

Interview with JavaFX Coding Challenge student winner Ramin Mohammadi

[Student Views and Reviews](#) recently caught up with [Ramin Mohammadi](#), developer of the [Caesar Photobook Mobile](#) application, and one of the student winners of the [JavaFX Coding Challenge](#). The complete interview with Ramin follows. A shorter version and the podcast can be found [here](#).

Student Views and Reviews: How long have you been using JavaFX?

Ramin Mohammadi: I've been using JavaFX since February this year, before then I hadn't heard much about it. The reason that I started to use JavaFX is because I started my final internship at the Caesar Groep. They were really interested in an evaluation of the possibilities of JavaFX. That's why they asked me to try out JavaFX and develop a couple of applications. After that I reviewed the applications and the process of learning the environment as well as the development process, the tools, etc. I was also asked to try and evaluate the differences and similarities between JavaFX and Adobe's Flex.

SVR: Did you learn JavaFX at school? If not, how did you learn JavaFX?

RM: No, I learned JavaFX all by myself, using the available resources on the Internet, like all the documentation, presentations and videos about JavaFX that are published by Sun. For instance, I used the Programmers Guide. I also followed the 15-week JavaFX Course by [JavaPassion](#).

Additionally, I used the main JavaFX website, for all the blogs posts, discussions, best practices and of course I tried out almost every sample that had a functionality that I might have needed for my own application.

SVR: Why did you choose to develop this application?

RM: I chose this subject for my application for a couple of reasons:

- 1) For the sake of my investigation (for the internship), I was required to develop a mobile application using JavaFX.
- 2) To fill a need for a user to present his photos, which he doesn't need to have on his mobile, but can host somewhere online (in a cloud). Using the application the user can now show his albums on his mobile, with all the "all the bells and whistles," like smooth screen transitions using animation, compelling UI thanks to the use of the Production Suite Plugin and of course all the needed functionalities which are common to a slide show.

SVR: What aspect of JavaFX did you find most useful in creating your application?

RM: There are a lot of aspects of JavaFX which I find useful, but specifically for this mobile application, I found the fact that I didn't need to learn anything new about JavaFX (or work in a different way) for developing a mobile application really useful. The whole development process doesn't differ in almost any way with, for instance, the process of developing a web application using JavaFX. I experienced all the ease of development with JavaFX when developing this application, and when comparing this with the development using JavaME, I can truly state that I never could have accomplished this result with only the use of JavaME in combination with the LWUIToolkit.

SVR: How does JavaFX make your application easier to use for the end-user?

RM: Like I mentioned earlier, I tried to make this application easy to use by handling the screen transitions with animations (for instance from the Album screen to the Thumbnails screen). I also designed the UI of the application in Illustrator and used the Production Suite Plugin, so I managed to design a completely new UI for my application, which could fill all the user's needs. Finally, I made this application compatible to use with cell phones that do (or do not) support a touchscreen.

SVR: Do you have plans for creating future applications with JavaFX?

RM: Yes, although not in the next couple of months, because I recently finished my education and I'm starting my trainee course here at Caesar Groep in September. But, Caesar Groep and I are really fond of the possibilities of JavaFX and especially in the near future when it becomes more mature and bigger, we are planning to use it for the development of mobile applications.

I personally am planning to deliver my own support to the JavaFX community in the near future. There are already some great additions to JavaFX which are available right now like the JFXtras, MemeFX, and others but none of these are available for developing mobile applications. That's why for my next JavaFX project I'm planning on making some useful ready and easy-to-use mobile components. These components are probably going to have the look and feel of the components that are now used in the iPhone.

SVR: Do you plan to use JavaFX in a school project?

RM: No, since I finished my education just a week ago this isn't possible anymore, but since my presentation at my school about JavaFX went so well I managed to draw the attention of my former tutors to JavaFX. And now because they're also very interested in JavaFX I've almost convinced them to use JavaFX into the school's program.

SVR: Do you know other students that are working with JavaFX?

RM: There are a couple students who also tried out JavaFX, and their first impressions were also close to mine.

SVR: Have you tried out JavaFX 1.2?

RM: Yes, since the first day of the 1.2 release I've managed to migrate my application to the 1.2 version of JavaFX with only small difficulties. After that I was able to use the new layout components like the Tile List for my thumbnails screen. The 1.2 version of the application is also available on my Google Code page of my application.

SVR: If you could add one feature to JavaFX right now, what would that be?

RM: It would be great to see some more ready-to-use components in the near future and it will be even better if these components could also be used for mobile specific applications. For instance some iPhone-ish look alike components would be awesome to have.

SVR: How do you think JavaFX compares to other RIA application development programs? In what ways is JavaFX better?

RM: For my graduation project I was also asked to evaluate JavaFX and compare it with Flex. In my opinion, JavaFX is doing some real good work. For an environment that was officially released about 8 months ago it's still competing against Flex very well; against all odds because Adobe's environment is now almost over 14 years old.

A couple of ways in which JavaFX is better, in my opinion:

- 1) The great ease of development
 - a) Declarative syntax
 - b) Bindings
 - c) JavaFX Script in general
- 2) Small learning curve when you are already familiar with scripting languages or Java
- 3) The possibility of deploying your application to different kinds of platforms (browser, desktop, mobile, etc.)
- 4) The possibility of using all the other Java families
- 5) The work-flow-supporting tools like the Production Suite Plugin

###