Wireless Sensor Networks

Design, Implementation and Documentation

Case Study:

The monitoring and forecasting of climate conditions play a crucial role in our everyday life. Technologies such as precision farming require precise climate monitoring and prediction to make informed decisions. The rise in global temperatures and incidents of extreme weather events highlights the need for a more stable weather system. Regular monitoring of climate parameters provides real-time insights that can assist in making informed decisions. To analyze the vast amount of climate data, various data analytics techniques are employed.

This project follows the climate conditions in Mars using data collected on the wireless Insight Lander. It allows the construction of models to output predictive results. The project highlights the software architecture as well as required risk assessments on performance, operation and application development.

Date: 10 Feb 2023

By Rebecca Duong as part of the application for position Platform Engineer – DroneShield Ltd.

Table of Contents

1 Introduction

In this report, we will discuss the application software that can be used to monitor the climatic conditions of Mars and predict the situation by using advanced technology. In this project, to monitor the climatic conditions, we will incorporate the use of a monitoring robot to work as wireless technology. We have used the software and hardware components to control the tracking of weather conditions technologies that have been discussed in this report. This system enables and ensures the mapping and monitoring of the accurate climatic outbreaks that can show us the impacts of climatic changes and explain us and facilitate the preparedness of plan to avoid the effects. However, the application used to integrate this monitoring system and observe the changes so that the situations can be controlled before impacts in the agricultural sector have been designed. The applications of our project have major roles in an agricultural sector like precision farming which rely on accuracy and prediction that assist with decision-making in this area of the economic sector. Additionally, ground-based information and the forecast models and monitoring of climate changes in the weather can also be traced and tracked by our dashboard application. We are planning to use Raspberry OS software alongside with Node-RED to run the dashboard of our application with a MYSQL database to manage the entire system and store the relevant climate data.

Also, we have discussed the projects different type of network sensor architecture hierarchies and the wireless network protocols that we have used in the connectivity of the system. We have also discussed the IoT system's assumptions while implementing this climate monitoring and prediction technology in the Mars exploration sector. We have also discussed its future improvement, like the availability and advancement in IoT as the technology is growing and engaging in various sectors. So, in the context of climate monitoring, the analysis has been broadly explained in this report.

2 Background

NASA’s Interior Exploration using Seismic Investigations, Geodesy and Heat Transport (InSight) Mission is a robotic lander that will place a single geophysical lander on Mars to study its interior. By using geophysical instruments, the InSight Lander will collect Mars’ atmospheric measurements such as wind, temperature, direction and pressure. Beneath Mars’ surface, Insight will measure the terrestrial formation using seismology, heat flow probe and precision tracking. It is the first outer space explorer to study Mars in-depth – its core, mantle and crust. Launched in May 2018, InSight has seen 849 sols (Mars’day) as of April 15, 2021.

The Insight Mission seeks to investigate the composition and interior structure of Mars. Its scientific goals aims to determine the rate and effect from meteorite impacts and Martian tectonic activity (Hoffman 2018). Previous missions on Mars involve in surface features like canyons, volcanoes and soil but none has delve deep into the surface – its foundation. The scientific reasons for us to study Mars is to search for life and preparation for human exploration on the Red Planet. Delving in Mars founding blocks advances human research in this direction for future applications in mining, tourism and technology innovation. As such, InSight’s primary objective is to study the earliest evolutionary processes that shaped Mars. Its singular Lander is based on the landing system designed for Mars Phoenix Lander to minimize costs and possible risks. The mission is launched on Atlas V-401 rocket from Vandenberg Airforce Base in California. InSight’s robotic arm is 1.8 metres long to lift a seismometer and heat flow probe from the deck to Mars’s surface. A camera on the robotic arm provides coloured 3D views of the instrument placement, landing site and ongoing activities. In total, InSight makes use of three science instruments:

- Seismic Experiment for Interior Structure (SEIS): Measuring marsquake orientation, acceleration via cartographic location;
- Heat Flow and Physical Properties Package (HP^3): A radiometer and heat flow probe designed to burrow into Mars’s soil with thermal sensors.
- Rotation and Interior Structure Experiment (RISE): Utilises the Lander’s radio to measure planetary rotation to better understand the interior of Mars (NASA 2019).

Scientists will interpret data sent via InSight: Mars Weather Service API to understand the planet’s history, planet interior and forces that shaped it. All activities are powered by solar panels and batteries.

Hibernation

Since 2018, InSight has detected more than 500 marsquake and started to measure the planet's core. As InSight relies on sunlight to power and maintain recurring activities via solar panels, a clean surface is crucial in generating the necessary power input. Periodic wind gusts called "cleaning events" help reduce Martian dust accumulation on the panels for ongoing science operations. However, the Red Planet's unpredictable weather since the start of 2021 with lesser wind gusts means fewer cleaning events is experienced. Elysium Planitia, InSight's landing site, is covered by a thick layer of dust on the panels. This result in an emergency hibernation at the start of Martian winter in February 2021. By pausing all scientific operations, the Lander is conserving energy to keep its system warm through the rigorous temperature drop at night (Lisano & Grover 2019). Currently, NASA preserves an optimistic view that the InSight Lander can come out of hibernation in July 2021 (NASA 2021).

The Death of Lander's 'Mole'

InSight's HP^3, developed and built by the German Aerospace Center, is a burrowing package to measure Mars internal temperature, providing details on Mars' evolution and geology using a heat engine. Since 2019, InSight has been attempting to scrape away the top layer of Martian soil to burrow the 'mole' (HP^3) to perform the intended experiment. In reality, this portion of the mission has ended in January 2021 because sufficient depth could not be reached. The soil's tendency to clump could not provide enough friction for the 'mole' to dig deep into the ground, even after efforts of hammering itself. A last attempt was made in January 2021 with 500 additional hammer strokes called for an end to InSight's efforts (NASA 2021).

Project Motivation

Our motivation to work with Mars's InSight Lander is to understand and utilize data collected via wireless sensors as part of advanced technology applications. On a technical aspect, InSight 's ability to monitor weather conditions via its Service Weather API shall give abundant information for us to supervise Mars' weather from Earth and make educated predictions using the collected data. The API provides a summary of data per-Sol (Martian days) for each of the last 7 Sols. During the initial period, InSight will report data back to Earth twice every day, down to once a day thereafter. Summary data is provided in JSON format.

As mentioned above with InSight's unique mission on measuring Mars' interior alongside seismic and thermal properties, we believe this project is a detailed comprehension of our knowledge learned. We tap into InSight's historical data to translate it into meaningful information as part of its purpose to further extend human knowledge of the universe. As such, we have decided to work on this project for this assignment.

3. Case Assessment

3.1 Aims and Objectives

Project aim: To monitor Mars weather conditions through wireless sensors and display them as meaningful information.

Objectives:
1. To retrieve data from sensor nodes in InSight: Mars Weather Service API securely
2. To display the filtered data through an application
3. To deploy the application and perform on-going support

3.2 Application Requirement

Functional requirements:
- The application is either a web-based application or mobile application
- The application should be integrated with InSight: Mars Weather Service API
- The application should be integrated with MySQL Database to store historical and current data
- Data is obtained via sensors available on Mars InSight Lander
- Data obtained is weather measurements per-sol (Martian days) for each of the last seven sols
- Summary data is in JSON format
- The application is hosted on a cloud service (TBC)
- The application should display data using dashboards
- The application theme is dark gray (#121212)
- The application should display historical and current data continuously
- If permitted, the application should predict future Mars weather measurements using historical data
- If permitted, the application uses a neural network to predict Mars weather
- Earth's operating system is Raspbian
- Mars InSight's operating system is InSight ROS
- Both operating systems are connected via Node-RED
- The application is fully tested before deployment
- The application shuts down in case of a cyber attack
- NASA staff maintains and monitor InSight
- The application is developed and maintain by the Earth team (allocated project managers and developers)

Non functional requirements:

- The application should load in 0.1 second (Miller 1968, Card et al. 1991)
- The application shall be easy to use by all user types
- Should there be a delay in data displays, a notification with a timer will show when the system is up again
- Access permissions to the underlying database and system configurations may only be changed by the system's administrator
- The database update process must roll back all related updates when any update fails
- Data updates must not interfere with current user's page view (continuous data flow)
- The application must be scalable to support 200,000 users at a time
- The Project adheres to all NASA licensing and other relevant licensing, including but not limited to InSight hardware, InSight: Mars Weather Service API, Engineering department Bylaws and Apache license 2.0.

4. Architecture

4.1 High level architecture design

The architecture of this system consists of both software and hardware. Concisely, the software part implements the function of controlling and adjustment according to requirements and running conditions. The hardware part enables launching and monitoring through powerful functions. There are corresponding middleware and firmware to ensure proper connection and running between software and hardware parts. This section demonstrates the 2 parts, respectively. A diagram for high-level architecture of the design is shown as following **Fig. 1**:

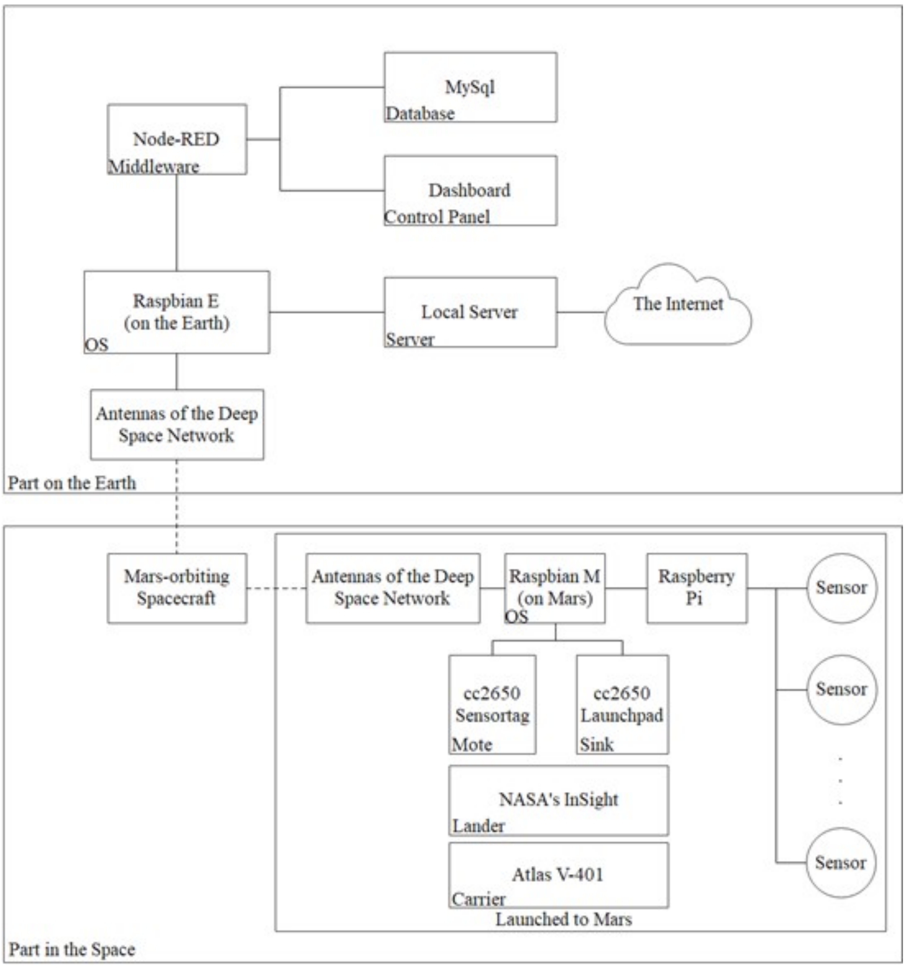


Figure 1. *High-level Architecture.*

The procedures of climate data display are 4 steps: 1) The sensors collect climate data on Mars and send the data to Raspbian M, 2) Raspbian M saves received climate data for temporary and send the real time data to Raspbian E through Mars-orbiting Spacecraft by using Antennas of the Deep Space Network on both sites, 3) Raspbian E displays received climate data on current time module of dashboard. Simultaneously, the data is saved into MySQL database with timestamp as history data which can be viewed on dashboard, 4) After the sum duration of history data is long enough for one sample capacity at least, the Raspbian E calculates the prediction of future climate on Mars with predefined algorithm and displays it on dashboard continuously.
As a service for displaying different types of climate data on Mars to clients, Raspbian E is connected to a local server so clients can visit and view the data based on certain protocols and standards.

4.2 Software Components

The prior task of software components is to implement the controlling of the whole system. Besides, certain firmware is utilized to interconnect each component and ensure the system running properly. Software components in this system include operating systems, database, virtualization services, middleware, and firmware.

4.2.1 Operating Systems

For the part on the earth, database and virtualization service are both connected to Raspbian via Node-RED. Besides, to provide services, there is a link between the Raspbian and a local server. The Raspbian sends and receives signals with the part in the space through antennas of the deep space network. On the other hand, the Raspbian can also support the data collecting system on Mars site. Hence, only Raspbian is needed as OS, which is simple and clear.

Raspbian:

Raspbian is the previous name of Raspberry Pi OS. The system is dedicated for driving Raspberry Pi. As a distribution of Linux, the applications compatible with Linux can all be driven on Raspbian. To access MySql database and Dashboard on Raspbian, Node-RED should be installed primarily. Then dashboard and database can be installed onto Node-RED.

In the part on the earth of this system, Raspbian plays a role of central controller. Because of good compatibility of Raspbian (Linux), Node-Red, Local Server, and Antennas of the Deep Space Network can all be connected via corresponding firmware.

Raspbian:

Raspbian is the previous name of Raspberry Pi OS. The system is dedicated for driving Raspberry Pi. As a distribution of Linux, the applications compatible with Linux can all be driven on Raspbian. To access MySql database and Dashboard on Raspbian, Node-RED should be installed primarily. Then dashboard and database can be installed onto Node-RED.

In the part on the earth of this system, Raspbian plays a role of central controller. Because of good compatibility of Raspbian (Linux), Node-Red, Local Server, and Antennas of the Deep Space Network can all be connected via corresponding firmware.

4.2.2. Middleware

Node-RED:

The paper by Rajalakshmi & Shahnasser (2017) provides a concise introduction of Node-RED: it is developed by IBM to combine hardware, API, and other online services in a smart way. Node-RED is a flow editor based on the browser, which can be installed in a Linux platform.

Node-RED has 2 components, which are Nodes and flows are the. Nodes are written in node.js, which can be installed easily from Node-RED library. To create flow diagrams, various Nodes need to be integrated. The Nodes are configured and stored using JSON. Because Node-RED supports smart combination between OS and online services and its compatibility on Linux, the part on the earth of this project links to database and visualization services via Node-RED.

4.2.3. Database

MySQL:

MySQL is an open-source relational database management system. The data saved in the database may related to each other, which is a common character of climate data. In addition, MySQL can support Linux system, which perfectly match the OS of this project.

The data collected from Mars is divided according to different types of parameters (e.g., temperature, humidity, illumination, air pressure, etc.). On the other hand, the system leverages a program to predict future data by using historical data. When showing on dashboard or control panel, the data is classified in terms of data types and period (i.e., historical, current, and predicted).

4.2.4. Visualization Service

Dashboard:

A Dashboard function can be installed on Node-RED. This software allows users to design and implements a control panel for various objectives according to different requirements.

As discussed in database, the climate data is classified based on time, which can be searched on dashboard with parameters of corresponding limiting condition (i.e., period, location, and certain climate type(s)).

4.2.5. Firmware

cc2650:

Since the Raspbian system on Mars is equipped with cc2650 for interconnection with sensors, the firmware for cc2650 launchpad and cc2650 sensortag should be downloaded into the Raspbian on Mars site in advance. The firmware includes 2 files named as “slip-radio.bin” for launchpad and “cc26xx-web-demo.bin” for sensortag respectively. The download procedures can be finalized on Contiki OS, although Contiki OS is not needed for system working.

4.3. Hardware components

Hardware components are the carriers, interconnectors, and executers of software components, which are mainly devices and antennas in this project. In the telecommunication and data processing parts, the hardware components include: Antennas of the Deep Space Network, Gateway, Motes & Sinks, Power Source, and Server. Besides, there are some other components for initial environment building i.e. carrier and lander.

4.3.1. Antennas of the Deep Space Network

Ka-band Deployable Mesh Reflector Antenna:

To implement telecommunication including downlink and uplink between the ground and the Deep Space network, a suitable antenna is necessary. Transmit power and antenna gain are the two main challenges to meet the communication system requirements of Deep Space satellites.
Ka-band Deployable Mesh Reflector Antenna is compatible with the Deep Space Network with both uplink (34.2-34.7 GHz) and downlink (31.8-32.3 GHz) capabilities introduced by Chahat et al. (2017). The pictures of real products are shown as following **Fig. 2**:

(a)

(b) (c)

Figure 2. *(a) Antenna. (b) Stowed antenna assembly fitting in 1.5U (10×10×5cm^3). (c) Deployed mesh reflector antenna.*

4.3.2. Gateway

Raspberry Pi:

Raspberry Pi is a small (credit card size) and affordable computer for programming learning. Because of its modularity and open design, it is suitable for climate monitoring on Mars. A real product picture of Raspberry Pi is shown as **Fig. 3**:

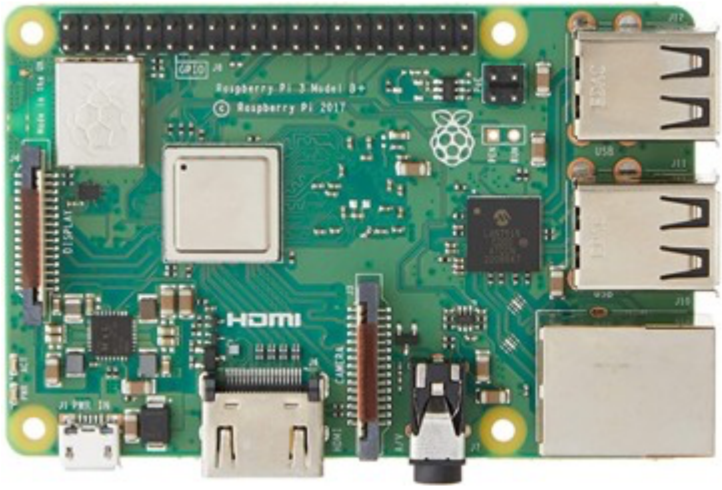


Figure 3. *Real product picture of Raspberry Pi.*

In this project, there are 2 modules connected on Raspberry Pi. The 1st module is for sensors to collect climate data. The 2nd module is for cc2650 to simplify layout by using short-range wireless transmit function.

Different types of sensors including temperature, humidity, and illumination sensors all mount on Raspberry Pi which is the gateway of sensors. The data from different sensors are classified according to data types. All the data is sent to Raspberry Pi. Then Raspberry Pi upload classified data to connected Raspbian.

For the aspect of cc2650, according to the high-level architecture diagram, Raspberry Pi bridges cc2650 and Raspbian on Mars site as a gateway. After the installation of firmware for sensortag and launchpad of cc2650, then make sure 6LoWPAN Setup is finalized and running.

4.3.3. Motes and Sinks

cc2650:

The product of cc2650 is utilized to quickly and wirelessly transmit sensor data to mobile devices. Therefore, it supports both Bluetooth and Zigbee functions. On the side of mobile OS, both iOS and Android systems are compatible with cc2650. Real product pictures of the mote of cc2650 Sensortag and the sink of cc2650 launchpad are shown as following **Fig. 4** and **Fig.5** respectively:

Figure 4. *Real product picture of cc2650 Sensortag.*

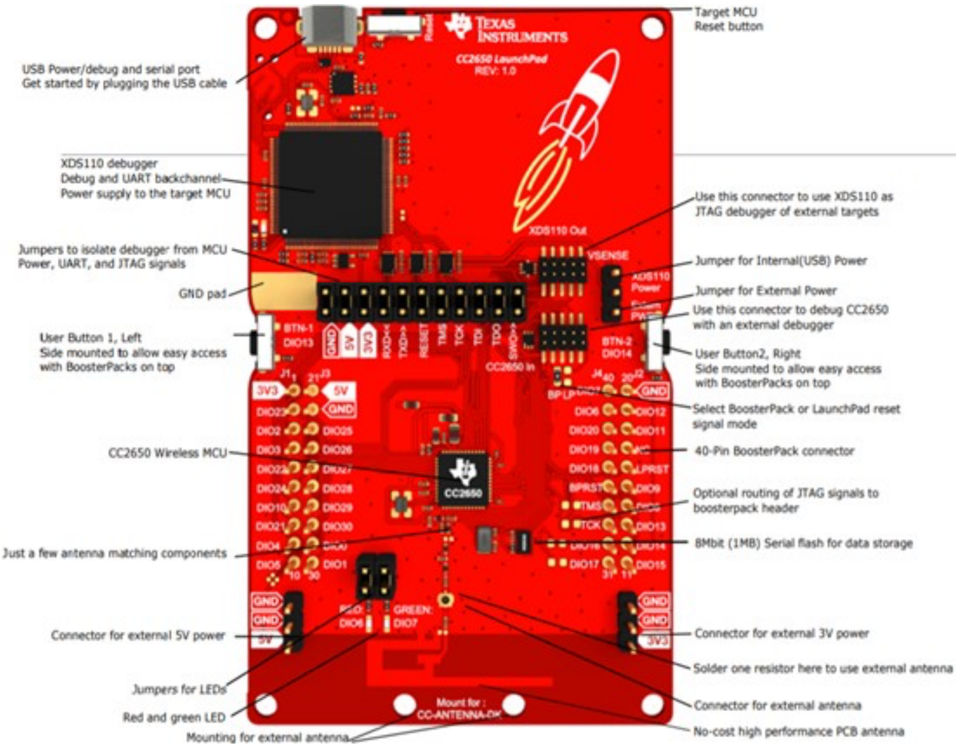


Figure 5. *Real product picture of cc2650 Sensortag.*

Based on its nature of short-range wireless transmission, cc2650 can be utilized to replace cables in short range between devices, which effectively simplifies layout. Besides, it can help to satisfy the requirement in future development such as providing mobile monitoring services for human visitors.

4.3.4. Power Source

Solar battery:

For the part in the space, solar batteries are the original power source. The generated solar power is converted to rated standard (depends on the specifications of devices) by corresponding converters.

220V AC:

For conventional PCs, 220V AC is rated standard power source.

USB power source:

For components include Raspberry Pi, cc2650 Sensortag, cc2650 Launchpad, and different types of sensors, the power source is from USB.

4.3.5. Server

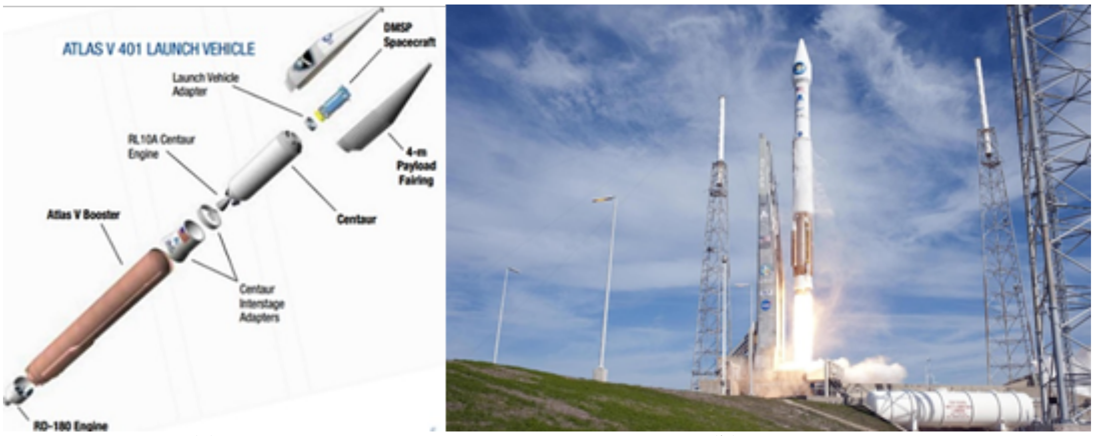
Local Server:

Comparing with other kinds of server, local server is easy to configure and manage. In addition, the security level of local server is also higher since the network configuration depends on local network rather than the network outside, which may have potential security risks. With the consideration of the possibilities of modification and reconfiguration, local server is the best choice.

4.3.6. Other Devices

The hardware components that are not related to communication system include 2 parts: 1) Atlas V-401 rocket is the carrier to launch the part in the space from the earth to Mars, 2) NASA's InSight robot plays the role of lander and explorer once and after the landing on Mars. Both components are mature technologies. Figure. 6 for Atlas V-401 and Figure. 7 for InSight are shown following:

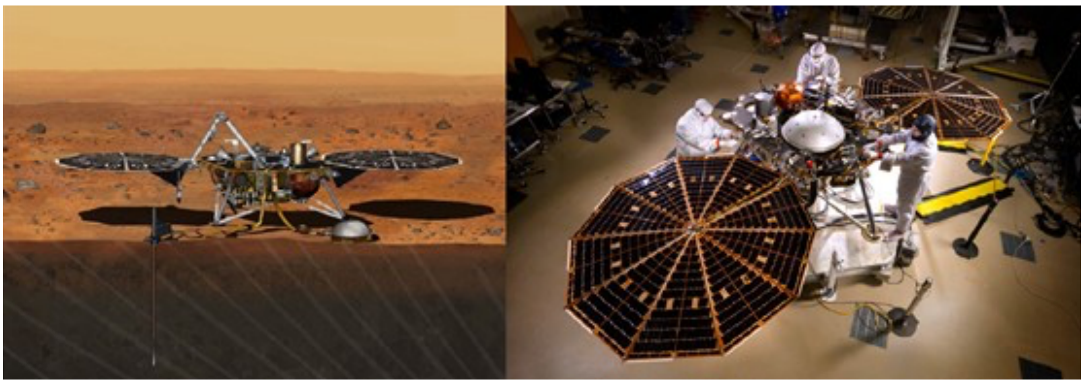
Atlas V-401:



(a) (b)

Figure 6. *(a) Anatomy diagram for Atlas V-401. (b) Real product picture of Atlas V-401.*

NASA's InSight:



(a) (b)

Figure 7. *(a) Solar-powered InSight Lander with its instruments deployed. (b) The InSight Lander during its first landed solar array deployment test at Lockheed Martin in Denver, Colorado (Lisano & Grover, 2019).*

5 Project Plan

5.1 Plan Overview

Below is a project scope statement for our project proposal. This may change during the course of the project but is highly inadvisable in order not to invite any scope creep into the project.

Title	Mars Weather Application
Project Objective	To develop a Mars climate monitoring software system using data from a robotic lander called InSight .
High Level Requirements (Deliverables)	• Obtain data from InSight Mars Lander various weather sensors via NASA API. • Create and populate a database with InSight Mars weather data. Display InSight current and historical Mars weather data in meaningful way.
Technical Requirements	• The application must use InSight: Mars Weather Service API. • The application needs an SQL database to store Mars weather data. • The application needs to display current and historical data. Application can be a web or mobile based application.
In Scope	• A web or mobile application that can present NASA's InSight Mars lander continuous weather measurements (temperature, wind, pressure).

Out of Scope (Limits/Exclusions)	• The team is responsible for adhering to all API terms and conditions of the various API and software terms of use and all licensing agreements.
Assumptions	• No further costs will be incurred for the duration of the project. • Insight Mars lander power source will not be depleted and will be able to charge via InSight's solar panels. • Insight's sensors will be fully functional and available without malfunction. • Schedule will be adhered to including all tasks their duration and required dependencies.
Constraints	• Time constraints, project has to be completed by 17th May 2020 approximately 12 weeks.) • Database integration dependence, most tasks require the database to be populated with Mars weather data before commencing. • Delay in data from InSight sometimes several days later. • Delayed data values from InSight will require recalculation. • Problems with InSight sensors will result in gaps in data. • NASA may turn off various functionality on InSight for many issues such as to save power,

	adverse weather conditions, to preserve the data and physical integrity of InSight or any other reason deemed suitable by NASA. • Project is to adhere to all NASA API licensing agreements, any additional software licensing agreements, and the Bylaws and standards.
Cost Estimate	Hardware costs occurred during development and deployment of InSight (Drier, 2020) include • \$596.3M Spacecraft Development • \$163.4M Launch Vehicle • \$26.95M Estimated ongoing mission costs per annum Software costs of the project currently include • \$120 Web server hosting costs per annum Human resource costs (estimated) required for the project include • NASA staff to maintain InSight: Mars Weather Service API. • NASA staff to monitor InSight. staff to provide consultation and advice.
Milestones	The key milestones are as follows: 1. Outline project requirements. 2. Complete and finalise design. 3. Create and populate an SQL lite database. 4. Obtain data from InSight Mars lander. 5. Create web / mobile application. 6. Develop a neural network to predict Mars weather (depending on schedule). 7. Deploy Application.

Table 1: Project Scope

5.2 Work Breakdown Diagram

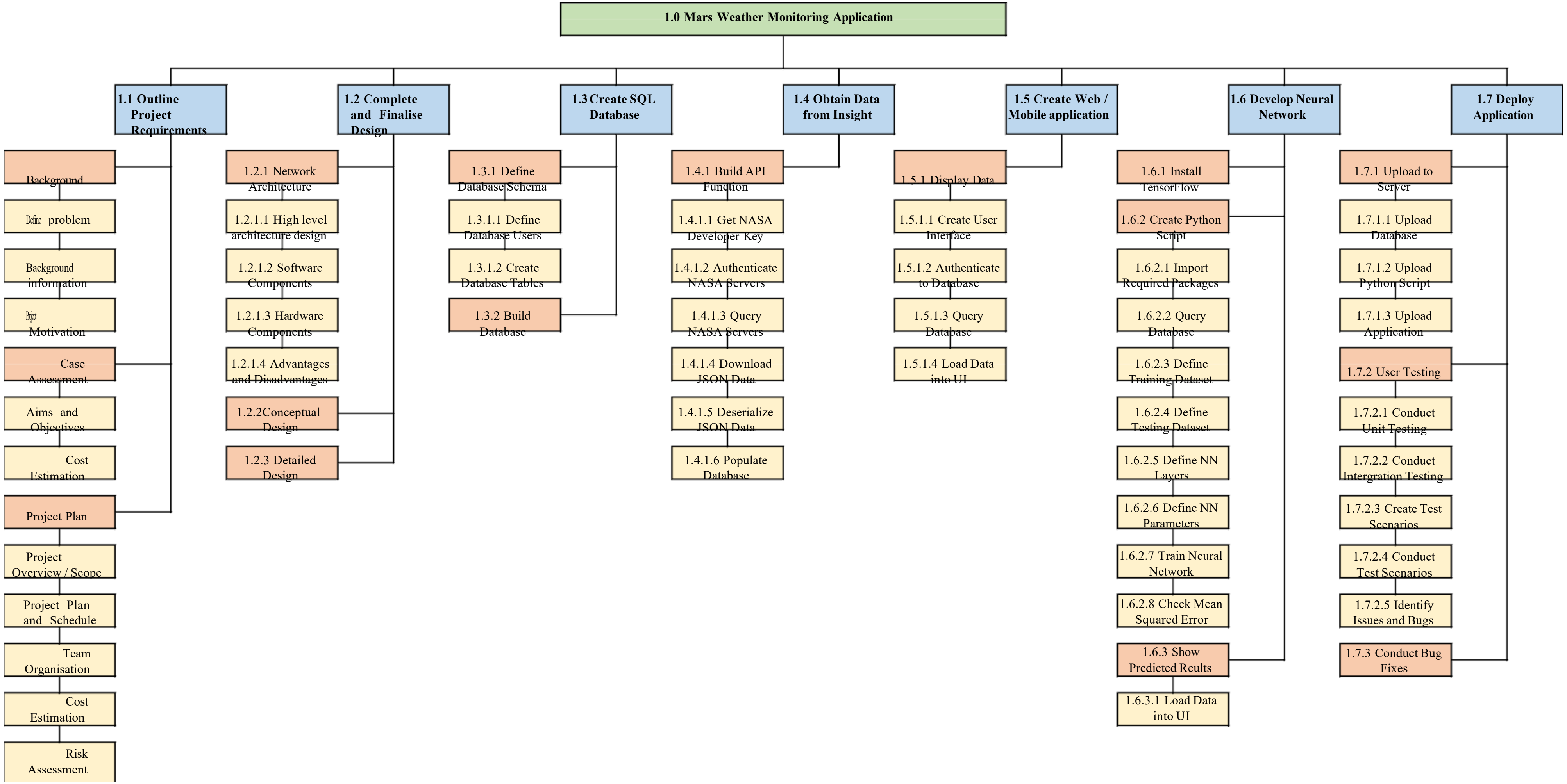


Figure 8. *Work Breakdown Diagram*

5.3 Work Breakdown Hierarchy

WBS Code	WBS Name	Duration	Predecessor
1	**Mars Weather Monitoring Application**	**75**	
1.1	**Outline Project Requirements**	**17**	
1.1.1	**Background**	**5**	
1.1.1.1	Define Problem	1	
1.1.1.2	Background Information	3	1.1.1.1
1.1.1.3	Project Motivation	1	1.1.1.2
1.1.2	**Case Assessment**	**5**	**1.1.1**
1.1.2.1	Aims and Objectives	3	1.1.1
1.1.2.2	Cost Estimation	2	1.1.2.1
1.1.3	**Project Plan**	**7**	**1.1.2**
1.1.3.1	Project Overview / Scope	1	1.1.2
1.1.3.2	Project Plan and Schedule	2	1.1.3.1
1.1.3.3	Team Organisation	1	1.1.3.2
1.1.3.4	Cost Estimation	1	1.1.3.3
1.1.3.5	Risk Estimation	2	1.1.3.4
1.2	**Complete and Finalise Design**	**12**	**1.1**
1.2.1	**Network Architecture**	**8**	**1.1**
1.2.1.1	High Level Architecture Design	3	1.1
1.2.1.2	Software Components	2	1.2.1.1
1.2.1.3	Hardware Components	2	1.2.1.2
1.2.1.4	Advantages and Disadvantages	1	1.2.1.3
1.2.2	**Conceptual Design**	**2**	**1.2.1**
1.2.3	**Detailed Design**	**2**	**1.2.2**
1.3	**Create SQL Database**	**4**	**1.2**
1.3.1	**Define Database Schema**	**3**	**1.2**
1.3.1.1	Define Database Users	1	1.2
1.3.1.2	Define Database Tables	2	1.3.1.1
1.3.2	**Build Database**	**1**	**1.3.1**
1.4	**Obtain Data from InSight**	**7**	**1.3**
1.4.1	**Build API Function**	**7**	**1.3**
1.4.1.1	Get NASA Developer Key	0.5	1.3
1.4.1.2	Authenticate NASA Servers	0.5	1.4.1.1
1.4.1.3	Query NASA Servers	1	1.4.1.2
1.4.1.4	Download JSON Data	1	1.4.1.3
1.4.1.5	Deserialize JSON Data	2	1.4.1.4

1.4.1.6	Populate Database	2	1.4.1.5
1.5	**Create Web / Mobile Application**	**7**	**1.4**
1.5.1	**Display Data**	**7**	**1.4**
1.5.1.1	Create User Interface	3	1.4
1.5.1.2	Authenticate to Database	1	1.5.1.1
1.5.1.3	Query Database	1	1.5.1.2
1.5.1.4	Load Data into UI	2	1.5.1.3
1.6	**Develop Neural Network**	**12**	**1.5**
1.6.1	**Install TensorFlow**	**1**	**1.5**
1.6.2	**Create Python Script**	**9**	**1.6.1**
1.6.2.1	Import Required Packages	1	1.6.1
1.6.2.2	Query Database	1	1.6.2.1
1.6.2.3	Define Training Dataset	1	1.6.2.2
1.6.2.4	Define Testing Dataset	1	1.6.2.3
1.6.2.5	Define Neural Network Layers	1	1.6.2.4
1.6.2.6	Define Neural Network Parameters	1	1.6.2.5
1.6.2.7	Train Neural Network	2	1.6.2.6
1.6.2.8	Check Mean Squared Error	1	1.6.2.7
1.6.3	**Show Predicted Results**	**2**	**1.6.2**
1.6.3.1	Load Data into UI	2	1.6.2
1.7	**Deploy Application**	**16**	**1.5**
1.7.1	**Upload to Server**	**3**	**1.5**
1.7.1.1	Upload Database	1	1.5
1.7.1.2	Upload Python Script	1	1.7.1.1
1.7.1.3	Upload Application	1	1.7.1.2
1.7.2	**User Testing**	**6**	**1.7.1**
1.7.2.1	Conduct Unit Testing	1	1.7.1
1.7.2.2	Conduct Integration Testing	1	1.7.2.1
1.7.2.3	Create Test Scenarios	1	1.7.2.2
1.7.2.4	Conduct Test Scenarios	1	1.7.2.3
1.7.2.5	Identify Issues and Bugs	2	1.7.2.4
1.7.3	**Conduct Bug Fixes**	**7**	**1.7.2**

Table 2: Work Breakdown Hierarchy

Total Days to complete **75**

5.4 Project Plan and Schedule

WBS Code	Task	Duration	Start Day	End Day
1	**Mars Weather Monitoring Application**	**75**	01-Mar-21	15-May-21
1.1	**Outline Project Requirements**	**17**	01-Mar-21	18-Mar-21
1.1.1	**Background**	**5**	01-Mar-21	06-Mar-21
1.1.1.1	Define Problem	1	01-Mar-21	02-Mar-21
1.1.1.2	Background Information	3	02-Mar-21	05-Mar-21
1.1.1.3	Project Motivation	1	05-Mar-21	06-Mar-21
1.1.2	**Case Assessment**	**5**	06-Mar-21	11-Mar-21
1.1.2.1	Aims and Objectives	3	06-Mar-21	09-Mar-21
1.1.2.2	Cost Estimation	2	09-Mar-21	11-Mar-21
1.1.3	**Project Plan**	**7**	11-Mar-21	18-Mar-21
1.1.3.1	Project Overview / Scope	1	11-Mar-21	12-Mar-21
1.1.3.2	Project Plan and Schedule	2	12-Mar-21	14-Mar-21
1.1.3.3	Team Organisation	1	14-Mar-21	15-Mar-21
1.1.3.4	Cost Estimation	1	15-Mar-21	16-Mar-21
1.1.3.5	Risk Estimation	2	16-Mar-21	18-Mar-21
1.2	**Complete and Finalise Design**	**12**	18-Mar-21	30-Mar-21
1.2.1	**Network Architecture**	**8**	18-Mar-21	26-Mar-21
1.2.1.1	High Level Architecture Design	3	18-Mar-21	21-Mar-21
1.2.1.2	Software Components	2	21-Mar-21	23-Mar-21
1.2.1.3	Hardware Components	2	23-Mar-21	25-Mar-21
1.2.1.4	Advantages and Disadvantages	1	25-Mar-21	26-Mar-21
1.2.2	**Conceptual Design**	**2**	26-Mar-21	28-Mar-21
1.2.3	**Detailed Design**	**2**	28-Mar-21	30-Mar-21
1.3	**Create SQL Database**	**4**	30-Mar-21	03-Apr-21
1.3.1	**Define Database Schema**	**3**	30-Mar-21	02-Apr-21
1.3.1.1	Define Database Users	1	30-Mar-21	31-Mar-21
1.3.1.2	Define Database Tables	2	31-Mar-21	02-Apr-21
1.3.2	**Build Database**	**1**	02-Apr-21	03-Apr-21
1.4	**Obtain Data from Insight**	**7**	03-Apr-21	10-Apr-21
1.4.1	**Build API Function**	**7**	03-Apr-21	10-Apr-21
1.4.1.1	Get NASA Developer Key	0.5	03-Apr-21	03-Apr-21
1.4.1.2	Authenticate NASA Servers	0.5	04-Apr-21	04-Apr-21
1.4.1.3	Query NASA Servers	1	04-Apr-21	05-Apr-21
1.4.1.4	Download JSON Data	1	05-Apr-21	06-Apr-21
1.4.1.5	Deserialize JSON Data	2	06-Apr-21	08-Apr-21
1.4.1.6	Populate Database	2	08-Apr-21	10-Apr-21
1.5	**Create Web / Mobile Application**	**7**	10-Apr-21	17-Apr-21
1.5.1	**Display Data**	**7**	10-Apr-21	17-Apr-21
1.5.1.1	Create User Interface	3	10-Apr-21	13-Apr-21
1.5.1.2	Authenticate to Database	1	13-Apr-21	14-Apr-21
1.5.1.3	Query Database	1	14-Apr-21	15-Apr-21
1.5.1.4	Load Data into UI	2	15-Apr-21	17-Apr-21
1.6	**Develop Neural Network**	**12**	17-Apr-21	29-Apr-21
1.6.1	**Install TensorFlow**	**1**	17-Apr-21	18-Apr-21
1.6.2	**Create Python Script**	**9**	18-Apr-21	27-Apr-21
1.6.2.1	Import Required Packages	1	18-Apr-21	19-Apr-21
1.6.2.2	Query Database	1	19-Apr-21	20-Apr-21
1.6.2.3	Define Training Dataset	1	20-Apr-21	21-Apr-21
1.6.2.4	Define Testing Dataset	1	21-Apr-21	22-Apr-21
1.6.2.5	Define Neural Network Layers	1	22-Apr-21	23-Apr-21
1.6.2.6	Define Neural Network Parameters	1	23-Apr-21	24-Apr-21
1.6.2.7	Train Neural Network	2	24-Apr-21	26-Apr-21
1.6.2.8	Check Mean Squared Error	1	26-Apr-21	27-Apr-21
1.6.3	**Show Predicted Results**	**2**	27-Apr-21	29-Apr-21
1.6.3.1	Load Data into UI	2	27-Apr-21	29-Apr-21
1.7	**Deploy Application**	**16**	29-Apr-21	15-May-21
1.7.1	**Upload to Server**	**3**	29-Apr-21	02-May-21
1.7.1.1	Upload Database	1	29-Apr-21	30-Apr-21
1.7.1.2	Upload Python Script	1	30-Apr-21	01-May-21
1.7.1.3	Upload Application	1	01-May-21	02-May-21
1.7.2	**User Testing**	**6**	02-May-21	08-May-21
1.7.2.1	Conduct Unit Testing	1	02-May-21	03-May-21
1.7.2.2	Conduct Intergration Testing	1	03-May-21	04-May-21
1.7.2.3	Create Test Scenarios	1	04-May-21	05-May-21
1.7.2.4	Conduct Test Scenarios	1	05-May-21	06-May-21
1.7.2.5	Identify Issues and Bugs	2	06-May-21	08-May-21
1.7.3	**Conduct Bug Fixes**	**7**	08-May-21	15-May-21
	Total Days	**75**		

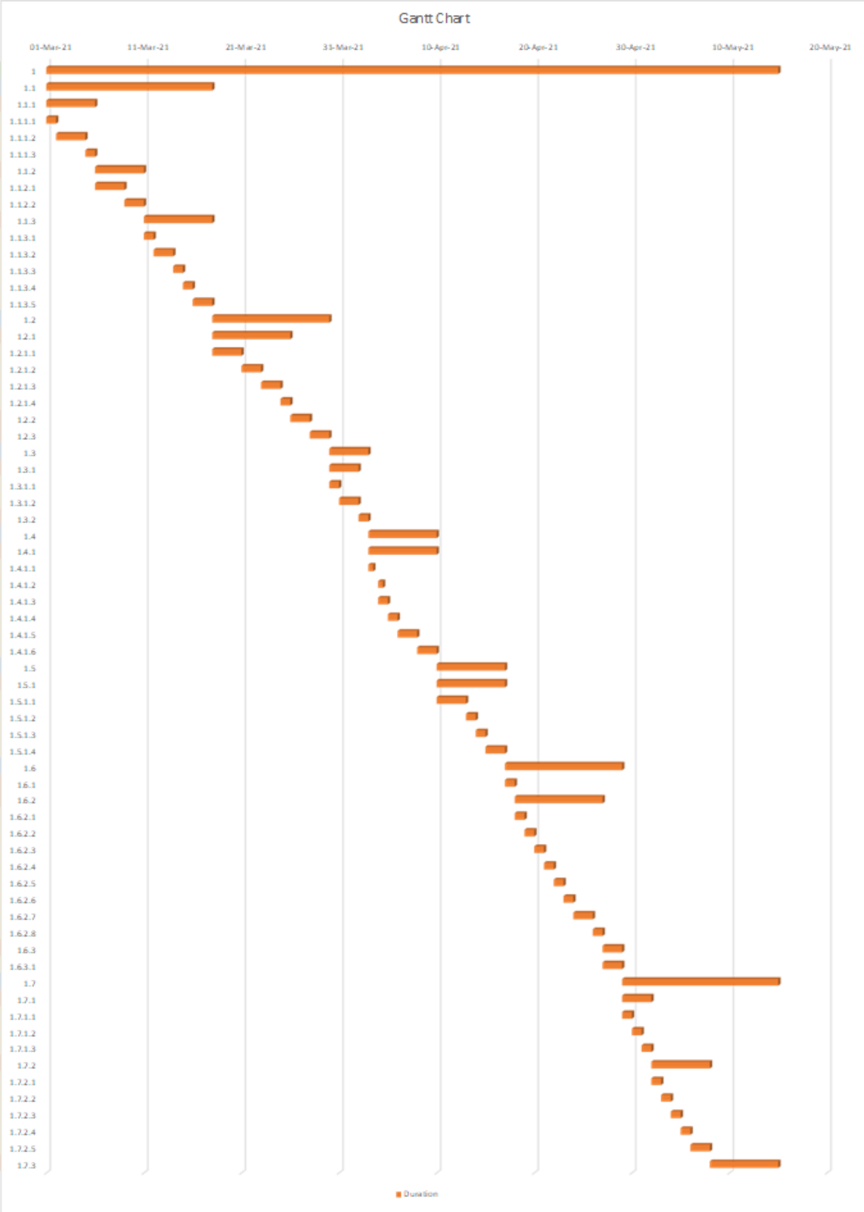


Figure 9. *Gantt Chart*

5.5 Team Organisation

We have decided to use a RACI matrix to organise and allocate each team members role and responsibilities.

RACI Chart				
Activity	**Person**			
	Person A	**Person B**	**Person C**	**Person D**
1.1 Outline Project Requirements	A,C	A,R	A,C	R,A
1.2 Complete and Finalise Design	R,A	A,C	A,C	A,C
1.3 Create SQL Database	A,C	A,C	R,A	A,C
1.4 Obtain Data from InSight	A,C	R,A	A,C	A,C
1.5 Create Web / Mobile Application	R,A	A,C	A,C	A,C
1.6 Develop Neural Network	TBC	TBC	TBC	TBC
1.7 Deploy Application	TBC	TBC	TBC	TBC
Assignment Presentation	TBC	TBC	TBC	TBC

Legend	
To be confirmed	TBC
Responsible	R
Accountable	A
Consulted	C
Informed	I

Table 3: RACI Matrix

5.6 Cost Estimation

Cost Estimates	
Item	**Cost**
Development	
InSight Mars Lander	$596,300,000.00
Deployment	
Launch Atlas Rocket	$163,400,000.00
Operations	
Ongoing Mission Costs (over two years)	$52,400,000.00
NASA Deep Space Network Crew	$1,000,000.00
NASA Software Team	$500,000.00
Software	
Web Server Hosting per year	$120.00
Operating System (Linux Distribution)	$0.00
Database (SQLite Free version)	$0.00
Node-RED (Open Source)	$0.00
Additonal Human Resources	
Staff	$150,000.00
Additional Investment	
CNES France	$90,000,000.00
DLR Germany	$90,000,000.00
Total	**$993,750,120.00**

Table 4: Cost Estimate (Glaze, 2020)

**Human Resources costs have been estimated and may differ significantly.*

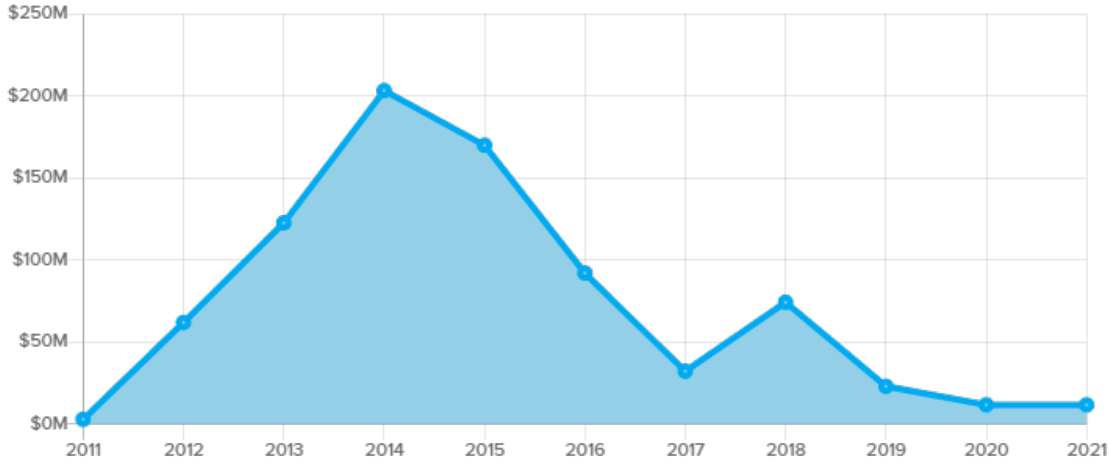


Figure 10. *NASA InSight cost per year (Drier, 2020).*

5.7 Risk Assessment

Risk ID	Task / Activity	Risk Description	Risk Score	Owner	Control measures	Response plan
Risk Register						
1	Retrieving sensor data	Gaps in data	H	NASA	> Troubleshoot sensors	> Run diagnostic tests > Estimate data values > Predict data values > Accept missing values
2	Mars InSight Hardware	Solar panels covered in dust	M	NASA	> Mars wind or dust storm to blow dust away	> Wait for correct weather conditions
3	Mars InSight Hardware	Hardware component failure	M	NASA	> Troubleshoot hardware	> Run diagnostic tests > Restart hardware > Disable power to hardware component
4	Retrieving sensor data	Delay in retrieving data	L	NASA	> Persist querying InSight's sensors for data	> Delayed data values from InSight will require recalculation.
5	Retrieving sensor data	Low power on sensors	L	NASA	> NASA may disable specific sensors and features of InSight to save power	> Use data from available sensors
6	Developing Application	Require additional software	L	Team	> Use free software	> Source software > Find open source substitute > Use contigency funds
7	Developing Application	Require additional hardware	L	Team	> Have spare devices in case one fails	> Source hardware > Use suitable hardware alternative > Use contigency funds
8	Developing Application	Require cloud services	L	Team	> Ask a cloud vendor for student credit	> Source hardware from University > Use contigency funds
9	Developing Application	Insufficient data to train neural network	L	Team	> Reduce training data	> Calculate estimated values > Edit neural network parameters
10	Developing Application	Processing power to run neural network	L	Team	> Reduce parameters of the neural network	> Reduce neural network layers > Reduce amount of training data > Reduce amount of testing data

Risk Score	Code
Negligible	N
Low	L
Medium	M
High	H
Extreme	E

Table 5: Risk Register

6 Future Improvement

6.3 Limitations of the project

This section has analyzed that our application has some limitations as IoT operating systems used in our system. We have also analyzed those usual necessities for programming running on Internet of things systems. Specifically, we get from this inspection of our application that none of the modern working frameworks of IoT operating system. Windows is competent to satisfy the various essentials of IoT frameworks, incorporating assorted tools limitations, just as a different organization working place, self-sufficiency and constant imperatives. A few activities pointed toward adjusting existing Operating systems of application for the Internet of things constraints used in our application architecture. The key highlights like finest energy proficiency and provide real-time applications in the system can't be productively carried out as additional items to the previously used frameworks. But, this is not enough to understand that after inspection of our system because such key highlights of operating system used not possess all aspects of the framework.

Lack of energy efficiency: The variety of systems used in this application architecture requirements that need to be fulfilled by Contiki OS and Debian software is not sufficient for this application. Energy efficiency is a factor that can impact the system that is purchasing such high-level hardware and software components for the implementation of climatic monitoring and prediction technologies in the agricultural sectors.

Connectivity: It is one of the major challenges we may face during the implementation of technologies to reduce the problem in the agriculture sector. Connecting with many devices while working will be a major challenge in IoT. This will confront the architecture of our system and current communication model and the software and hardware used in primary technologies.

Lack of Safety and security: In this system, we have faced a lack of practices that can make our system insecure because now IOT has turned into serious security concerns for the applications used in communications protocols used in the projects. This shows the lack of guidance while choosing the security measures for the proposed software used in the application.

Longevity and compatibility: According to the recent inspection, the Internet of things is substantial in various ways, with a wide range of advancements contending to turn into the norm. So, these issues of IoT have causes challenges involve the distribution of additional tools and techniques while interfacing the IoT systems. Other similarity issues originate from the cloud services of the system, the absence of machine-to-machine communications between the system, and varieties of working frameworks among Internet of things applications.

6.4 Possible improvements

This section recommends that there is a need for enhancement in various agricultural sectors where we implement climate monitoring and prediction technologies. Consequently, we need to reduce the problems like memory management, energy efficiency, and communications faced during the implementation of climate monitor technologies. Some IoT applications we can use in our system to secure and improve the functionalities of our system.

RFID-based system: Radio-frequency identification system is a system that we can use to collect the data by automatically identifying the objects, which will help secure the data. It is a technology used as a non-contact radio wave mobile device to send and receive the data. With these advancements in the technological field of climate prediction and monitoring, we can enhance our system security.

Smart devices of IoT: These devices of IoT are used to connect the networking of the system for better communication and connection. By using smart devices of the Internet of things, we can improve our networks and resolve the connectivity issue. Secure IoT systems are reliable, efficient and convenient measures to advance the agricultural sector.

Smart grids: - in this, we can use different sensors to improve our system's reliability and efficiency. Smart grid is an advancement of sensors while implementing new applications. We need to approve the behavior of the system. So, these smart grids are used to automatically control the details about the energy consumption and behavior of the system.

7 Conclusion

In conclusion, the project plan develops a Node-RED dashboard monitoring the climate conditions on Mars. We have illustrated the required hardware components to ensure successful delivery of our project. We plan to achieve the software aspect by utilizing NASA Mars InSight API, MySql database and Node-RED for the dashboard.

Understanding and utilising data collected via wireless sensors is our key motivation for this project. NASA's Mars InSight Mission not only aid current space exploration missions but will directly contribute to future technological advancements in the agricultural industry, human's race to survival and forthcoming planetary missions.

8 References

Bhasin, K., Barnes, P., Reinert J., & Golden, B. (2013). Applying model based systems engineering to NASA's space communications networks. *2013 IEEE International Systems Conference (SysCon)*, 325-330. https://doi.org/10.1109/SysCon.2013.6549901.

Chahat, N., Sauder, J., Thomson, M., Rahmat-Samii, Y., & Hodges, R. (2017). Ka-band Deployable Mesh Reflector Antenna compatible with the deep space network. *2017 11th European Conference on Antennas and Propagation (EUCAP)*, 546-548. https://doi.org/10.23919/EuCAP.2017.7928834.

Drier, C. (2020). Cost of Mars InSight. *Planetary.org.* https://www.planetary.org/space-policy/cost-of-mars-insight

Glaze, L. (2020). Explore Solar System and Beyond. Slide 17. *NASA.* https://science.nasa.gov/science-pink/s3fs-public/atoms/files/Glaze-Clarke-Bleacher-LPSC-March202-TAGGED.pdf

Hoffman T. (2018). InSight: Mission to Mars, *2018 IEEE Aerospace Conference*, 1-11, https://doi.org/10.1109/AERO.2018.8396723.

Lisano, M., & Grover, M. (2019). Riders on the Storm: NASA InSight Lander and the 2018 Mars Global Dust Storm, *2019 IEEE Aerospace Conference*, 1-7. https://doi.org/10.1109/AERO.2019.8742008.

Lekić, M., & Gardašević, G. (2018). IoT sensor integration to Node-RED platform. *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, 1-5. https://doi.org/10.1109/INFOTEH.2018.8345544.

Mittal, N., & Cloutier, K. (2020). Developing and Implementing a Process to Verify and Validate the NASA InSight Mission's Instrument Command Products. *2020 IEEE Aerospace Conference*, 1-9. https://doi.org/10.1109/AERO47225.2020.9172307.

NASA (2019). Surface Operations. *NASA.gov*. https://mars.nasa.gov/insight/timeline/surface-operations/.

NASA (2021). NASA InSight's 'Mole' ends its journey on Mars. *NASA.gov*. https://www.nasa.gov/feature/jpl/nasa-insight-s-mole-ends-its-journey-on-mars.

NASA (2021). InSight is meeting the challenge of winter on dusty Mars. *NASA.gov*. https://mars.nasa.gov/news/8858/insight-is-meeting-the-challenge-of-winter-on-dusty-mars/?site=insight.

Rajalakshmi, A., & Shahnasser, H. (2017). Internet of Things using Node-Red and alexa. *2017 17th International Symposium on Communications and Information Technologies (ISCIT)*, 1–4. https://doi.org/10.1109/ISCIT.2017.8261194.