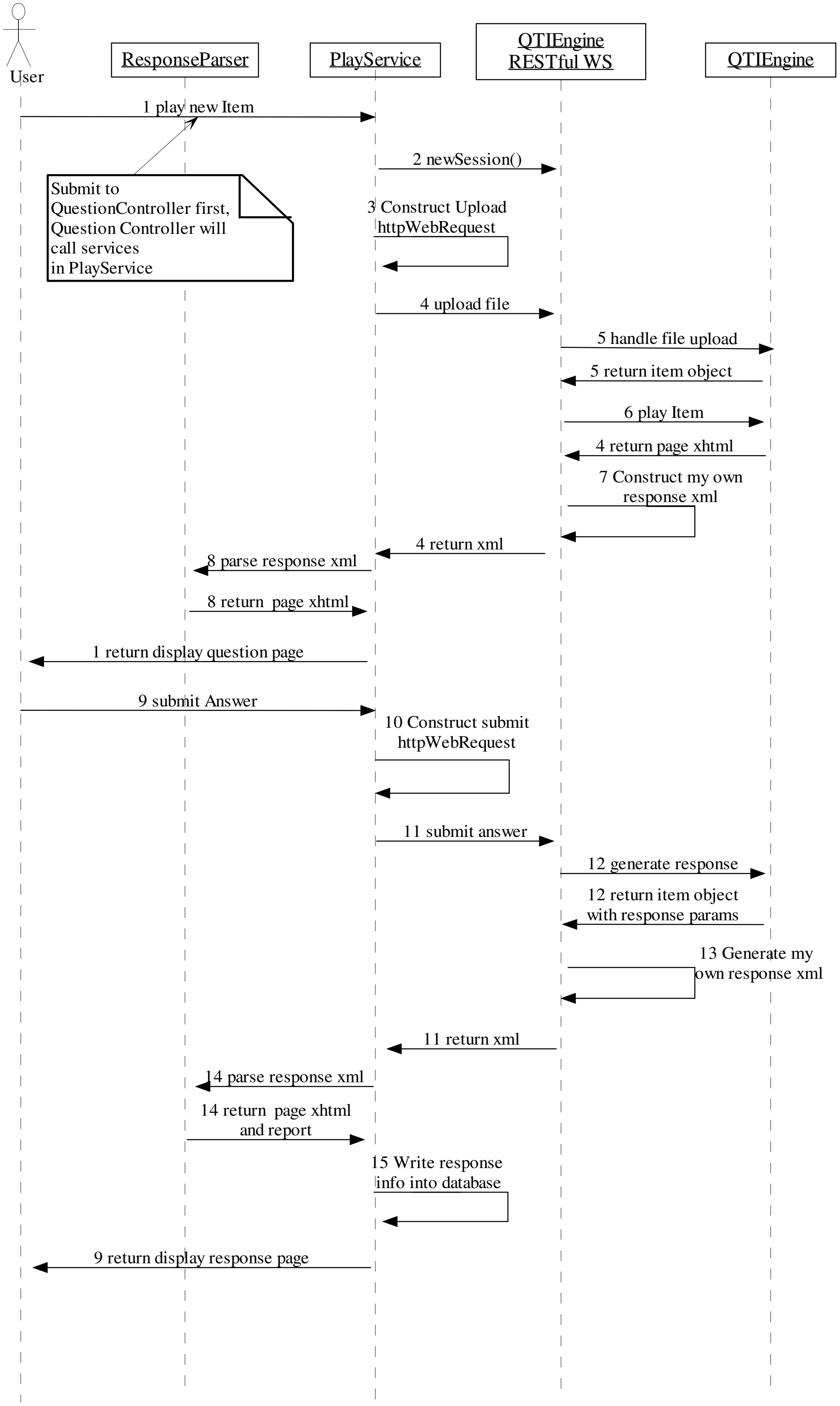
User
ResponseParser
PlayService
QTIEngine RESTful WS
QTIEngine
1 play new Item
Submit to QuestionController first, Question Controller will call services in PlayService
2 newSession()
3 Construct Upload httpWebRequest
4 upload file
5 handle file upload
5 return item object
6 play Item
4 return page xhtml
7 Construct my own response xml
4 return xml
8 parse response xml
8 return page xhtml
1 return display question page
9 submit Answer
10 Construct submit httpWebRequest
11 submit answer
12 generate response
12 return item object with response params
13 Generate my own response xml
11 return xml
14 parse response xml
14 return page xhtml and report
15 Write response info into database
9 return display response page

Explanation

1. http://lslvm-pz3.ecs.soton.ac.uk:81/Question
   User clicks the “play” button for an item. The request will be forward to:
   http://lslvm-pz3.ecs.soton.ac.uk:81/Question/Play/21
   This url means QuestionController, play action (method), 21 is an id parameter. After processing the request, the Play method in QuestionController.cs will call playItem method in PlayService.cs.
2. ~/Services/PlayService.cs, newSession() method calls RestController.groovy, newSession() method API
   See Deployment.docx Section 3 QTIEngine Server for detail.
3. ~/Services/PlayService.cs, `public string Upload(HttpContext context, Question question)`
4. RestController.groovy, upload() method
5. In upload() method in RestController.groovy,
   For item xml, applicationService.handleXml()
   Return an XmlNode presents the item
6. RestController.grooy, playItem() method. This method call Render.renderPage() method in QTIEngine.
   The return is xhtml page as a string
7. playItem() calls RestService.createNormalResponse() method to create the response xml. But this response only contains the page xhtml. The response variables(SCORE, numAttempt, etc) are not included.
8. QuestionController.cs, `private ActionResult parseResponse(string responseString,int id)`
   This method uses services of ResponseParser.cs
9. User click the submit button to submit the answer.
10. PlayService.cs, `public string playItem(HttpContext context, Item item)`
11. RestController.grooy, playItem() method. This time, this method will execute, assessmentItem.setResponses() and assessmentItem.processResponses()
12. The same as 6
13. The same as 7, but in the return, response variables are included, see playItem.xml for example.
14. The same as 8.
15. In PlayService.cs, `public void SaveItemResult(XDocument xDoc, Item item, DB db)`