Tom Davison

Software Engineer

0422931765 | TomWilliamDavison@gmail.com | TomDavisonDev.com

SUMMARY

4 years of experience developing, testing and maintaining a suite of software utilising .NET core, MVC design patterns, C, C#, WPF, WinForms, bash and API development. Proficient with Linux based operating systems, Jenkins CI/CD pipeline. Personal experience developing web apps using MongoDB, ExpressJS, AngularJS, Node.js as well as Gastby. A detail-oriented and self-driven individual with strong critical thinking skills. Proven ability to work independently on a wide range of products while delivering high-quality results that meet project requirements and industry standards.

EXPERIENCE

Personal Project – EasyTest

I developed EasyTest, a web-based test management system. This platform allows users to better track and manage customer requirements while providing a document trace to an internal testing kit which allows the creation test cases targeting the testing of software functionality. The development of this included:

- The development of a REST backend, featuring a secure login server as well as various endpoints for retrieving, deleting, adding, inserting as well as getting data for the frontend UI.
- Implementation of reactive objects representing database statistics or broader functionality such as movement of test steps.
- Implemented start scripts and endpoint tests to ensure platform is running smoothly
- Successfully released the first version of the platform, which is currently running on a live server with SSL authentication.

Software Development Engineer — Axxin | March 2020 – April 2022

As a developer for Axxin, I worked on the development of .NET code running on a standalone device using the Mono framework, as well as firmware written in C and accompanying desktop systems using WinForms and WPF. My role included debugging and troubleshooting multiple branches of a full stack system that has been in production for over 10 years.

Automation API

- Design and implementation of a device API for R&D uses including the modification of system database entries as well as execution of key components via firmware hooks.
- Integration with key systems, including allowing API to retrieve, validate and import configuration and data files.
- Modification of device firmware in C to allow for finer control over hardware events through the Automation API.
- Creation of a scripting language and interpreter for sequential execution of API commands.

Project Magneto

- Design and addition of a new multi-page workflow which allowed R&D users to insert and identify new test cartridge equipment as well as run complex logic including API commands during the test workflow via a custom scripting language.
- Intergration with Automation API, giving R&D users the ability to run several software and hardware operations during testing sequences
- Addition of a scripting language, which allowed user to run lists of commands with more complicated logic during a test workflow
- Released fully integrated system to customer which allowed R&D team finer control over creating test workflows.

Software Quality Assurance Engineer — Axxin | November 2018 – March 2020

As a Quality Assurance Engineer at Axxin, I was responsible for the quality standard of the software. This included testing new features and bugs as well as the creation and execution of software test cases, protocols, and validation reports that met internal, ISO, and FDA standards. In addition to these responsibilities, I also implemented several scripts and efficiency tools for the team.

- Design and programming of medical device scripts and macros used by the quality assurance team to reduce time commitment and margin of error.
- Troubleshooting and analysing software in Bash, C#, and C code, iterating through changes and identifying inconsistencies.
- Creation of macros in Bash to automate test running and expose software defects.
- Created a tool when the team transitioned between test management systems which greatly increased the speed at which test protocols could be imported. Allowed to the team to work on more crucial deadlines

Automation API

- Creation of macro and script templates for co-workers from other disciplines to create responsive macros without deep programming knowledge.
- Designed and created macros and scripts which dynamically responded and reacted to device level events
- Created macro and script templates which allowed co-workers from other disciplines to create responsive macros without having a deep understanding of programming

Device Mirror Software

- Independently created software using C# and windows forms which utilized the automation API to control medical devices remotely
- Expanded upon the creation of this project to allow users to create and record macros using GUI events
- Further expanded the software to support the exporting of created macros to Bash

EDUCATION

Bachelor of Computer Science, Swinburne University of Technology, 2018

REFERENCES

Available on Request