back-end)
Build prototype
Project proposal
Product development (Front-end)
Week 7
Week 8
Week 9
Week 10
Week 11
Week 12



October
Mo 15-21
October
Mo 22-28
November
Mo 29-4
November
Mo 5-11
November
Mo 12-18
42
43
44
45
46
47
Prepare for progre
Testing and debug
Refine prototype
Prepare for final demo
Complete final report
12
Week 13
Week 14
Week 15
Week 16
Week 17 + 18