|  |
| --- |
| Name: J.A. Mujeeb |
| Student Reference Number: 10707284 |



|  |  |  |
| --- | --- | --- |
| Module Code: CNET343SL | Module Name: Distributed Systems | |
| Coursework Title: Weather Reporting System Proposal | | |
| Deadline Date: 03/12/2021 | | Member of staff responsible for coursework: Mr. Pramudya Thilakaratne. |
| Programme: BSc (Hons) Plymouth Software Engineering | | |
| Please note that University Academic Regulations are available under Rules and Regulations on the University website [www.plymouth.ac.uk/studenthandbook](http://www.plymouth.ac.uk/studenthandbook). | | |
| Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.  J.A. Mujeeb – 10707284  G.M.D.D. Ratnayake – 10707351  S.O. Perera – 10707315  N. S. De Alwis – 10707160  M. D. A. Medhavi – 10707278  P. P. L. Dilhani – 10709402  ***We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.***  Signed on behalf of the group: J.A. Mujeeb | | |
| Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***    Signed: | | |
| Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.  I \*have used/not used translation software.  If used, please state name of software………………………………………………………………… | | |
| **Overall mark \_\_\_\_\_% Assessors Initials \_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_** | | |

\*Please delete as appropriateSci/ps/d:/students/cwkfrontcover/2013/14

Contents

[2. Overview/Introduction 4](#_Toc89532896)

[3. Objectives 4](#_Toc89532897)

[3.1 Simplicity 5](#_Toc89532898)

[3.2 Uploading pictures - 5](#_Toc89532899)

[3.3 API Implementation - 5](#_Toc89532900)

[3.4 Blog Posts - 5](#_Toc89532901)

[3.5 Alerts and Notifications - 5](#_Toc89532902)

[3.6 Identification of End Users - 5](#_Toc89532903)

[3.7 Future Implementations 6](#_Toc89532904)

[3.7.1 Chat Bot Feature - 6](#_Toc89532905)

[4. Target Users 6](#_Toc89532906)

[4.1 General Public – 6](#_Toc89532907)

[4.2 Agricultural Sector Employees - 6](#_Toc89532908)

[4.3 Transport Sector Employees – 6](#_Toc89532909)

[4.4 Construction Sector Employees – 7](#_Toc89532910)

[5. . Application Features and Description 7](#_Toc89532911)

[5.1 Speed - 7](#_Toc89532912)

[5.2 Resource sharing. 7](#_Toc89532913)

[5.3 Openness: 7](#_Toc89532914)

[5.4 Concurrency: 7](#_Toc89532915)

[5.5 Scalability: 7](#_Toc89532916)

[5.6 Fault Tolerance: 8](#_Toc89532917)

[5.7 Transparency: 8](#_Toc89532918)

[5.8 User mobility - 8](#_Toc89532919)

[5.9 Performance - 8](#_Toc89532920)

[6. Application Technologies with proper Justifications 9](#_Toc89532921)

[6.1 Heroku 9](#_Toc89532922)

[6.2 Back-end 9](#_Toc89532923)

[6.3 Middleware 9](#_Toc89532924)

[6.3.1 What is middleware? 9](#_Toc89532925)

[6.4 Load Balancing 10](#_Toc89532926)

[7. Time Frame 10](#_Toc89532927)

[8. Project Plan 11](#_Toc89532928)

Weather Reporting System

# Overview/Introduction

The use of weather reporting in day-to-day life is very important. Its utilization could greatly influence the outcome of a scenario. It could be something as simple as deciding whether you should take your umbrella on your way out, or even as major as handling cultural operations, agriculture and farming or livestock protection implementations.

With Sri Lanka’s weather rapidly varying, weather can transition from largely homogeneous temperatures to torrential rain in a heartbeat. With the adaptation of unexpectedly frequent changes, it is crucial to make sure to be prepared.

Sri Lanka is liable to prevailing and predictable effects of climate change. Preceding natural disasters (such as tsunamis, floods, landslides, droughts, and cyclones) can greatly substantiate this. Profoundly weather-sensitive sectors in Sri Lanka include transport, agriculture, construction, energy, and disaster risk management.

It is important to note that even with weather stations, live broadcasts, and television/radio stations existent, not all are aware of the continuously differentiating weather conditions.

The impacts and effects of climate change could invite climate change-induced hazards and disasters. The unawareness of weather reports may affect the lives of many. Thousands of citizens are prone to being affected by the threat of climate change. They may find themselves in life threatening situations. Weather prediction is essential in order to provide citizens with pragmatic information. Furthermore, this also aids in the reduction of weather-related losses, personal safety and health, enhancement of societal benefits and in supporting economic prosperity.

It is abundantly clear that weather forecasting reports are essential to mitigate the effects civilians getting caught in a severe crisis.

# Objectives

Our goal is to present a high-performance system that encompasses paramount customer satisfaction, convenience, imperative security, user-friendliness, and maximum authenticity.

We have planned to specifically cater towards users that may be unfamiliar or inexperienced in the usage of technology.

Furthermore, we want our distributed system to be able to yield a weather forecast at any time of any place with the utmost accuracy and efficiency of the mobile and web application.

Our project will consist of:

1. Web Application
2. Mobile Application

We will be incorporating the following characteristics into our applications:

## Simplicity

We will be considering and implementing HCI principles in both our web and mobile application. Simplicity compliments an easily navigable user interface. A clean and sleek layout adulated with a uniform colour scheme will increase the intrinsic attention of the user. As we plan to particularly cater towards users that may have unfamiliarity with technology, we decided to include simple English in order to ensure that those who interact with this website can easily understand. We will be considering end users of all ages when implementing our web and mobile application.

## Uploading pictures -

Users can upload pictures. This will help in determining the current real time weather in the location of the individual that uploads the image. Anyone that may be near that location or even those who plan to reach that area will be notified in advance. In addition to that, this also strengthens the process of data gathering.

## API Implementation -

API’s (Application Program Interface) play a considerable role in our distributed system project when it comes to gathering data for the Weather Forecasting display. The API poses as a pathway and gives your application software or product the ability to communicate with another software.

## Blog Posts -

Users will have access to Blog Posts within the website. Articles will be published from a variety of sources. Users will be provided with useful information related to weather forecast statistics and updates. This enhances societal benefits as it aids in the protection of life and property, supports economic prosperity and safety. Blogging also anchors many benefits. It is a superb way to drive traffic to the website and also in building credibility.

## Alerts and Notifications -

It is important to incorporate the Alerts/Notifications feature in our application as it plays a role in improving the application performance. It ensures complete visibility of critical information. Furthermore, this also improves customer service, loyalty, and satisfaction. In our case, our users will be notified when the weather is extremely hazardous. This would greatly help in protecting lives, health, and safety. The mode of which the notification is being delivered could vary as this depends on the status, severity and how close that person is to the location of the disaster. We plan on implementing an alert in the form of a vibration. This is greatly similar to a haptic vibration.

## Identification of End Users -

We have considered our end users to be in the age range of 10 – 60. The end users must have an internet connection in order to be able to access this web and mobile application. A simple understanding of English Language will be necessary. Our web and mobile application will have Cross Platform compatibility.

## Future Implementations

### Chat Bot Feature -

User communication is essential as our application aims to deliver smooth engagement. Chatbots have the ability to replicate a human presence. The utilization of a Chatbot would improve our communication strategy in furtherance of delivering a better experience to our users. The 24/7 availability, instant user response, personalization and answer consistency will leverage good customer relationships. In our project, we aim to develop a Chatbot feature in which users can virtually interact and get information in a hassle-free manner. This can be with regards to emergency contact information, natural disaster alerts and other crucial aspects.

# Target Users

## General Public –

Weather forecasting is important when it comes to protecting life and property. It plays a role in providing an early warning for a possible life-threatening hazard. A substantial amount of end users for weather forecasting applications are the public. Many aspects of the general public’s life are influenced by the weather. Hazardous weather conditions such as thunderstorms or lightning strikes can cause power outages or even lead to deaths. Weather forecasting aids citizens in planning their day or week. When citizens get an idea on what type of weather is supposedly expected, they can prepare on how to dress accordingly.

Weather forecasting would help businesses plan with regards to transportation hazards that can result from severe weather. This also relates to flying and driving.

## Agricultural Sector Employees -

With over 30% of Sri Lankan citizens employed in the agricultural sector, this dominant economic force plays a main role in reducing poverty. Contributing 7.4% to the national GDP, it is unquestionable that Agriculture is an immensely important area in the Sri Lankan economy in terms of employment for most of the Sri Lankan workforce.

Weather forecasting enacts a compelling impact on the agriculture industry. This is because weather forecasting knowledge drives an agriculturist’s business decisions, aiding them with efficient planning decisions and cost minimizations. As a result, this would help in maximizing profits.

## Transport Sector Employees –

Transport Sector employees heavily rely on weather forecasting as the weather could affect daily operational decisions. Decisions can revolve around the amount of cargo that a plane or the repercussions of flooding, high winds, and wet surfaces. Delivery reliabilities, considerable competence and transit time can all be affected by adverse impacts from weather conditions. When it comes to the safety, reliability, efficiency, and operations of transport systems, it is crucial that weather forecasting conditions must be taken into consideration.

## Construction Sector Employees –

Heavy rain, thunderstorms Intense or very low temperatures and other hazardous conditions pose a dangerous threat for the safety of Construction Sector workers. It can also reduce their ability to concentrate. Due to this, Construction Sector Employees heavily rely on weather forecasting and find it very useful. It is eminent that they are fully aware of their surroundings in order to work safely. The utilization of a weather forecasting system would help Construction Sector Employees dress accordingly and be informed and prepared.

# . Application Features and Description

## Speed:

**Lagging and extensive loading would result in an unsatisfactory user experience. This could occur when loading images and content. If this occurs, the end user would feel** bored **and** frustrated. **Due to this,** fast loading screens are extremely vital. Our aim is to keep things quick and simple. The relevant set of graphics will be implemented and large tables and databases or extraneous animations will not be fetched.

## Resource sharing:

It is important to co-operate and share materials and information with alternative parties. We plan to have resource sharing as a part of this app. It will be used in furtherance of sharing images of the current real-time weather within your surrounding areas. This will be uploaded so that other end users can also view this information and utilize it to their advantage.

## Openness:

We have decided to avoid implementing sign up or sign in features. The system will be up for public use. This is due to the exuberant number of users all around the world that utilize a weather forecasting application. It would be beyond inconvenient if end users are required to frequently sign in and out in order to check the weather.

## Concurrency:

With Multiple estimations happening at a specified or same time, our app will be multiple running. You will be viewing weather forecasting readings in addition to checking for any images uploaded by other users.

## Scalability:

Scalability is a measurement that can be taken to check for an increment or decrement in our distributed project’s performance.

To check the scalability, we will be executing various methods of testing and adding necessary functionalities. If the time factor is within our advantage, we will do our best to add more functionalities to increase our project performance.

## Fault Tolerance:

To avoid the occurrence of any system malfunctions or faults, we will be looking into Fault Tolerance. Functionality Testing must be conducted in order to assure that the web and mobile application delivers faultless functionality. Our system must have the means to work efficiently and effectively. We will be utilizing load balancing to maintain user requests. In addition to that, we will also have backups in order to avoid the risk of probable data loss (in more than one location). Files will be updated on GitHub religiously.

## Transparency:

We aim to make our web and mobile applications more transparent as this will increase their utility and trustworthiness with the end user. An application is considered to be transparent if the end users are aware of the criteria that plays a role in influencing its output. In turn, this promotes accountability and reduces an impeding risk.

In order to hide the distributed system nature from the end users, we can adapt to access transparency, location transparency, concurrency transparency or fault transparency. We plan to implement some of these transparencies. This will ensure that the end user interacts with the UI properly.

## User mobility -

To guarantee that the web and mobile application works well on both IOS and Android platforms, cross platform testing will be executed. It is crucial that our applications can be used in different locations with different hardware components.

We plan to use flutter as our coding base. This is because it will be compatible for both IOS and Android platforms.

## Performance -

Functionality testing will be carried out to ensure that our web and mobile application upholds peak performance and is professional and competent. End users would want the applications to have a smooth, fast, and responsive interface. If the application starts conflicting from slow loading times, crashing or terrible user interface, this will result in many user complaints. Regression testing and system validation will also be carried out to ensure that the web and mobile application has faultless functionality. Furthermore, a cluster of servers will be created to make sure that the system is backed up into these servers. Even if a fault occurs in one area, it can be eliminated with the help of the rest of the servers and keep the application running continuously.

# Application Technologies with proper Justifications

## Heroku

Heroku is a developer-friendly cloud Platform as a Service (PaaS) that supports many programming languages. Heroku is a server that developers can use to deploy, manage, and scale fashionable apps. Our platform is elegant, flexible, and simple to use, providing developers the best path to obtaining their apps to plug. Heroku is absolutely managed, giving developers the liberty to specialize in their core product while not the distraction of maintaining servers, hardware, or infrastructure. The Heroku expertise provides services, tools, workflows, and polyglot support-all designed to boost developer productivity. (Heroku, n.d.) Pros of using Heroku are allows the developer to focus on code instead of infrastructure, enhance the productivity of cloud app development team, and monitor and enhance performance though rich application monitoring.

## Back-end

We plan to utilize the Firebase real-time database, NGINX, and Node js.

Firebase real-time database is a cloud-based database that allows users to store and sync data in real time. The advantage of implementing the Firebase real-time database is that developers like us do not require the support of backend to build apps. This is because it comes with SDKs for various platforms, including iOS, Android, and Web. It is also optimized for offline use as well. Firebase also has an easy sign-in process.

Node.js is a transcript framework that shines in real-time web applications using push technology over WebSocket. (toptal, n.d.) Node.js associates ASCII text files, cross-platform and back-end JavaScript runtime environment (which runs on the V8 engine and executes JavaScript code outside an online browser). Node.js allows developers to use JavaScript in order to write down command line tools. Furthermore, server-side scripting provides dynamic website content before the page is distributed to the user's web browser.(Wikipedia, n.d.)

NGINX is an open-source software which can be used in media streaming, load balancing, web serving and reverse proxying. It acts as an internet server designed for optimum performance and stability. In addition to its hypertext transfer protocol server capabilities, NGINX can even perform as a proxy server for email (IMAP, POP3, and SMTP) and a reverse proxy and load balancer for hypertext transfer protocol, TCP, and UDP servers**.** (Nginx, n.d.)

## Middleware

### What is middleware?

Middleware can provide services and other components to the application (that the application is unable to provide). It is mostly referred to as a “Software Glue” because of its ability to connect two completely contrasting applications.

When it comes to developing the application, Middleware can help in efficiency and cost-effectiveness. (Anon., 2018).

Middleware handles API management, which we will included in the project. We plan to implement Middleware in our project.

With the utilization of an API, it is feasible to connect entirely different products and services through a common layer.

## Load Balancing

Load balancing performs the following functions:

* load estimation
* load levels comparison,
* performance indices
* system stability,
* amount of information exchanged among nodes,
* job resource requirements estimation,
* job's selection for transfer
* remote nodes selection and more (Alakeel, 2009)

With the utilization of load balancing, we are able to create paths with the aim of trafficking requests that arrive at the same time.

Load balancing enables a certain task to be implemented independently, share information, and move from one computer to another within the proposed distributed system.

Apart from load balancing, load sharing is an entity that must be taken under consideration.

By implementing load balancing, we can achieve:

1. Highly efficient system performance.

2. Treat all tasks in the system equally regardless of their origin.

3. Can overcome fault tolerance (performance wise).

4. Improve in terms of Adaptability.

# Time Frame

**Gantt chart:**

Timeline

Description automatically generated

Figure 1: Time Plan Frame

Chart, waterfall chart

Description automatically generated

Figure 2: Gantt Chart

# Project Plan

|  |  |
| --- | --- |
| Planning | Description |
| Frontend Designing | With the consideration of HCI principles, we will be implementing simple English, attractive colours, eye-catching imagery and other aspects to our web and mobile application. |
| Research | Extensive research will be carried out with regards to similarly implemented projects and newly improved functionalities. |
| Web Application and App | Web Application and App design and development will begin after frontend design is taken into regards. |
| Design | Design documents (UI, Architecture, Database) |
| Implementation of frontend | Website – With the research and implementation of HCI principles, we will integrate and execute the required principles and research into the frontend. |
| App – Similar concept will be followed for the app as well. |
| Implementation of backend | Website – researching efficient storing mechanisms, properly server and client research, use of API’s and more. |
| App – Similar protocols will be followed accordingly from the website. |
| Testing | Three stages of testing will be allocated to the project – integration testing, unit testing and system testing. |
| Maintenance | Any possible new functionalities, possible cost reductions, new idea implementations and bug fixing to get a proper working project. |
| Report | Documentation of the process of the project, future implementations will be added to the report. Detailed diagrammatic representation will also be included. |

Table 1: Project Plan

Bibliography

Alakeel, A., 2009. [Online]   
Available at: https://www.researchgate.net/publication/268200851\_A\_Guide\_to\_Dynamic\_Load\_Balancing\_in\_Distributed\_Computer\_Systems

Anon., 2018. *Red Hat.* [Online]   
Available at: https://www.redhat.com/en/topics/middleware/what-is-middleware

Anon., n.d. *Intelligent Application Availabity.* [Online]   
Available at: https://us.sios.com/what-we-do/high-availability/

Anon., n.d. *Transparencies.* [Online]   
Available at: https://www.cl.cam.ac.uk/~jac22/books/ods/ods/node18.html

Anon., n.d. *What is an API?.* [Online]   
Available at: https://www.mulesoft.com/resources/api/what-is-an-api

Assistance, C. f. E. i. D. M. &. H., 2021. *SRI LANKA: DISASTER MANAGEMENT REFERENCE HANDBOOK 2021.* [Online]   
Available at: https://www.preventionweb.net/publication/sri-lanka-disaster-management-reference-handbook-2021

Bradley, S., 2011. *vanseo design.* [Online]   
Available at: https://vanseodesign.com/web-design/simplicity/  
[Accessed 08 December 2021].

Dix, A., n.d. *human-computer-interaction.* [Online]   
Available at: https://www.interaction-design.org/literature/topics/human-computer-interaction

Eilts, M., 2018. *The Role of Weather—and Weather Forecasting—in Agriculture.* [Online]   
Available at: https://www.dtn.com/the-role-of-weather-and-weather-forecasting-in-agriculture/

Guide, S. L. -. C. C., 2021. *Agricultural Sector.* [Online]   
Available at: https://www.trade.gov/country-commercial-guides/sri-lanka-agricultural-sector

HABY, J., 2021. *REASONS FOR A WEATHER FORECAST.* [Online]   
Available at: https://www.theweatherprediction.com/habyhints3/985/

Heroku, n.d. *About.* [Online]   
Available at: https://www.heroku.com/about  
[Accessed 01 December 2021].

Nginx, n.d. *What is NGINX?.* [Online]   
Available at: https://www.nginx.com/resources/glossary/nginx/  
[Accessed 01 December 2021].

Ryan, 2019. *Webiotic.* [Online]   
Available at: https://www.webiotic.com/api-for-mobile-apps/

toptal, n.d. *Why The Hell Would I Use Node.js? A Case-by-Case Tutorial.* [Online]   
Available at: https://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js  
[Accessed 01 December 2021].

Unkown, 2014. *Software Construction.* [Online]   
Available at: https://web.mit.edu/6.005/www/fa14/classes/17-concurrency/  
[Accessed Fall 2021].

Wikipedia, n.d. *Node.js.* [Online]   
Available at: https://en.wikipedia.org/wiki/Node.js  
[Accessed 01 December 2021].