

**SATHYABAMAINSTITUTE OF SCIENCE AND TECHNOLOGY**

**(DEEMEDTOBEUNIVERSITY)**

**AccreditedwithGrade“A”byNAAC**

**JEPPIAARNAGAR,RAJIVGANDHISALAI,CHENNAI 600119**

**MARCH– 2023**

**CTSCAFÉPROGRAM**

**HEALTHCHECKWITHPYTHON**

**BYSTUDENTS:RANJITH.G(TEAMLEADER) - 41731100A.MOHAMEDAAFTAAB - 41731075**

**KAUSHIK L – 41731862**

**R.RONALD - 41611165**

**MENTOR: Ms Parveen**

|  |  |  |
| --- | --- | --- |
| **TechnicalSolutionApproach**  CONTENTS: | | |
|  | **TOPICS** | **PAGENO** |
| 1.Introduction | 3 |
| 1.1Aboutthisdocument | 3,4 |
| 1.1.1Purpose& Scopeofthedocument | 4,5 |
| 2.ComponentDesign | 6 |
| 2.1ComponentDesign  Diagram | 6 |
| 2.1.1 Low Level Design | 7,8 |
| 2.1.2 Technology and Frameworks to be used | 9 |
| 3. Solution Approach | 10,11 |
|  |  |

**1. INTRODUCTION**

Application health has become the core of any business. An application which is not stable(unhealthy) leads to lots of revenue loss, customer trust and negative impacts on business. To avoid such cases, we should regularly monitor the status of our application. Using these health checks, we can be notified of outages before customers and reduce the impacts of outage by fixing them or informing the customer about the ongoing outage with time to up which leads to gaining the trust.

A health check is a way of monitoring an application's health, by verifying that all of its dependencies and services are running and responding as expected.

Health checks are typically used in production environments to ensure that an application is running smoothly, and can be integrated into a continuous delivery pipeline to detect issues before they become critical.

With health check, developers can create custom health checks that can be run at any time to verify the state of the application. The library provides a simple API for defining health checks, and includes a number of built-in checks forcommon services such as databases, cache systems, and message queues.

Monitoring depends on complexity and criticality of the application. A simple application may require some monitoring its dependencies once a day, whereas a critical application may be required to be monitored as frequently as possible. We can also provide a status page to see the results and add functionality to notify developers about problems.

Health check can allow near-real-time information about the state of your application. It helps the end users to constantly monitor the health of the program, be proactive in identifying any difficulties, and make the appropriate adjustments rather than waiting on the end user or system to alert them there isa problem with the application. The health check app makes it easy to send yourtest results and other information directly to you.

**1.1 ABOUT THIS DOCUMENT:**

We are using Flask and redis to implement health check. Health Check is a Python library used to build and monitor the health of services. It provides a set of utilities for building simple health checks for services, containers, and infrastructure.

Health Check can be used to monitor the status of different components of a system, such as a web server, a database, a queue, or any other third-party service used by the application. The library supports different types of health checks, such as HTTP endpoint checks, TCP port checks, and custom checks.

When we create health checks, we can create very granular, specific checks for certain services, which helps us greatly when diagnosing issues with our application infrastructure, as we can easily see which service/dependency is performing poorly. Our application may still be up and running, but in a degraded state that we can’t easily see by simply using the application, so having health checks in place give us a better understanding of what a healthy state of our application looks like. Instead of relying on our users reporting an issue with the application, we can monitor our application health constantly and be proactive in understanding where our application isn’t functioning correctly and be proactive in needed.

A health check can specify a database query to run as a Boolean test to indicate if the database is responding normally.

1.1.1 PURPOSE AND SCOPE OF THE HEALTHCHECK

The purpose of performing a health check using Python is to ensure the proper functioning of a system or a service. This checkup validates the health of the system by verifying whether it is running properly or not, and whether it is meeting its performance requirements or not. It is essential to perform health checks regularly to prevent failures, improve the response time, and keep the system up to date. If any problem occurs while application is running then it will inform to the end users. Python programming language is utilized for automating the health check process and simplifying the detection of performance and security issues. This Python-based health check process can be scheduled to run at regular intervals and send notifications in case of any issues found. Overall, health checking using Python is vital to improving the reliability and availability of the system, ensuring the continuous delivery of services, and enhancing the customer experience.

SCOPE OF THE HEALTH CHECK

The scope of healthcheck using Python can vary depending on the specific use case, but generally involves testing the integrity, availability, and performance of various components in a system or application. Some common areas that might be covered by a healthcheck include:

**Database connectivity**: Checking that the application can successfully connect to and interact with its databases

**Network connectivity**: Testing that the application can access external services and communicate with the network

**Server resources**: Checking that the application has sufficient resources such as CPU, memory, and disk space

**API endpoints**: Testing that the API endpoints are functional and responding as expected

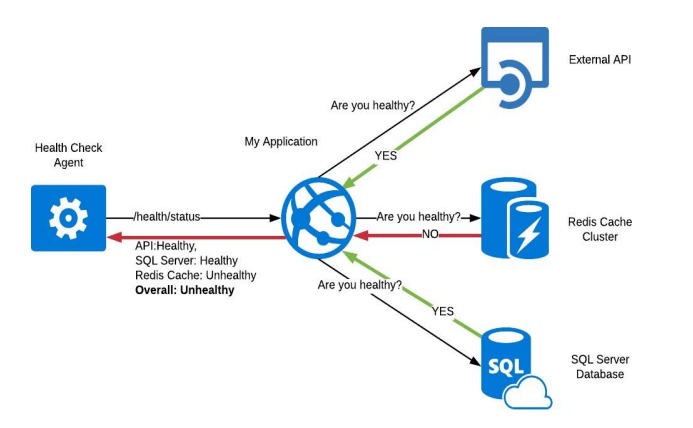
**Security**: Ensuring that security measures such as password strength, encryption, and firewalls are in place and working properly.

**Error logs**: Checking application error logs and identifying the potential issues that require debugging.

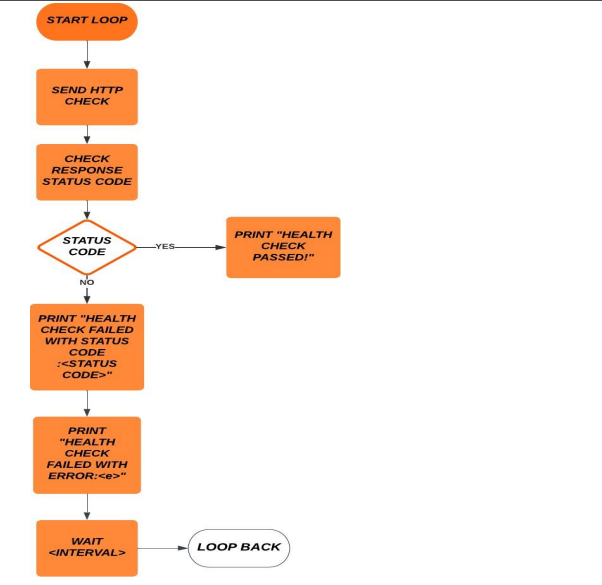
By implementing a comprehensive healthcheck strategy that includes multiple areas like these, developers can ensure that their applications are reliable, secure, and performant, reducing the occurrence of production failures and improving overall user experience. Python, with its extensive range of libraries, can help in automating the healthchecks and provide a more efficient diagnosis of issues.

**2. COMPONENT DESIGN**

In this topic we will look into how we can design a proper health check component using python programming, along with the flowchart to illustrate the process. First, let’s start with the python code to get a better understanding.

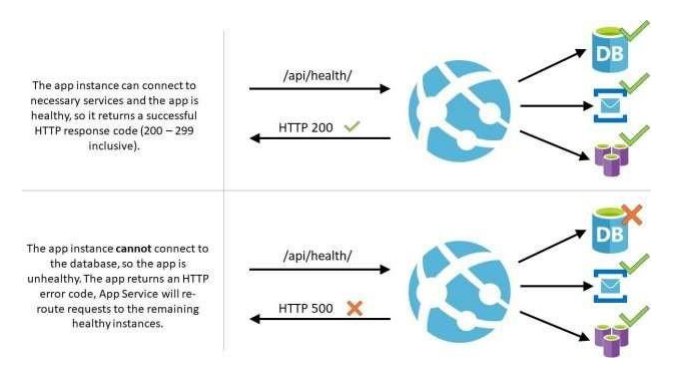
2.1 Component Design Diagram:

To illustrate the flow of this program:Flowchart for health check:



The above flowchart shows the basic process of the health check component. The loop continually sends HTTP requests to the specified URL and checks the response status code. If the status code matches the expected value, the health check passes and a message is printed to the console. If not, a different messageis printed depending on the type of error encountered. The loop then waits for a specified interval before starting the process again

2.1.1 LOW LEVEL DESIGN



**3.TECHNOLOGYANDFRAMEWORK**

Building a healthcheck system requires a combination of technologies and skills,including backend and frontend development, data processing, visualization, anddeployment.Inourproject weareusingthefollowingtechnologies.

**Visual Studio 2019**:Visual Studio is an integrated development environment(IDE)thatwillbeusedtodevelopthebackendofahealthchecksoftwar eusing.NET Framework or .NET Core. Visual Studio provides a rich set of tools and features thatmake iteasier to write ,test, and deploycode.

**SQL lite:** SQLite is a software library that provides a relational database management system (RDBMS) implemented as a small, lightweight, and self-contained SQL database engine. It is not a standalone application but rather a library that can be integrated into other software applications. SQLite is widely used as an embedded database engine in various applications and platforms due to its simplicity, small footprint, and ease of use.

**Flask:** Flask is a popular web framework for building web applications using the Python programming language. It is known for its simplicity, flexibility, and ease of use. Flask provides the necessary tools and libraries to handle HTTP requests, routing, template rendering, and more, making it an excellent choice for developing web applications and APIs.

**Redis**: Redis (REmote DIctionary Server) is an open-source, in-memory data structure store that can be used as a database, cache, and message broker. It provides a simple yet powerful key-value store that supports a wide range of data structures and offers high-performance data access and manipulation.

**Python :** Python is a popular programming language that will be used to buildthe backend of a Healthcheck system. With its extensive libraries and frameworkssuch as Requests and Selenium, We will use Python to automate the process of checking the health of various components in the system, such as servers, databases, and applications. With help of Python we wil create scriptst hat can be run in a Docker container, which will be deployed to various environments to run the Health check system