CV – **Jelle De Vleminck**

Personal Information

10/06/1997
Ternat
Belgian

Bachelor Computer Science, Cum Laude

Profile

Junior Java Developer

Java	Junior
Spring (boot)	Junior
Software security	Junior
Javascript	Junior

Key Certificates

Issuer	Certificate	Acquired
Oracle	OCA, Java SE 8 Programmer I	2018
Oracle	OCP, Java SE 8 Programmer II	2019

Languages

	Speaking	Understanding	Writing
Dutch	Mother tongue	Mother tongue	Mother tongue
English	Fluent	Fluent	Fluent
French	Intermediate	Intermediate	Intermediate

Career History

Axxes
08/2018 - Present

Client	Axxes – JAVA Traineeship
Period	08/2018
Location	Antwerp
Description	**Internal Training** General & Methodologies • Java • Developing Enterprise Applications • Version Control with GIT • Scrum • SOLID • Clean Code • Monitoring & Logging • Continuous Integration • Linux • Docker • Amazon Web Services • Communication skills Back end & databases • Spring: Spring Data, Spring Security • JPA • Hibernate • Big Data • MongoDB Front end • JavaScript • Angular • React Testing • Software Testing • Mocking & Unit Testing

Client Period Location	Axxes – Brussels Airport Company 10/2018 - Present Zaventem
Description	**Backend developer in the integration team of BAC** Automate the airport through integrations with microservices **Tools:** Java 8, Akka (actor model) framework, Spring boot, RabbitMQ, MariaDB, Atlassian stack, Openshift

College

Client Period Location	Mobco (internship) 03/2018 – 06/2018 Dilbeek
Description	**Developer** <u>Project 1:</u> **Environment:** The Mobile Monitoring Service is a platform to which several of the customers are connected and which will carry out a permanent series of tests (each consisting of a script with various elements) on the customer's infrastructure. Based on the results of the tests (and therefore also the results of each of these elements) and the frequency of these results, the customer should be proactively informed via an 'alarm' or 'alert'. Developing the Alerting Module for the Mobile Monitoring Service: • interpretation of the information to verify if' the customer must be notified • expanding the 'connectors' to the customer's systems (email, sms, SCCM, …) • to set up a visualization via the portal that allows the customer to discover the root cause as quickly as possible This project was developed within the software development team and was in collaboration with an external software company called Quamotion **Tools:** C#: .NET Core 2.0, Javascript (Vue.js), HTML/CSS, SmsEagle, Visual Studio, Git <u>Project 2:</u> **Environment:** The enterprise contacts application is an application made for the **European Court of Auditors** that runs on both Android and iOS and provides the following features: • Full listing in alphabetical order of all contacts found in Active Directory, within the predefined search filters • Detail view of the contact details • Detail view of the presence information retrieved in Skype • Detail view of the contact picture provided via an HTTP connection

	• Tasks view (tasks retrieved from Oracle database) • Useful numbers view • My colleagues view (grouped on an AD property) • My contacts view (individual list of contacts) • Continued functionalities when offline (caching) except for presence The Enterprise Contacts Solution exists out of 3 main components: • Contact Gateway Service (Skype for Business, Picture, Contact/Task/Useful Numbers get and post endpoints) • Data extractor (AD, Oracle, Useful numbers JSON) • Native Mobile Application (Xamarin Forms Ios/Android) My task in this project was to develop the data extractor and the contact gateway The data extractor is a console application that runs once, does the job and quits. Therefore, the console application should be scheduled using Windows scheduled tasks. The data that will be extracted depends on the argument passed with the executable: • "contacts" argument will extract the Active Directory contacts • "tasks" argument will extract the tasks from the Oracle database • "usefulnumbers" argument will extract the useful numbers from the appsettings.json file Every time the application runs it does the following: 1. Generate the data for the chosen argument; 2. Check if the data file already exists on the local machine, if it does, check if the generated data is different from the data in the file on the local machine; 3. If the data doesn't exist or is different, save the data to a file on the local machine and POST the data to the contact gateway endpoint. If it is the same, do nothing. Tools: C#, .NET Core 2.0, Xamarin Forms, Visual Studio, Git

Client	Comprosoft (Final Work)
Period	2017-2018
Location	Brussels
Description	Developer, Architect Environment: It often happens that employees incur costs that are reimbursed by their employer. Some typical examples of this are parking costs and restaurant costs. The employee must submit a proof to his company. This means that all paper receipts must be kept and then issued. Once the costs have been issued, they must be approved. If the costs are approved, the employee will see these costs paid back at his subsequent monthly wage. This is a lengthy process that entails some problems. It is not easy for the employee to keep an overview of the costs to be recovered. This makes it difficult for the employee to check whether all his costs have been reimbursed. The employer is at

	the end of the month with a stack of costs that he has to enter manually with the accompanying employee. Mistakes can also easily be made here. Or what happens if an employee loses his proof? The objective of the system is to simplify the life of both the employee and the employer. We want to achieve this by digitizing the process. Employees must be able to easily register costs. These costs must immediately be passed on to the employer. The employer must be able to process these costs easily. Our central research question is: How can one manage the costs of employees in an efficient and well-organized way? This final work is made by 2 people. My tasks in this project were: • Research backend technologies • Research & implementation text recognition (possibility to scan receipts) • Saving & viewing pictures • Full implementation Alexa with security features (possibility to enter costs with voice) • Statistics (backend and frontend) • Superadministrator / application management features • Filtering costs • Scanning permissions • Warning system for missing proof • Workshop Angular for students (part of the final work) • Deployment application Tools: Angular4, HTML/CSS, PrimeNG, Spring Boot, Bootstrap, Hibernate, MySql, Jaspersoft, SendGrid, Alexa, TravisCI, Git, NGINX

Client	Canguru (Project Mobile Apps - school project for external customer)
Period	2017
Location	Brussels
Description	Developer, Architect Environment: The project consists of creating a time-sheet application, an application that keeps track of the hours worked and staff absences. In addition, the application analyzes the collected data and provides feedback. The application is usable on PC, Android and iOS. I was both developer and architect in this team. We decided to use Angular and Spring Boot. No one in the team had already worked with these technologies. but due to the relevance, we decided to get started with this. I have worked out code examples in both technologies for the team. And then explained how this works so that everyone

	could get started quickly. Subsequently I mainly worked on the backend, while the others worked more on the frontend. Requirements of this project: • Cross platform • Reporting (general, personnel with overtime, number of hours worked per company, …) • Validation and feedback (overtime, fill in timesheets in time, default hours / attendances) • Different companies and user roles (Consultant, HR, Manager) • Billing in pdf Tools: Spring Boot, Angular2, Bootstrap, HTML/CSS, Git

Client	Integration project (school)
Period	2017
Location	Brussels
Description	Developer Environment: the goal of this project was to have different open source systems work together using RabbitMQ. In this project the 2nd year was divided into groups. Each group worked with a different open-source system. My team was responsible for alerting and monitoring, using ElasticSearch and Kibana. The other opensource components were: Frontend, CRM, Cash register system, Cloud Tools: PHP, RabbitMQ, ElasticSearch, Kibana

Education and Extra Curricular

2015-2018	Bachelor Computer Science, Erasmus Hogeschool Brussel
2013-2015	Informaticabeheer, Don Bosco Groot-Bijgaarden
2009-2013	ASO Economie Wiskunde
2017-2018	Tutor student-independent through “Het BijlesBureau”

Knowledge and Experience Summary

Languages & Frameworks

Expert	Java, Spring, Hibernate, HTML5, Git, Gradle, Maven
Experienced	Software security, Akka (actor model), Javascript, Angular, VueJS, CSS, C#, ASP.NET, .NET core, Entity Framework, MySQL, JEE, Amazon ASK, C++, RabbitMQ, Linux
Knowledge	PL/SQL, Oracle, MSQL, Windows Server, Usability design, Android, Swift, Xamarin, Raspberry PI, PHP, Laravel, ElasticSearch, Kibana