# Design Document for Log Bot

## Problem Statement

Every day we come across various issues across different apps for which we analyse logs. Most of the time issues are same but the time to analyse the issue differs based on persons knowledge about the workflow and debugging steps. For example, if Android Enrolment fails or our id env is down, there are some standard strings which we search in logs to get the exact issue. But what if you don’t know the string to search in logs.

## Solution

**Log Bot** is a one-step solution to analyse the logs and show / fix the probable issue. Just the user / script can upload log file in **Log Bot** and mention the issue, then **Log Bot** will do the rest from searching different issues and listing down the probable cause. Also **Log Bot** will have the capability to suggest / fix some of the common issues based on analysis.

## How **Log Bot** Will Work

**Step 1 Log Collection**

Either we can collect logs manually and upload to **Log Bot** or scripts can be written to fetch the logs from various sources and pass it to **Log Bot**.

**Step 2 Log Analysis**

Once **Log Bot** receives the log file it searches pre-defined strings in the file and lists down the accurate matches and probable cause of issue along with debugging steps.

**Step 3 Required Action**

Based on log analysis Bot will have the capability to fix certain most commonly issues. For example, how about if a Bot can analyse issues after deployment and automatically do the needful.

## High Level Design

### Log Bot UI

From UI anyone can upload log details and select various criteria to analyse log. With UI anyone can upload log file, specify criteria and perform log analysis.

### Log Bot service

This will enable Log Bot to be integrated as a service for log analysis / debugging / fixing the issues. For example, if log bot has to be integrated with CI-CD pipeline then anyone can use log bot as a third party service for log analysis and debugging.

## Low Level Design

### Log Bot UI

**Upload Log Details** – From this UI developers can upload log details specifying the exact string and the issue description if this string appears in logs. For example if any log is printed for web service failure, developer can specify the log string here and the name of web service which have failed. Further there is an option to mention the debugging steps as well.

**Log Analysis UI** – From this UI anyone can specify criteria and upload log file and search. Log Bot will search for all the possible conditions and list down the analysis details.

**Log Bot Backend Service** – This will be responsible for handling all kinds of backend actions like storing the log details, analysing the logs, handling requests from third party clients etc.

### Log Bot Service

Instead of adding logs and getting the analysed data in UI, any third party client can integrate Log Bot service in their project and use REST services to communicate with Log Bot.

## Infrastructure Details

|  |  |
| --- | --- |
| Parameter | Value |
| Programming Language | Frontend – HTML, CSS, Javascript  Backend - Python |
| Framework | Frontend – Bootstrap  Backend – Python Flask / Python Django |
| Architecture | MVC |
| Database | MySQL, IBM DB2 |
| Hosting Services | IBM Cloud |

## Work Estimate

|  |  |  |
| --- | --- | --- |
| Work | Description | Resource |
| Designing UI for log upload and log analysis |  |  |
| Backend logic to search logs and return the result. This will also include storing log data in database. |  |  |
| Demo for using log bot as a service |  |  |
| Collecting log data |  |  |