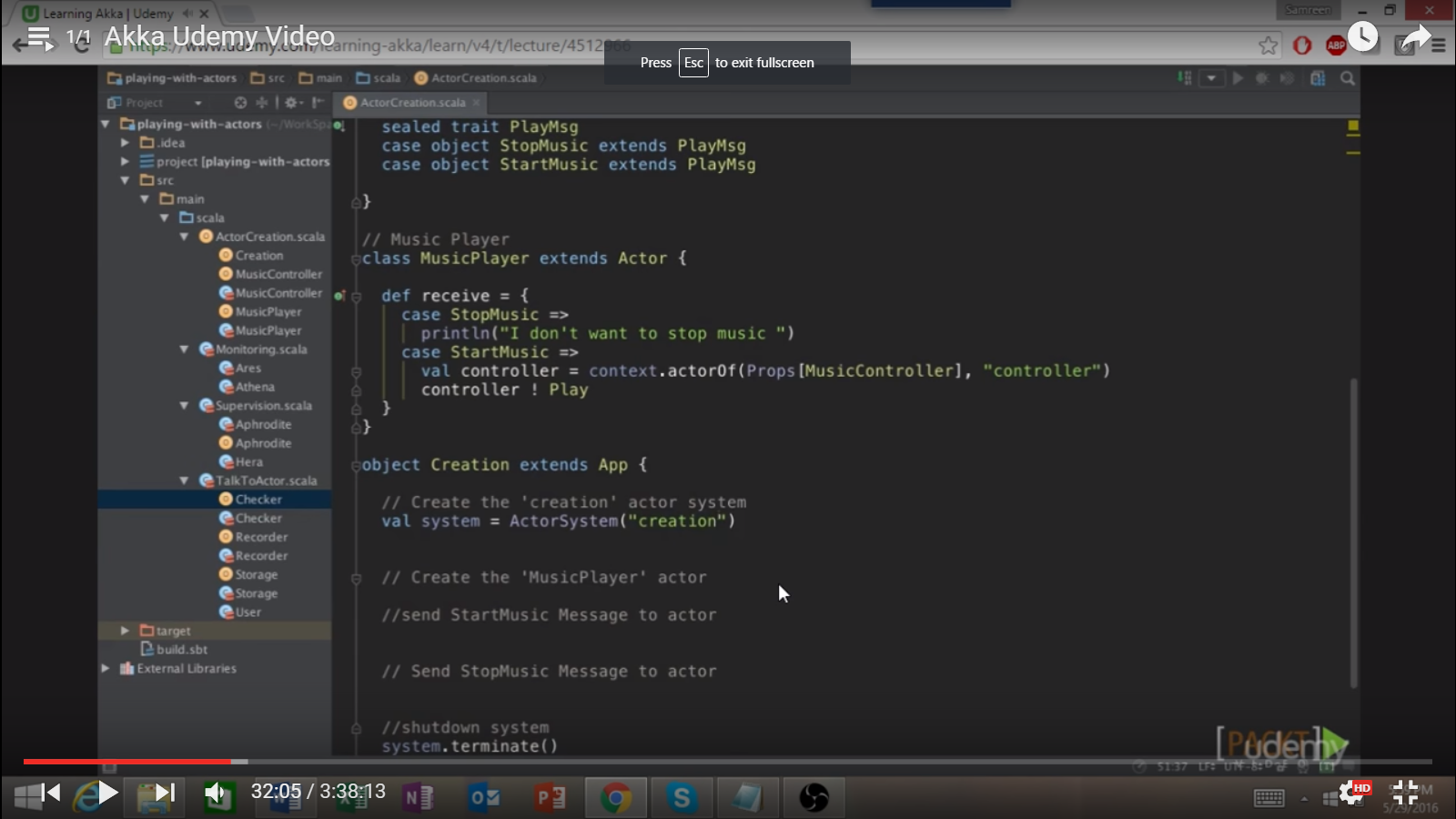
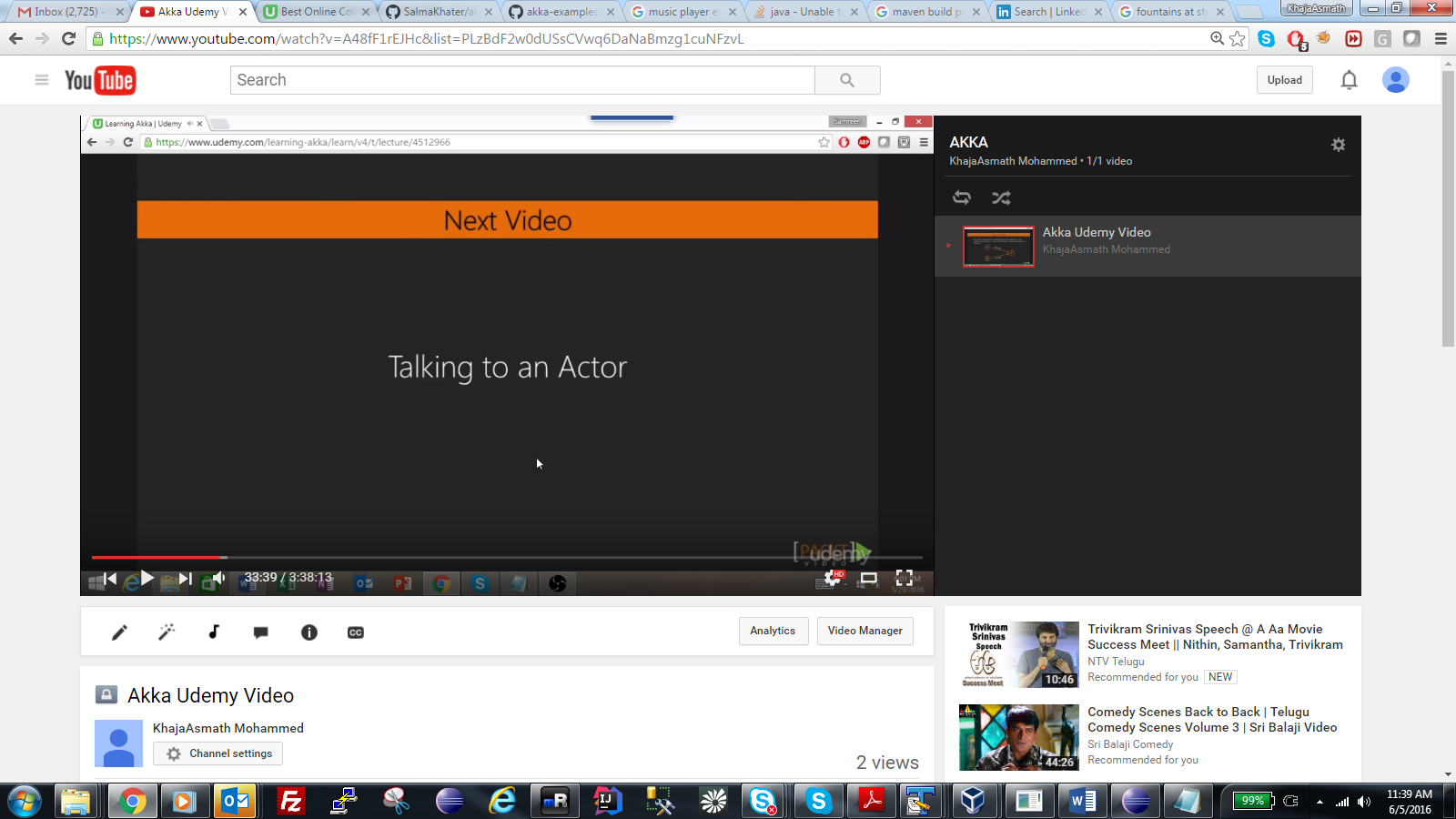


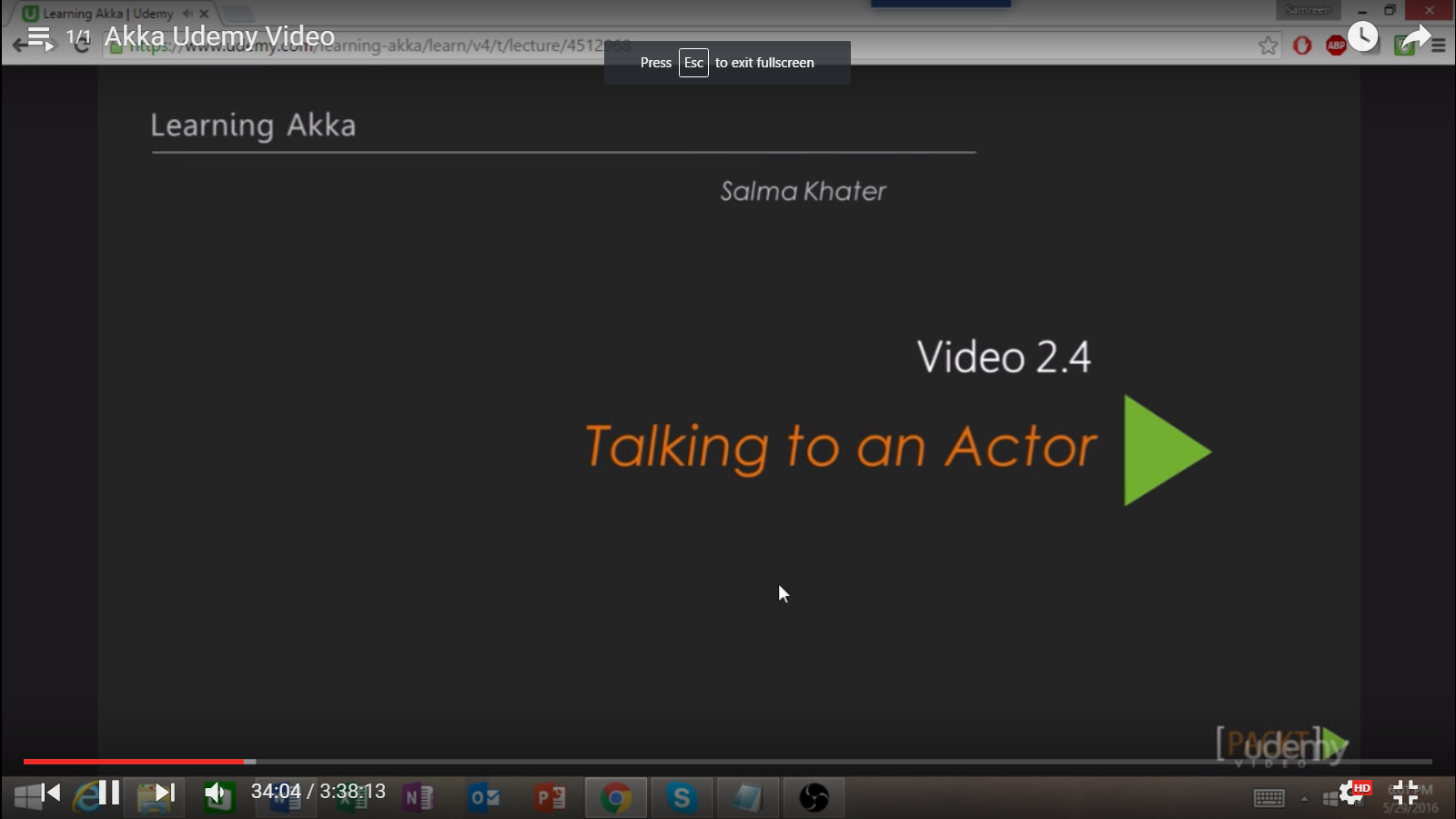
Code base for this is present in <https://github.com/SalmaKhater/akka-examples/blob/master/playing-with-actors/src/main/scala/ActorCreation.scala>

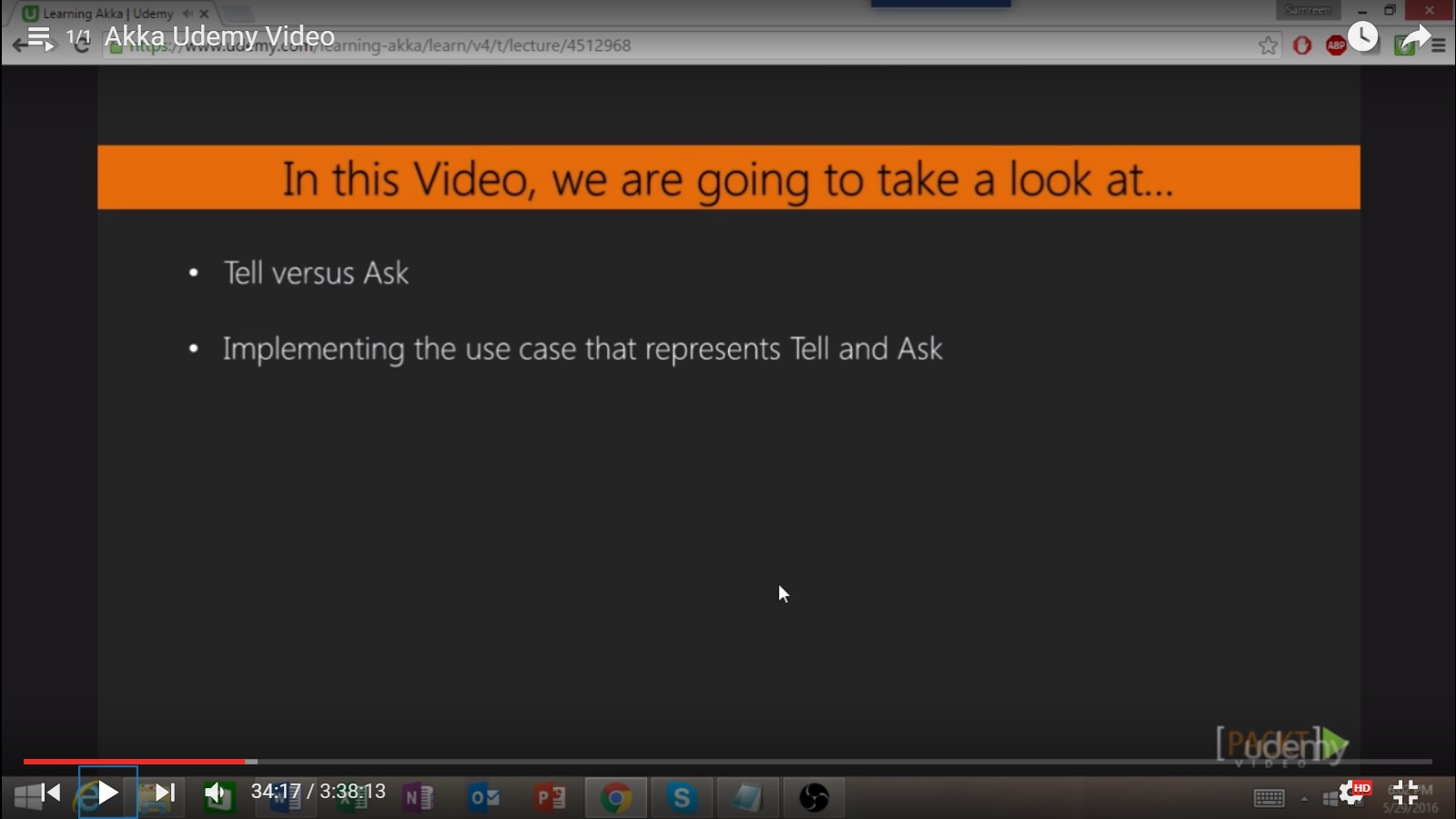
Akka Examples of Salma Khater

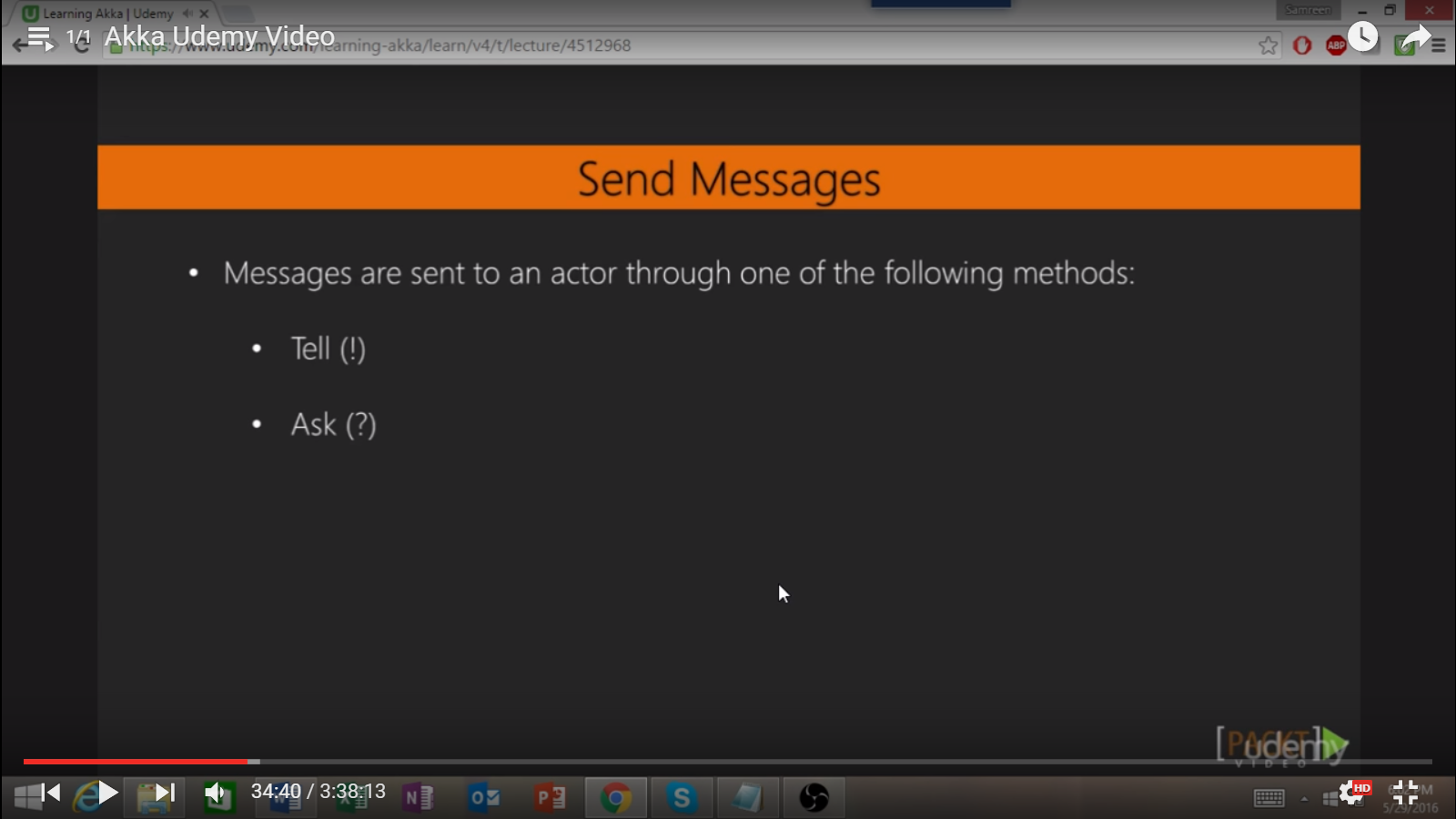


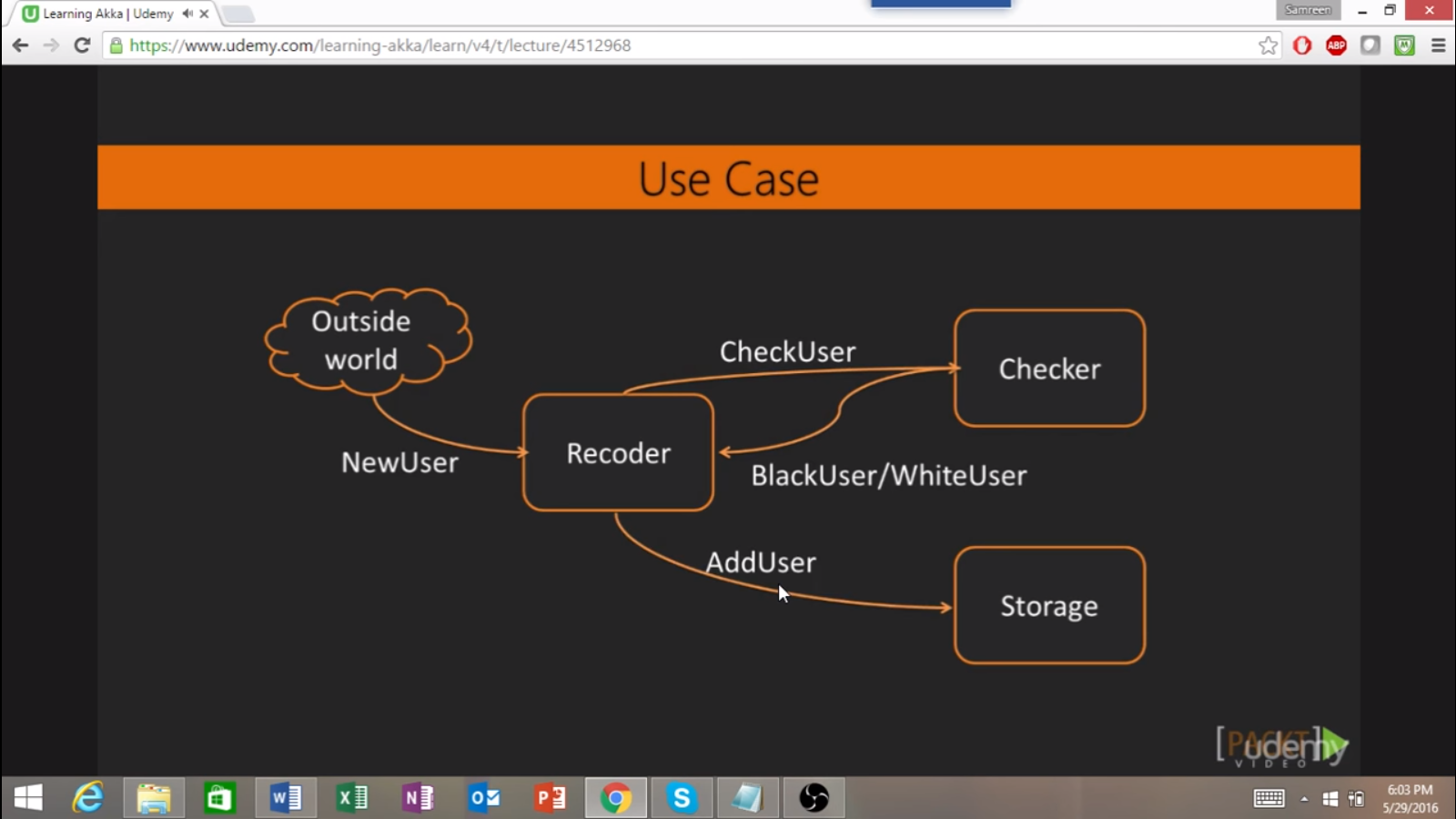
We are creating companion object of music player and music controller.











Three actors here .. Recorder, Checker and Storage

Recorder gets messages from outside world to add new user..

Records talks to checker and see if the user is backlist or not, if not backlist then white user is send. if not then add that user to storage.

In our case, recorder sends deals two types, check if user is present and it not present add that to storage.

**Source Code: SalmaKhater/AkkaExamples.**

1. **Define case class representing username and email.**
2. **Now define, our messages --🡪 New user inside recorder, check user under recorder and add user under storage.**
3. **Now define, responses from the checker like white user and black user. It has separate trait like responses inside the checker.**

**Now Define actors….**

1. **Storage actor: When he receives actor, he adds that to the users list.**

|  |
| --- |
| var users = List.empty[User] |
|  |  |
|  | def receive = { |
|  | case AddUser(user) => |
|  | println(s"Storage: $user added") |
|  | users = user :: users |
|  | } |

1. **Checker actor 🡪 this actor contains list that are backlisted. When checker receives messages to check the user,**

|  |
| --- |
| val blackList = List( |
|  | User("Adam", "adam@mail.com") |
|  | ) |

|  |
| --- |
| def receive = { |
|  | case CheckUser(user) if blackList.contains(user) => |
|  | println(s"Checker: $user in the blacklist") |
|  | sender() ! BlackUser(user) |
|  | case CheckUser(user) => |
|  | println(s"Checker: $user not in the blacklist") |
|  | sender() ! WhiteUser(user) |
|  | } |

**When this actor receives to check user, if checks that user with the backlisted user, if present then it sends that black user object to sender else it will send white user object to the sender.**

1. **Now let’s define Recorder actor, when it receives new user message, recorder actors asks checker actor to send message if that user is backlisted or whitelisted user. Note: if it asking checker not sending the message so we use ? not!**

**? is not terniary operator it is asking message**

**If user is not backlist, we are going to add that user so we are using ! in this case.**

|  |
| --- |
| def receive = { |
|  | case NewUser(user) => |
|  | checker ? CheckUser(user) map { |
|  | case WhiteUser(user) => |
|  | storage ! AddUser(user) |
|  | case BlackUser(user) => |
|  | println(s"Recorder: $user in the blacklist") |
|  | } |
|  | } |

