Rainfall Data Collecting System

By

CodeCraft

3. System requirements specification

3.1 Functional

collecting data from the user

- starting time and the location of the rainfall
- ending time and the location of the rainfall
- intensity of the rainfall (1,2,3,4,5)

are the most important and collectable data from rainfall which we can collect from the user without annoying or exhausting the user.

Time: If the user gives the command to start and end the system will automatically keep track of the time and get that data. Recorded in Day, Hour, and Minutes.

Location: With the user's permission from the beginning, the system will be able to keep track of the user and get his location data without interfering with the user.

Intensity: The user can give his view of the intensity of the rainfall as the user feels.

Maintain the accuracy of the data

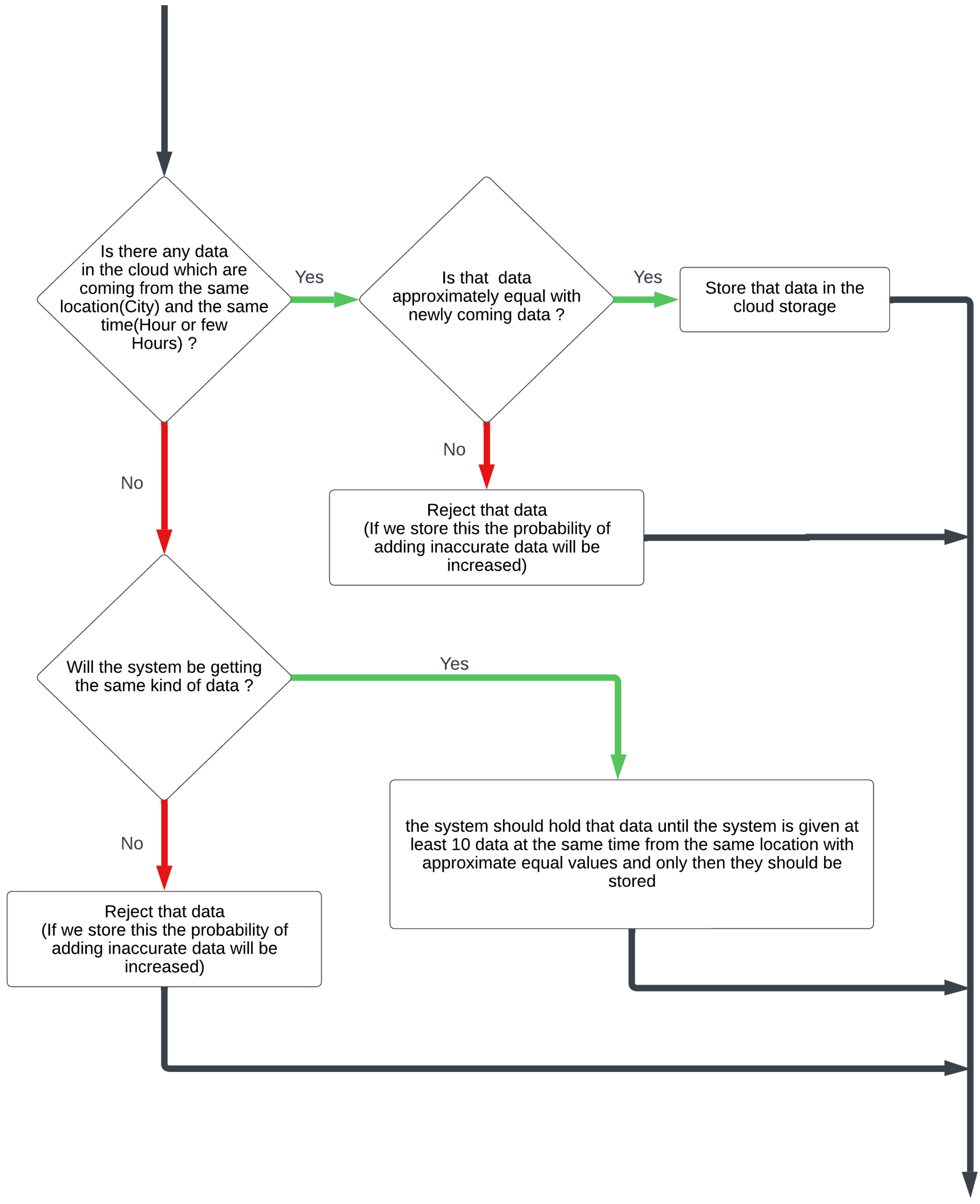
Before sending data to the cloud, we should confirm the data manually by the user, and if there is a mistake or inaccurate data that is recorded due to the user's carelessness, the system should give another chance to recorrect them manually. In that case,

Time: Give a chance to change Minutes and the Hour Manually

Location: Give a chance to select the corresponding city/nearest city

Intensity: Give a chance to change this as the user wants

After sending the data to the cloud, the system should follow the steps shown in the below diagram.


Is there any data in the cloud which are coming from the same location(City) and the same time(Hour or few Hours) ?
Yes
Is that data approximately equal with newly coming data ?
Yes
Store that data in the cloud storage
No
No
Reject that data (If we store this the probability of adding inaccurate data will be increased)
Will the system be getting the same kind of data ?
Yes
the system should hold that data until the system is given at least 10 data at the same time from the same location with approximate equal values and only then they should be stored
No
Reject that data (If we store this the probability of adding inaccurate data will be increased)

Storing data in the cloud

After validating the data and checking accuracy, the system should store those data in the cloud.

3.2 Non-Functional

Security of data

Cloud storage should be more secure. They should be read-only because the system adds data after checking the accuracy, so there is no need to change them afterward.

If the data in the cloud storage is gained by another third party except the system and the user, it will be harmful to the privacy of the users. So it is better to consider this fact when selecting an appropriate cloud storage.

In order to maintain security, the application should be unbreakable and impenetrable to hackers. Because the application keeps track of the user's GPS data, the application will be a super opportunity to harm users' privacy by hackers.

For cloud security, we suggest providing user authentication based on the email address and the password for each user to access the cloud database, using safe data transferring protocols such as HTTPS and encrypting data.

For local security, we must make the application super impenetrable, and we suggest encrypting stored data.

Keeping the size of collected data at a low level

Users will be stepped out from the system if it consumes more data to upload the rainfall data. If the size is increased the time to upload will be also increased and in that case, the lack of a good internet connection will affect the uploading.

To get rid of that situation, the system should use TFTP.

Keeping the users' historical data

The data which are uploaded by the user will be stored locally. So that user can use them for guessing or predicting their personal situations.

Reliability

The System should be reliable to the user to keep them with the system. The System does not give many more things to the user rather than being a logbook about the rainfall data that the user has experienced. So, the system should use less effort from the user to collect data and upload data. The application should be able to be used even for older people.

And the GUI also should be simple and user-friendly.

The hardware requirements also should be kept at a low level, such that anyone with email, password, mobile phone with a touch screen and a GPS module, and internet connection at least one or two minutes per day to upload data.

Speed

The system should be quick, if it isn’t, the users will be stepped out from the system.

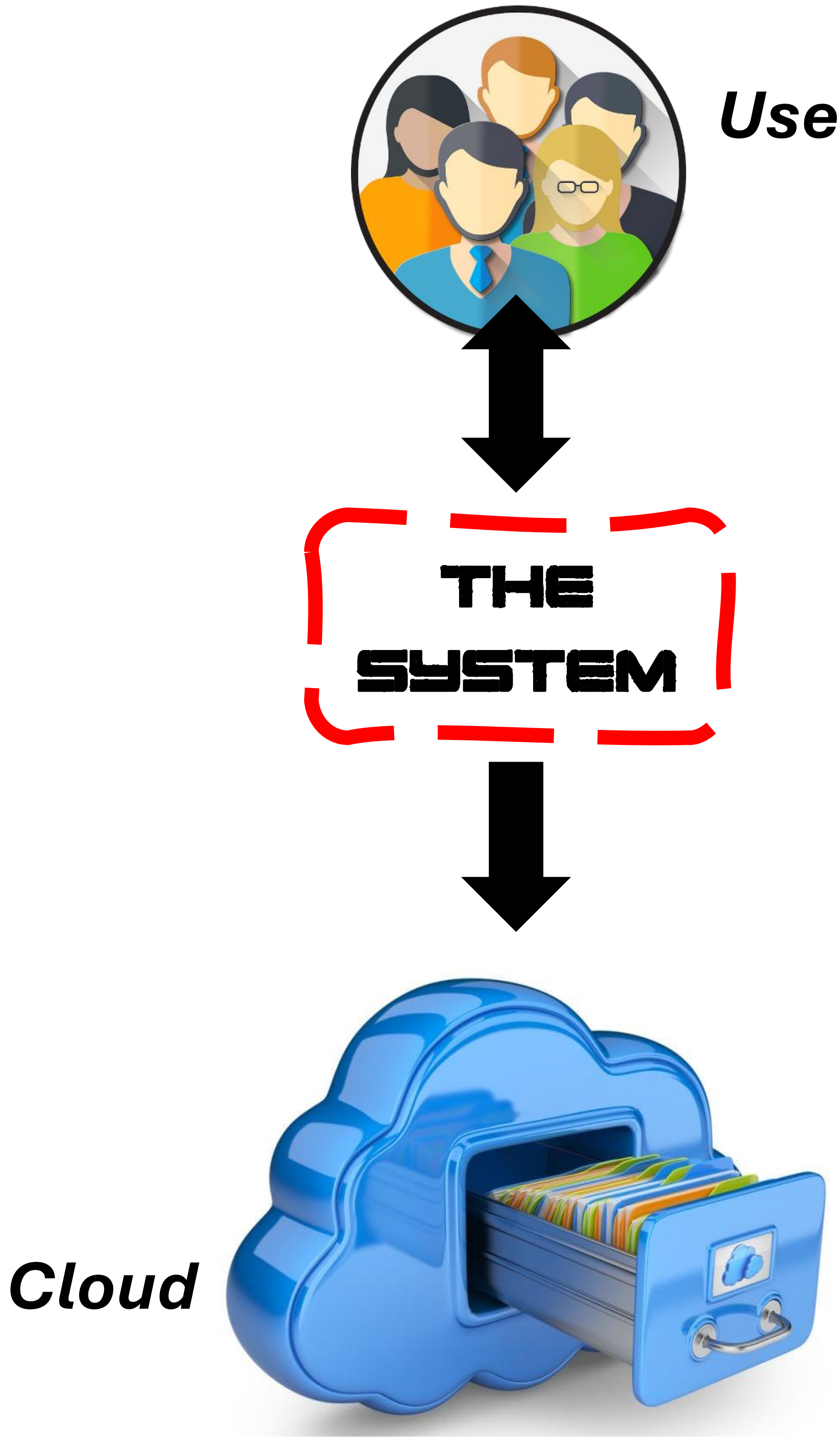
So, we have to use the fastest ways to collect data, check accuracy, and upload data.

Motivating people to add data

Without any motivation to stay with the system, we can’t hold users. So, we are suggesting a rewarding method in order to give “**Accurate data**”. Since the system checks the accuracy of the data, we can keep track of the amount of the data that has been uploaded by the corresponding user, and monthly or yearly we can give a reward to the users.

Ex: Star Points

However, there is a problem with getting night time data because the number of active users will be too small and the probability of storing data in the cloud storage will be nearly zero (according to our accuracy checking method). To get rid of that, we suggest increasing the reward during the night. Somehow it will be useful to increase the amount of data during the night rather than a void.



The relationship between users, the system, and the cloud storage