Niki Micallef

nikmic94@gmail.com | nikmic94.github.com | (+356)79941014

EMPLOYMENT HISTORY

CCBill *(Software Engineer)* **Aug 2017-Present**

- Formed part of a multi-national team working on the full stack development of an administration portal. My job was to implement the required changes and ensure their quality through the creation of automated unit tests.
- Designed and implemented a number of contract-first APIs to be used within a microservice architecture.
- Owner of a data migration effort where I was an active participant in all stages of the project including design, implementation, testing, deployment and maintenance.
- Standardized the database dump process through the creation of a scripted solution and relevant documentation.

University of Malta *(Tutor)* **Oct 2016-Present**

- Delivered tutorials to undergraduate students on 'Fundaments of Software Testing' and 'Verification Techniques'.

Ricston Ltd. *(Junior Software Developer)* **Nov 2016-Aug 2017**

- Developed and maintained a number of Mule applications for a leading UK telecommunications company which copied and merged data from third party APIs and database systems into a local database.
- Compiled and maintained test suites for the applications developed by the team.
- Formed part of a team that introduced a Centre for Enablement (C4E). My role consisted of evaluating and introducing tooling such as a continuous integration pipeline while also producing the first reusable assets.

Ixaris Systems Ltd. *(Junior Programmer)* **Jul 2016-Oct 2016**

- Worked within a team of 8 people using the Scrum and BDD methodologies. Our team was focused on improving existing products with the intent of supporting a major client. This included introducing new functionality both to the core platform and to the APIs.
- Improved the administration console by adding or modifying front-end views in order to support new functionality.
- Enhanced the test suite by writing automated tests for front-end and back-end components within the system (unit, integration and end-to-end) while maintaining high levels of code coverage.

EDUCATION

University of Malta *(MSc. Computer Science – Mainly Research)* **2017-2019**

- Investigated the advantages of specification model-based performance testing through a comprehensive use-case evaluation. Through the research we determine whether a model-based approach produces more realistic user behaviour in comparison to a deterministic solution and whether the former solution is more maintainable.
- Completed a number of study units including computer graphics, concurrency and distributed computing, systems security and cryptography.

University of Malta *(BSc. Hons. Computing Science - Second Class Honours, Upper Division)* **2013-2016**

- Gained a holistic approach to computing science by learning about various topics including data structures and algorithms, programming paradigms, data science, software engineering and software testing amongst others.
- Used a number of programming languages including Java, Python and Scala while working on various assignments.
- Investigated the benefits of runtime monitoring within the domain of cheat detection for games through the creation of a maintainable, reusable and language independent behavioural cheat detection framework which achieves a high level of cheat detection while introducing negligible overheads.

LANGUAGES, LIBRARIES AND FRAMEWORKS*

Beginner – Python, Scala
Intermediate – Cucumber, CSS, Git, HTML, Mule ESB, Selenium, SQL
Advanced – Angular 2+, Java 8, JavaScript, Maven, RxJS, Spring (AOP, Boot, Core, Data, MVC), TypeScript
Master - JUnit

COURSES

- **LAS3014 – The Spring Framework** *(University of Malta)*
- **LAS3014 – Modern Java for the Java Developer** *(University of Malta)*

SOCIAL

- **GitHub** – github.com/nikmic94
- **LinkedIn** – linkedin.com/in/nikimicallef
- **Skype –** niki.micallef

* Beginner: Know the basics and can write code snippets using examples/tutorials. Intermediate: Can write code to solve personal problems & can use documentation when problems are encountered. Advanced: Can write, refactor and optimise code. Can teach the language to a peer. Master: Can produce well-structured, industry grade code using best practices available.