# Project Scope

## Project Statement

The aim of this project is to create a web application system to keep track of the exact number of intruders within a specific area using predefined sensor pattern and having the ability to access this system over the internet via cloud technologies.

## Key Technologies

### Google Firebase – Cloud based database

This is a cloud platform where the data is stored securely on google database. System administrator can select which country to store this data and restrict access of the database against intruders which is important.

### Beaglebone – IoT kit web server

This is the IoT hardware used to handle all the computation of the intruder detection. It takes the sensor reading and compute it and send the following data to firebase and store it there. It is also possible to store a local database for local use and backup purposes.

### Node.js – Programing Language

Node.js is a JavaScript programming language used for this project and it is used to program the computation of the intruder counter and relaying messages between Firebase and BeagleBone.

### HTML and CSS – The building blocks of webpages

HTML stands for hypertext markup language which allows the creation of web pages and web application but in a very basic form. These said webpage are then enhance visually from the help of CSS which stands for cascading style sheets.

### SSH – Secure Shell

This is a network protocol used to control the server remotely. This is mainly used because the BeagleBone hardware does not include a display and controller. So as system administrators we must remotely access the BeagleBone locally using USB port or over the internet via SSH from a laptop or desktop.

## Key Functionality

First, the system must be able to be access anywhere where internet is assessable. It must be able to record the total number of long motion and short motion and store this information in Firebase. These said long and short motion will be compute by BeagleBone to determine the accurate number of intruders within the area of operation. All these data will be shown to the users through a webpage in HTML design using CSS. Besides that, the users must be able to reset the database, turn the LED on or off and sensor on or off from the server.

## System Constraint

The system server will not be able to handle large number of users due to hardware limitations because it is using a IoT grade processor. False positive sensor reading might also occur from the sensor that could disrupt the system performance and accuracy.

## How the system works



1. Sensor sends a stream of reading to BeagleBone server
2. BeagleBone compute the data and send it to a network interface or switch
3. Data will be stored on firebase and could be stored locally for local access and backup purpose
4. Users than access a webpage created to view the data from Google Firebase via the internet.

# Risk Register

This risk register is scale from 1 to 5 for both likelihood of occurrence and impact towards the project. Besides that, the risk register is being separated into five different sub sections to classify them into their respected category of risks. In the risk register, it has risk prevention to show the precaution used to prevent this risk from happening and risk mitigation to show what steps will be taken when this risk happens to prevent further damage to the project.

### Likelihood Scale Description

1. Very low or no chance to occur
2. Low chance to occur
3. Medium chance to occur
4. High chance to occur
5. It will occur in given point of time within the project

### Impact Scale Description

1. Low or no impact towards the project
2. Slight impact but no issues with continuation of project
3. Medium impact and minor problem towards the project
4. High impact and major problem which should be important to fix
5. Catastrophe impact and must be resolve to continue the project.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk Description** | **Likelihood** | **Impact** | **Risk Prevention** | **Risk Mitigation** |
| **Human Factor** |  |  |  |  |
| Steep learning curve of new programming languages | 4 | 3 | Get to know and improve on the knowledge on the languages used for the projects | Change programming environment back to familiar languages |
| Lack of familiarity on Cloud Computing | 4 | 2 | Research on regarding topic before hand | Get support from team members |
| Lack of support from service providers | 1 | 3 | Not preventable | Get support from user community |
| Loss of team member | 1 | 5 | Constant communication and replacement is prepared | Replace team member |
| **Hardware Issues** |  |  |  |  |
| Hardware failure - Sensor/Beaglebone | 3 | 5 | Stress testing to ensure reliability | Replace hardware with similar or same hardware |
| Cloud computing outage- Electrical failure/system failure | 3 | 5 | Have a backup cloud computing service provider | Redirect traffic to backup cloud computing service provider |
| Compatibility with similar system in case of system environment change | 2 | 4 | Ensure programming languages and hardware used can be used on wide range of devices | Ensure the change has the best compatibility and future proofing |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk Description** | **Likelihood** | **Impact** | **Risk Prevention** | **Risk Mitigation** |
| **Security Issues** |  |  |  |  |
| Third Party Intrusion - gaining control of the system | 1 | 5 | Use a reputed cloud computing service provider with enhance security | Terminate any connection to server and retrace intrusion method used |
| Third Party Intrusion - data spoofing/stealing | 1 | 5 | Use a reputed cloud computing service provider with enhance security | Use a better data encryption and transmission method |
| Lack of control over data | 2 | 3 | Limit the amount of user having access over data | Enforce stricter policy regarding the usage of data and who to use it |
| **Legal Issues** |  |  |  |  |
| Location of data center | 1 | 2 | Select the best and closest location based on the users’ location | Choose the next best location for data center |
| Privacy and Legal Agreements | 1 | 5 | Make sure every implementation has been approved legal and compliance with laws and regulation | Edit the system to ensure compliance with law and regulation |
| **Connection/Reading Issues** |  |  |  |  |
| False positive sensor Reading | 3 | 5 | Stress testing to ensure reading are accurate on all circumstances | Revisit programming logic to ensure false positive reading does not occur again |
| Loss of connection between BeagleBone to firebase | 3 | 5 | Have backup connection between BeagleBone to Firebase | Use the next best connection between BeagleBone and firebase |