01/02/2024

Arawn Davies

Year 3 Computer science bsc. de montfort university

microCMDB

microCMDB Software Manual and System Documentation

Contents

[Introduction 1](#_Toc162833736)

[microCMDB Project Overview 1](#_Toc162833737)

[microCMDB.CLI 2](#_Toc162833738)

[microCMDB.OS 2](#_Toc162833739)

[microCMDB.Web 2](#_Toc162833740)

[microCMDB.Test 2](#_Toc162833741)

[Development Lifecycle 2](#_Toc162833742)

[Stage 1: Requirement Analysis 3](#_Toc162833743)

[Stage 2: Design 3](#_Toc162833744)

[Stage 3: Development 3](#_Toc162833745)

[Stage 4 Testing 3](#_Toc162833746)

[Stage 5 CD/CI 3](#_Toc162833747)

[Critical Analysis and Reflection 3](#_Toc162833748)

[What Went Right 3](#_Toc162833749)

[What Went Wrong 3](#_Toc162833750)

[What Could Be Done Differently Next Time 3](#_Toc162833751)

[Conclusion 4](#_Toc162833752)

[Appendix 4](#_Toc162833753)

[References 4](#_Toc162833754)

## Introduction

Introduce the problem your project addresses, its significance, and your objectives.)

## microCMDB Project Overview

One of the main objectives of this final year project is to provide a cross-platform system that implements a relational database model that represents a set of objects and entities that may exist in a complex network environment. For this, I knew the best development language and frameworks most applicable for this project would be C# and .NET Core, as I had the best knowledge of these through previous research and work carried out on my various software projects held publicly on GitHub and privately for Leicestershire County Council.

.NET Core provides a framework layer called ASP.NET Core, which allows a Web HTTP-based API or Model-View-Controller architecture providing CRUD functionality. Projects created using .NET Core can be ran under both Windows, macOS and Linux, and in containers such as Docker. To provide for even more simplicity, less overhead and lower system requirements, I also utilized the Cosmos Operating System project, of which I have contributed to a significant amount and served as a core team member of the project and community.

The aim is to have a common code base that provides functionality for creating, reading, updating, and deleting various entities and the relationships between them. These models, classes and methods would then be referenced and compiled along with the frontend components that provide further interactivity and accessibility for the system.

Explain each major component of your system, including the web interface and the minimalist layer you plan to add. Provide a rationale for each design decision, such as the choice of models, architecture, and technologies.

Describe how each component contributes to addressing the problem and achieving the project objectives. Discuss any trade-offs or challenges encountered during the development process.

### microCMDB.CLI

The CLI project includes the command line interface, a common runtime environment for interacting with the various entities and methods included in the system.

### microCMDB.OS

This project contains the Hardware Abstraction Layer which allows microCMDB to be cross-compiled to a bare-metal x86 Assembly operating system kernel. This would allow microCMDB to be ran on a wide range of different platforms utilising the x86 processor architecture, including virtualization hosts VirtualBox, QEMU and VMware, to physical hardware hosts such as an Intel NUC or other Small Form-Factor nodes.

It uses the C# Open-Source Managed Operating System as a .NET cross-compiler that allows the Windows/Mac/Linux CLI project to share a common codebase with the freestanding OS variant.

### microCMDB.Web

One of the main components of this project is the Web UI interface, currently built using a SQLite and Entity Framework backend with ASP.NET Core MVC.

### microCMDB.Test

The fourth main component project of the microCMDB system is the test suite, comprised of classes and methods which will extensively test the system-specific methods.

## Development Lifecycle

Development of the microCMDB project will be done in a series of steps.

Firstly, is the Requirements Analysis stage. This involves the collection, examination and investigation of a system to be implemented that provides a solution to the problem. Secondly, there will be the initial Design work of the system and its underlying system architecture and frameworks. It is of paramount importance that the design work and accompanying documentation is of good detail before developers can begin work on an initial implementation. During this stage, the initial models are created and scaffolded methods and functions are defined.

Once a sufficiently operational system has been implemented, initial Testing work is carried out. It is during this process where analysis of the system is carried out and the degree to which it meets the requirements, as defined during the second stage, is measured using metrics and testing frameworks.

Before the system is fully operational and live, there is the process of Delivery and Deployment. This involves the installation of the system and setup of permissions in its production environment, i.e. prepare for usage in real-world business operations.

Lastly, is the Review and Feedback stage, where user input and customer feedback is collected from the end-user and used to further improve the systems security, functionality, performance and any other issues which may have arisen during design and development.

These must be completed in the order specified for maximum operational and developmental performance, but it can be expected in this scenario to be revisiting steps while development continues.

Discuss the software development methodology you followed during the development lifecycle. Provide evidence of the application of an appropriate software development methodology. Discuss each major stage and how validation and verification were applied. Explain why you chose this methodology and how it was applied throughout the project.

### Stage 1: Requirement Analysis

During traditional software development, the analysis of requirements focuses on the needs, expectations, and objectives of the stakeholders, whether that’s the end-users, product owners or developers and other persons involved. For the microCMDB project specifically, the stakeholders would be my supervisor, advisors, and peers who provide peer reviews and feedback, and the academic staff in charge of the final year projects of students and submissions.

Unlike typical software development projects, the objectives, functionality, and requirements for microCMDB are self-identified and documented as opposed to usually either provided solely or contributed towards by the client and/or end-user. As a result, focus can be dedicated to compliance of the academic guidelines instead of stricter and more complex restrictions dictated by a third party.

### Stage 2: Design

(Continue describing each stage.)

### Stage 3: Development

Provide a detailed overview of each stage of the development lifecycle, such as requirements analysis, design, implementation, testing, and deployment. Explain how validation and verification techniques were applied at each stage to ensure the quality and correctness of the system.

### Stage 4: Testing & CI

Appveyor and C#’s MSTest unit testing suite were used to provide continuous testing and build feedback. Instead of having to constantly clone and compile the database system each time I want to check the status of a single component, I can visit the Appveyor project site and view the status from there. Docker was also chosen to provide containerization support of the microCMDB system, further assisting in achieving one of the main objectives including cross-platform support.

### Stage 5: Review and Feedback

## Stage 6: Deployment and Delivery

## Critical Analysis and Reflection

(Discuss what went right and wrong, what could be done differently next time, and provide an appraisal of the product. Analyse the approach taken with hindsight and the software/tools used.)

Reflect on any modifications or adaptations made to the development process based on feedback or changing requirements.

Reflect on both the successes and challenges encountered during the development of the microCMDB project. Discuss what went right in terms of meeting project objectives, implementing features, and delivering a functional system.

Analyse areas where things did not go as planned or where improvements could have been made. Consider factors such as technical difficulties, design limitations, or project management issues.

## What Went Right

(Discuss the successful aspects of your project.)

## What Went Wrong

(Discuss the challenges and failures in your project.)

## What Could Be Done Differently Next Time

(Reflect on the lessons learned and how they can be applied to future projects.)

Offer a critical appraisal of the final product, highlighting its strengths and weaknesses. Discuss the overall effectiveness of your approach and any lessons learned that could be applied to future projects.

Provide insights into what you would do differently next time based on your experiences with this project. This could include changes to the development approach, tools used, or project management strategies.

## Conclusion

(Summarize the key points of your report and the significance of your project outcomes.)

## Appendix

(List the references you cited in your report.)

## References

(List the references you cited in your report.)