Olasunkanmi Olayinka – SEC01 (NUID 001512266)

Big Data System Engineering with Scala
Fall 2022
Assignment No. 06

-List of Tasks Implemented

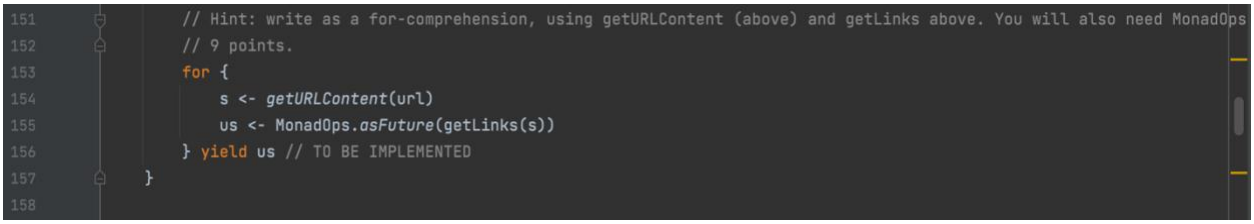
- Implement WebCrawler.scala:
    1. getURLs
    2. for-comprehension using getURLContent
- Implement MonadOps.scala:
    1. asOption

-Code

getURLS:

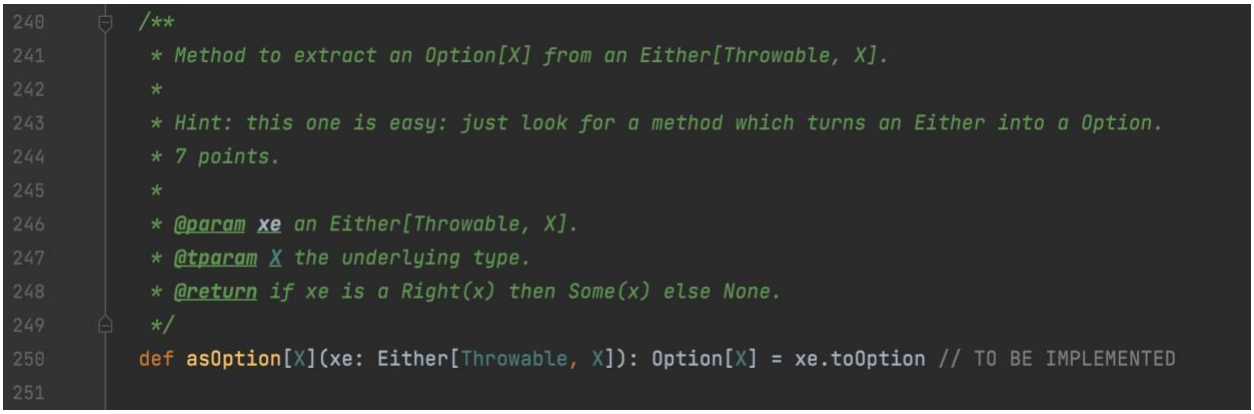
```
140     // Hint: write as a for-comprehension, using the method createURL(Option[URL], String) to get the appropriate URL
141     // 16 points.
142     def getURLs(ns: Node): Seq[Try[URL]] = for {
143         n <- ns \\ "a"
144         href <- n.attribute("href")
145     } yield validateURL(createURL(Some(url), href.text))
146
```

for-comprehension using getURLContent:



```
151     // Hint: write as a for-comprehension, using getURLContent (above) and getLinks above. You will also need MonadOps
152     // 9 points.
153     for {
154         s <- getURLContent(url)
155         us <- MonadOps.asFuture(getLinks(s))
156     } yield us // TO BE IMPLEMENTED
157 }
158
```

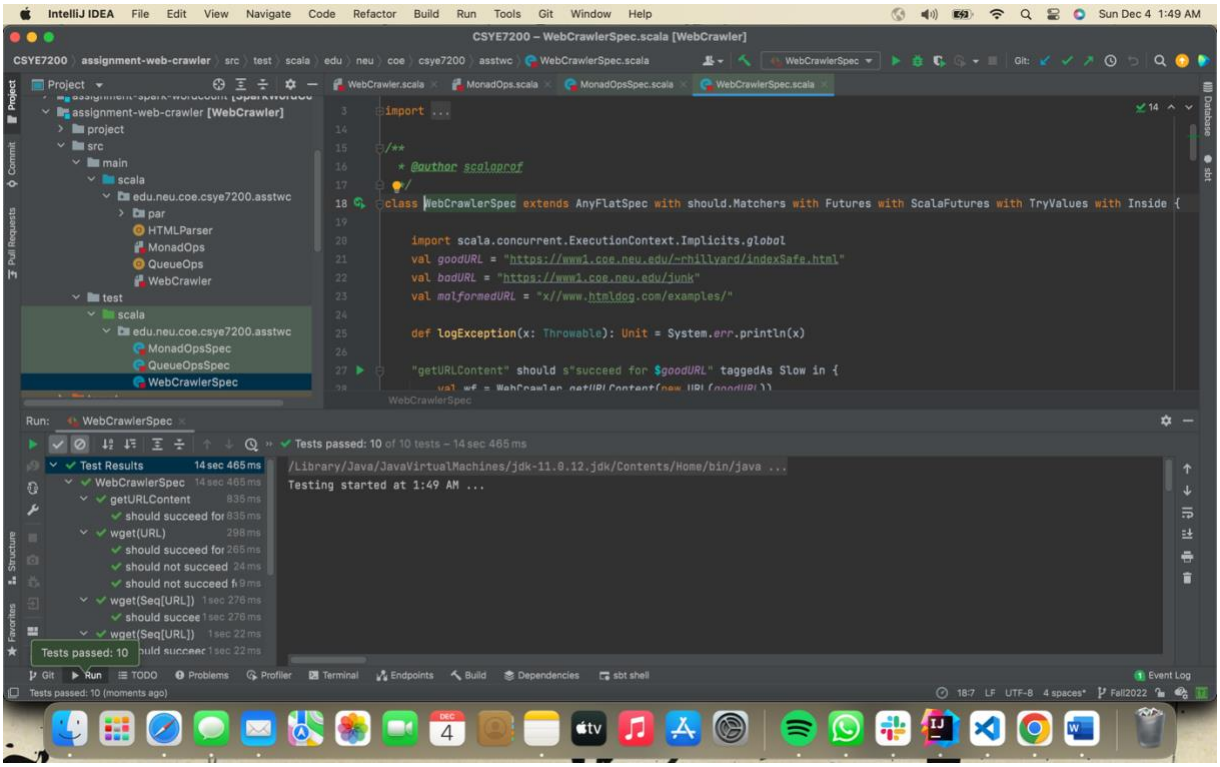
asOption:



```
240 /**
241  * Method to extract an Option[X] from an Either[Throwable, X].
242  *
243  * Hint: this one is easy: just look for a method which turns an Either into a Option.
244  * 7 points.
245  *
246  * @param xe an Either[Throwable, X].
247  * @tparam X the underlying type.
248  * @return if xe is a Right(x) then Some(x) else None.
249  */
250 def asOption[X](xe: Either[Throwable, X]): Option[X] = xe.toOption // TO BE IMPLEMENTED
251
```

-Unit tests

WebCrawlerSpec.scala:



MonadOpsSpec.scala: