# Play with Scala
## (Functional Programming Course with Scala)

Author: Martin Zuber (B00066378)
Project Supervisor: Luke Raeside

## Introduction

The idea is to develop a web application for a computer science students who would like to learn Scala programming language and functional programming paradigm. The application will be developed using Scala itself and Play Framework.

### Research Questions

- What exactly is functional programming paradigm and what benefits does it bring to application development?
- What are the advantages and disadvantages of Play! framework in terms of productivity, performance and scalability and in comparison to other Web Application frameworks such as Java EE?

### Benefits

- Filling the 'educative hole' in the Computer Science courses curriculum by creating learning web resource.
- Acquiring valuable skills and deeper insides into functional programming paradigm and design patterns.
- Learning Scala programming language and better understanding of Java SE 8 platform and Java Virtual Machine (JVM).
- Building valuable skill set in contemporary web application development and acquiring a better understanding of MVC design pattern.

### Deliverables

- Working and deployed web application prototype.
- Course content in form of lectures and exercises.

### Technologies

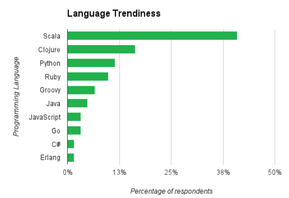
Ubuntu Server 14.04 LTS, Scala 2.11, Play 2.4.4, SBT, Typesafe Activator, MySQL server, IntelliJ IDEA 15 with Scala plugin, Git and GitHub.

## Why Functional Programming?

- Referential transparent functions are easier to reason about or test.
- Encourages safe way of programming by promoting immutability
- Very concise, expressive even elegant code.
- Modular in the dimension of functionality and reusability.
- Safe multithreading without the race conditions.
- A higher level of abstraction and pattern matching.

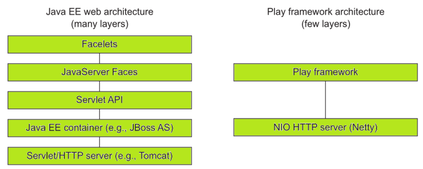
## Why Scala?

- JVM language which compiles to Java byte code.
- Fully compatible with thousands of Java libraries.
- Multi-paradigm and probably most evolved programming language.
- Scalable syntax by design.
- Powerful tools, libraries and frameworks.




## Why Play! ?

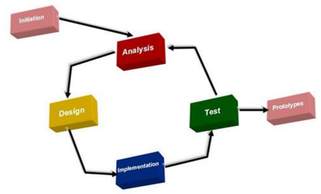
- Prefers simplicity over complexity.
- Embraces web architecture instead of hiding it
- Promotes REST style.
- Utilizes real-time reactive web programming.




**Java EE "lasagna" architecture compared to Play's simplified architecture**

## Methodologies

### Prototyping SDLC model




Instead of creating 'a big design in advance', prototyping methodology is allowing the developer to change the design in each iteration of the design / development / evaluation / refining circle. I should expect major changes in design based on my lack of experience in MVC web development and technologies I decided to use. The prototyping methodology will allow me to have simple, but functioning prototype reasonably early. As more insights are acquired from the research, the prototype can be re-designed to implement additional functionality or re-thought the design flaws in terms of performance optimization and the functional programming design patterns additions.

## Expected Results

The project would be considered successful if it would meet at least two following criteria and goals. In the case of three or more goals accomplished I would consider project outcome to be very successful.

- The comprehensive research on functional programming paradigm and design patterns performed.
- Acquisition of the declarative style coding skills.
- The acquisition of skill-set in rapid web development using Play MVC framework.
- The gain in Scala coding skills and creation of enough learning material content for Scala course.
- The delivery of fully functioning prototype of web application.

## Conclusion

At the end of this project, I will hopefully introduce some of Scala dedicated learning material in the form of an interactive web application. It can be used in a fictional course module for a fictional college. The application main purpose is to promote the interest into Scala programming language and functional programming paradigm.

This research project will not contribute at all to the discipline area. Maybe only in a sense that it will hopefully bring one more student with the passion for programming languages to the functional programming paradigm. And maybe if other students or lecturers will see how elegant and declarative functional programming really is, the contribution could be a bit more significant.

## Future work

In the future, I would like to explore Scala in depth and study the functional programming further. For instance, in JVM ecosystem, there are other functional languages which I would like to study, such as Clojure.

At some point in the future, I'm planning to study Haskell programming language as well.

## References

*Brikman, Yevgeniy, The Play Framework at LinkedIn: Productivity and Performance at Scale*, https://www.youtube.com/watch?v=8z3h4Uv9YbE&ab_channel=NewCircleTraining (2013-06-26)

*Paul Chiusano, Runar Bjarnason, Functional Programming in Scala.,* Shelter Island, NY 11964, Manning Publications Co., 2015. ISBN 9781617290657.

*Peter Hilton, Erik Bakker, Francisco Canedo, Play for Scala,* Shelter Island, NY 11964, Manning Publications Co., 2014. ISBN 9781617290794