

Abstract

Welsh Clinical Portal (WCP) is a digital application used by healthcare professionals across Wales to access up-to-date patient information. Its key features include viewing GP records, radiology images, results, and medical documents. WCP relies on multiple data source services, and when these are slow or unavailable in the System-Integration-Testing (SIT) environment, diagnosing issues can be time-consuming for testers.

Aim

The aim of this project was to develop a monitoring webpage for WCP services, with the design guided by S. Few's [1] recommendations on effective dashboard layouts.

Objectives

- 1) Collect & analyse WCP services to be monitored
- 2) Design and build a user-friendly webpage that displays the service's statuses by making API calls to their endpoints and checking their response times and status codes
- 3) Test and evaluate the webpage using unit, system, usability & acceptance testing

References

- [1] S. Few, *Information Dashboard Design: The Effective Visual Communication of Data*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2006.
- [2] Y. Ravalji and R. Mishra, "Optimizing Angular Dashboards for Real-Time Data Analysis," *International Journal of Multidisciplinary Innovation and Research Methodology (IJMIRM)*, vol. 3, no. 4, Oct.-Dec. 2024. [Online]. Available: https://www.researchgate.net/profile/Vardhansinh-Ravalji/publication/391428947_Optimizing_Angular_Dashboards_for_Real-Time_Data_Analysis/links/6816a444df0e3f544f51c6db/Optimizing-Angular-Dashboards-for-Real-Time-Data-Analysis.pdf [Accessed: May 7, 2025].

Service Checker

DHCW Test Environment Service Status Page

Check All

Service Name	Service Status	Check Status
Mock API	Available (HTTP Response = 200 / Response Time = 167ms)	Check
Master Patient Index Service (MPI)	Available (HTTP Response = 200 / Response Time = 136ms)	Check
Notification Service	Slow (HTTP Response = 200 / Response Time = 895ms)	Check
Mini Reference Service	Available (HTTP Response = 200 / Response Time = 184ms)	Check
Documents Service	Slow (HTTP Response = 200 / Response Time = 705ms)	Check
SOLR Service	Slow (HTTP Response = 200 / Response Time = 1028ms)	Check
National Document Repository Service (NRDS)	Slow (HTTP Response = 200 / Response Time = 817ms)	Check
Test Requesting Service (WRRS)	Slow (HTTP Response = 200 / Response Time = 638ms)	Check
Patient Administration Service (WPAS)	Slow (HTTP Response = 200 / Response Time = 549ms)	Check
Referral Service (WCCG)	Available (HTTP Response = 200 / Response Time = 308ms)	Check
SNOMED Service	Slow (HTTP Response = 200 / Response Time = 530ms)	Check
WAP	Available (HTTP Response = 200 / Response Time = 130ms)	Check

Figure 1. Service Checker Dashboard

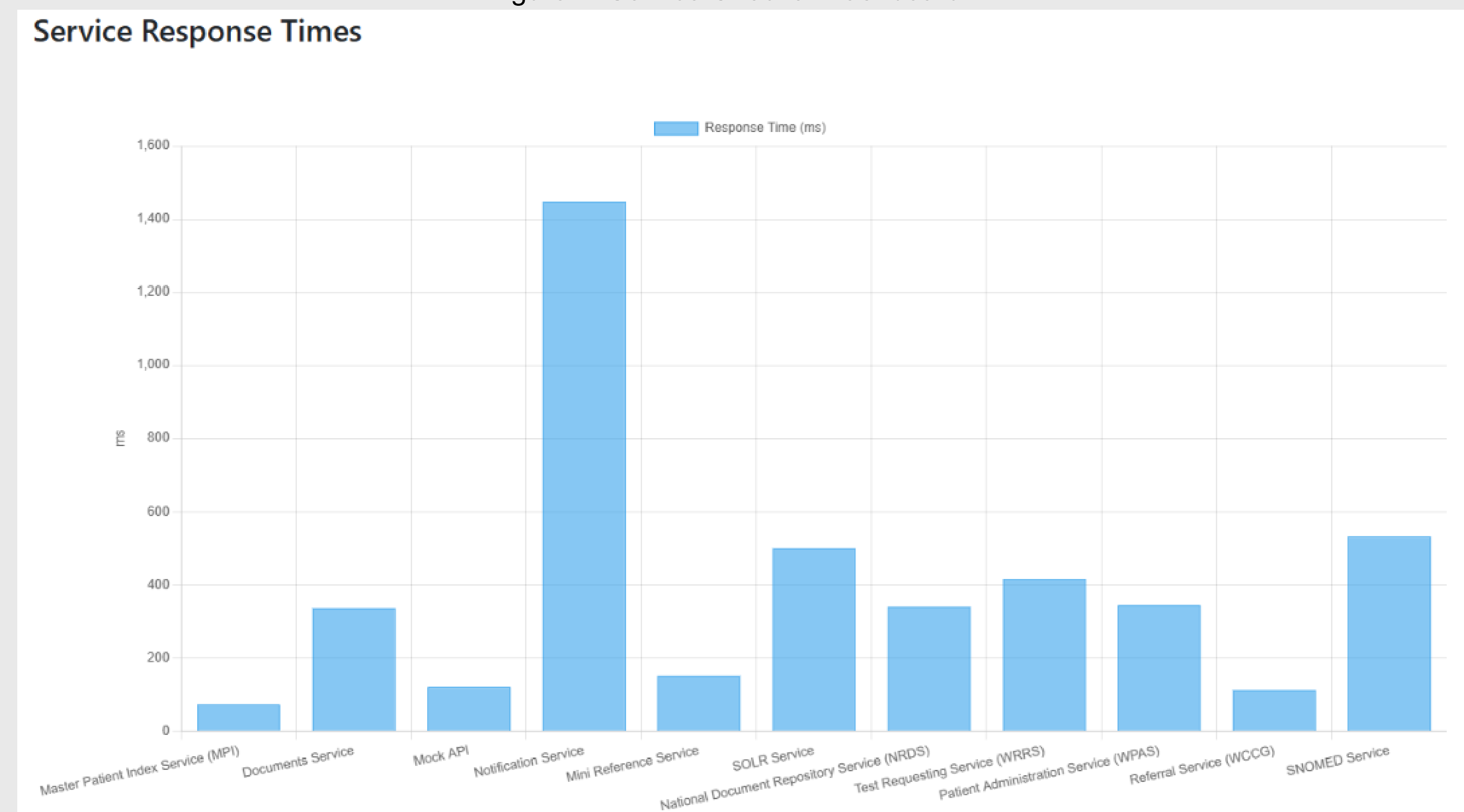


Figure 2. Service Checker Bar Chart

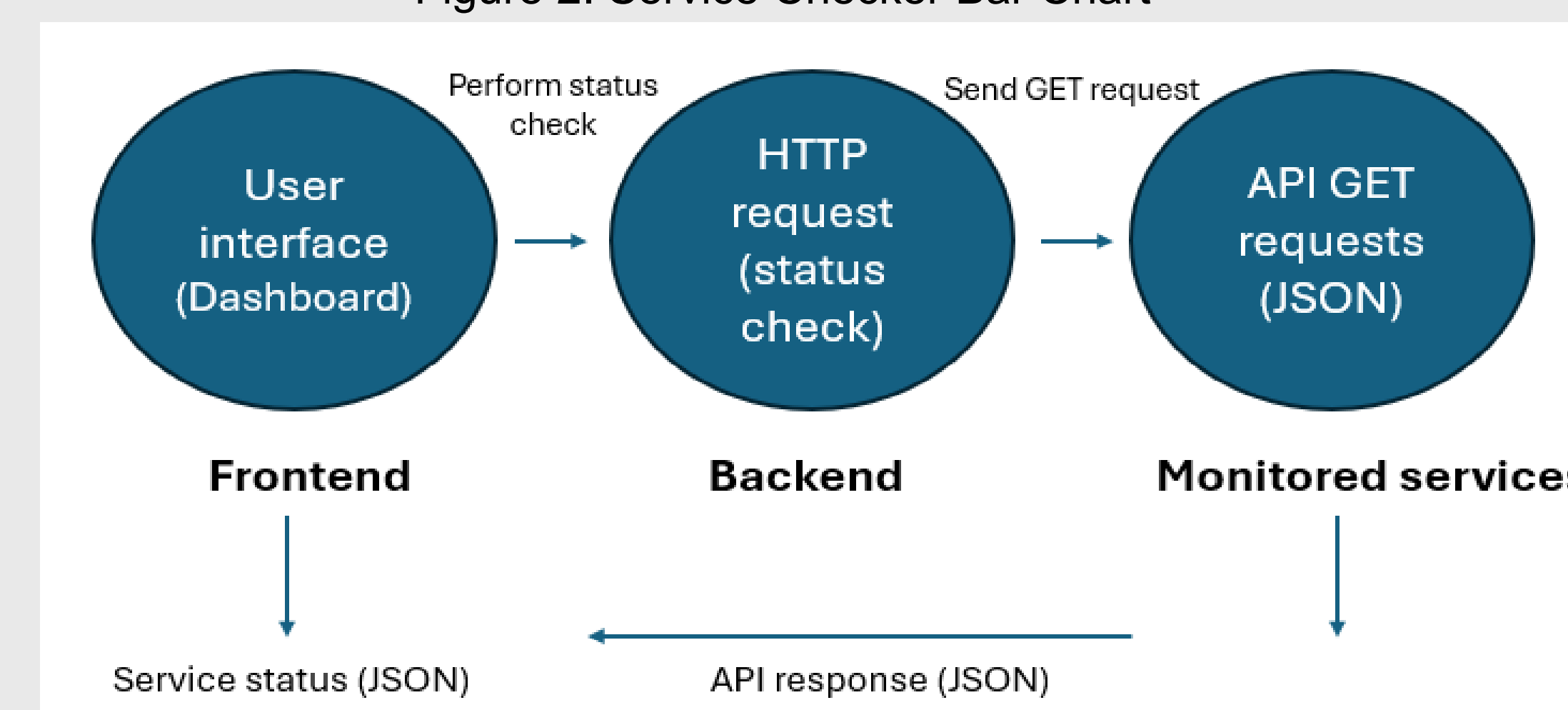


Figure 3. Dashboard's architecture

Methods

The project began by identifying the WCP services that needed monitoring. The webpage was then built using ASP.NET Core MVC, Bootstrap and Razor. The page makes API calls to each service's endpoint to check their response times and status codes, showing if they are available, slow, or unavailable.

The dashboard used manual refresh instead of automatic polling, following Ravalji & Mishra's [2] research showing this can be more effective for smaller dashboards that don't need constant updating. The availability results are displayed on a dashboard, and the service response times are displayed on a bar chart.

Unit, system, usability & acceptance testing confirmed the webpage worked as expected.

Future Work

- Integrate dashboard and bar chart into existing Test Tools webpage
- Make webpage more customisable
- Log service response times and availability to database

Results

- The page confirmed certain services, such as SNOMED and SOLR, were consistently slower than others
- Bar chart particularly helpful for quickly spotting slower services
- The dashboard made it quicker to diagnose issues than using previous manual methods such as Postman
- Testers said it was simple to use, especially helping those without Postman experience

Conclusion

- The objectives and aim of the project was met, with a working dashboard that helps testers check service availability & diagnose issues faster
- The project showed how a clear dashboard can improve efficiency and support everyday testing work