Design Document for caArray

Sequence and Class Diagrams

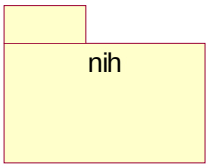
Issue <1.2>

Revision History

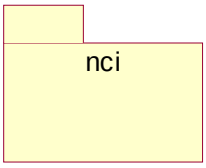
Date	Issue	Description	Author
01/02/2004	1.1	Initial version	Carleen Dickerson
3/26/2004	1.2	Printed for final review	Sumeet Muju

Model Name: caArray.mdl

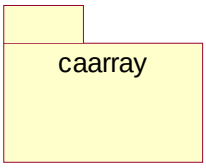
1. Logical View::gov



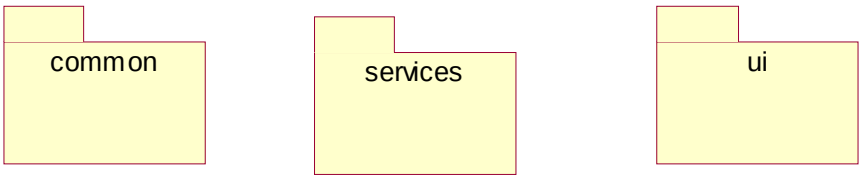
2. Logical View::gov::nih



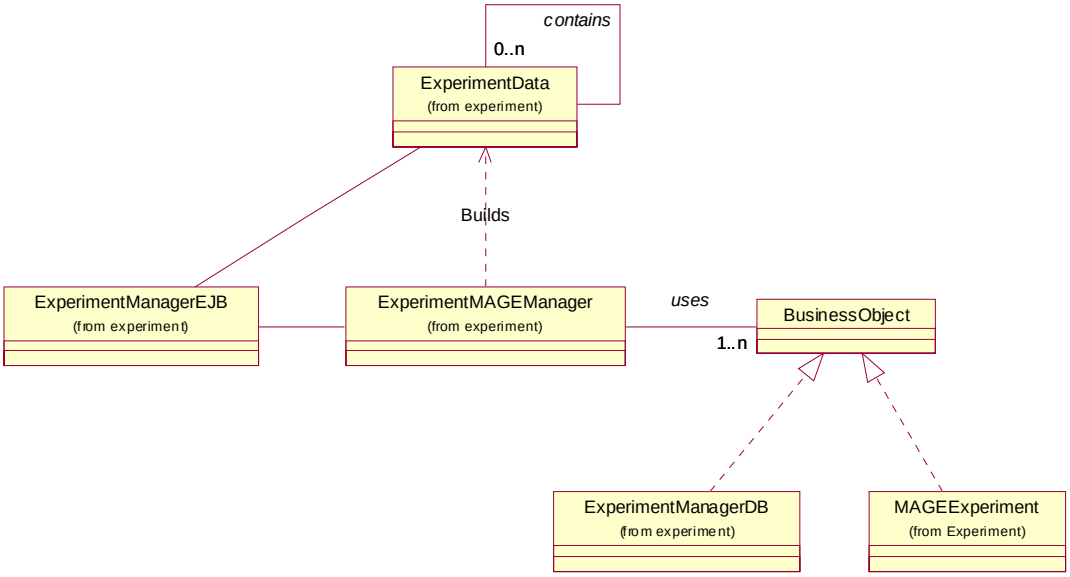
3. Logical View::gov::nih::nci



4. Logical View::gov::nih::nci::caarray



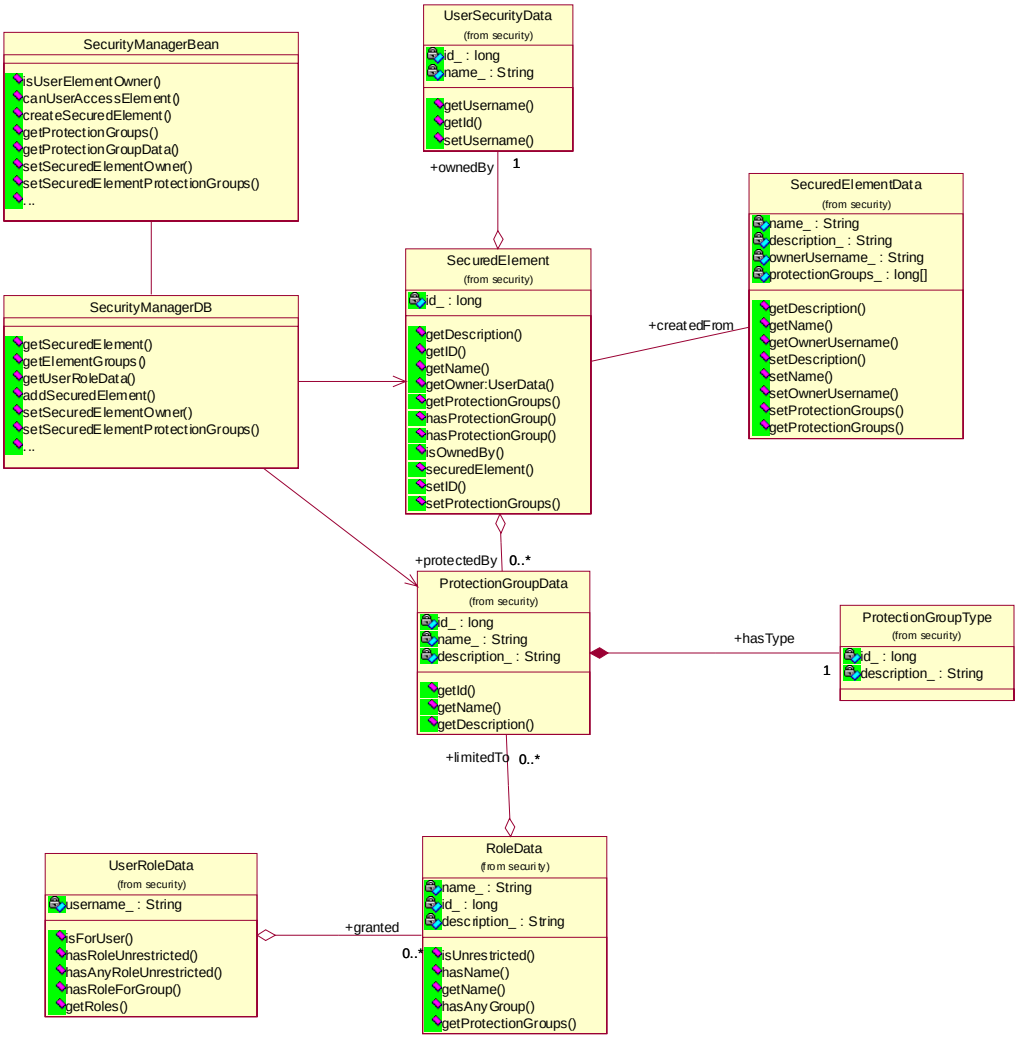
5. Logical View::gov::nih::nci::caarray::services



5.1 BusinessObject

5.2

6. Logical View::gov::nih::nci::caarray::services::security



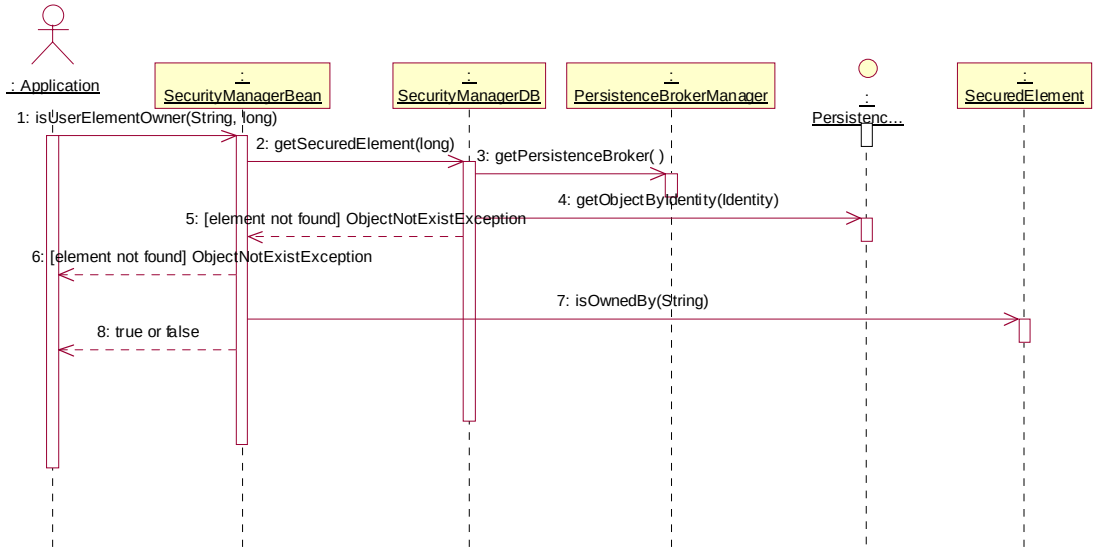
This diagram shows the classes that are used to perform security management in the caArray system.

6.1 SecurityManagerBeanThe SecurityManagerBean is the EJB that is utilized by other portions of the system to perform run time security checks to see if a user is authorized to perform operations on a particular secured data element. It is also used to control the visibility of secured elements and to create new secured elements.

6.2 SecurityManagerDBThe SecurityManagerDB class encapsulates all logic used to interact with the system database.

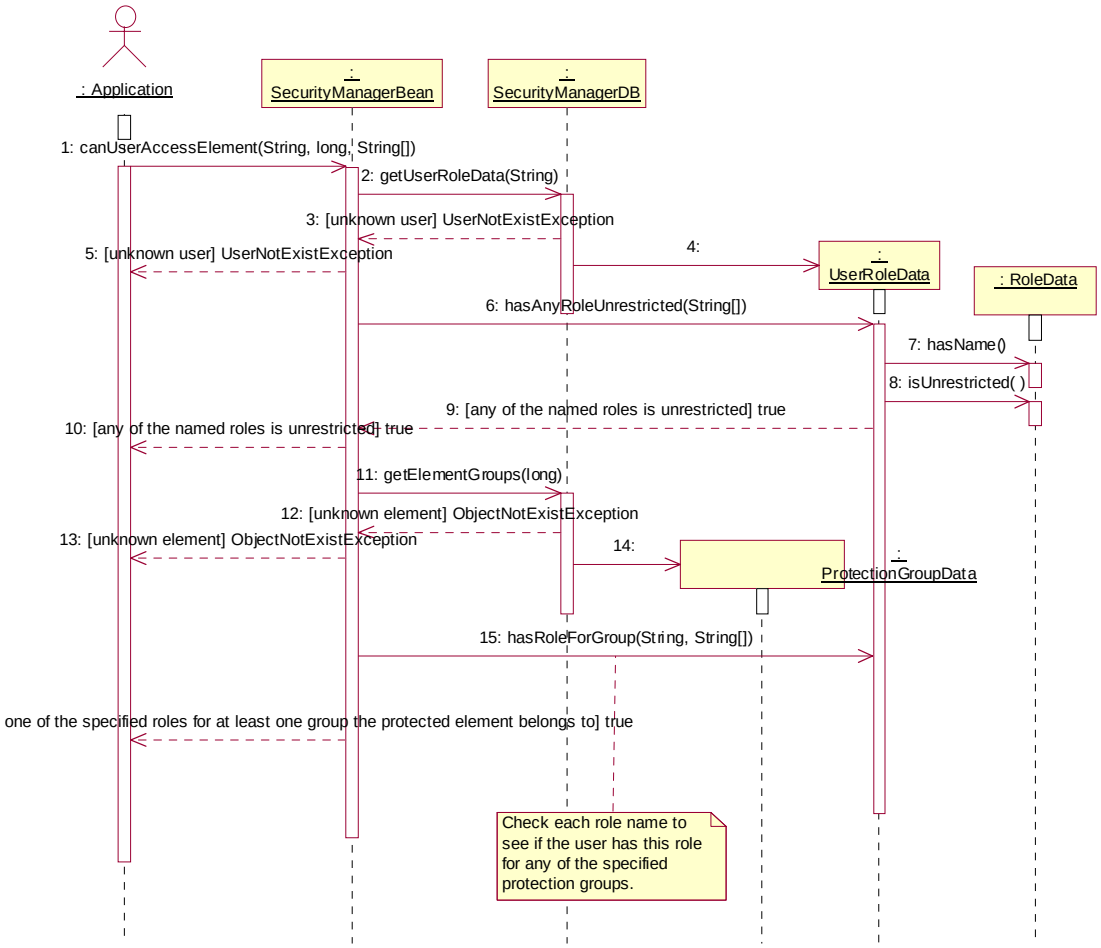
6.3

7. SecurityManagerBean.isUserElementOwner



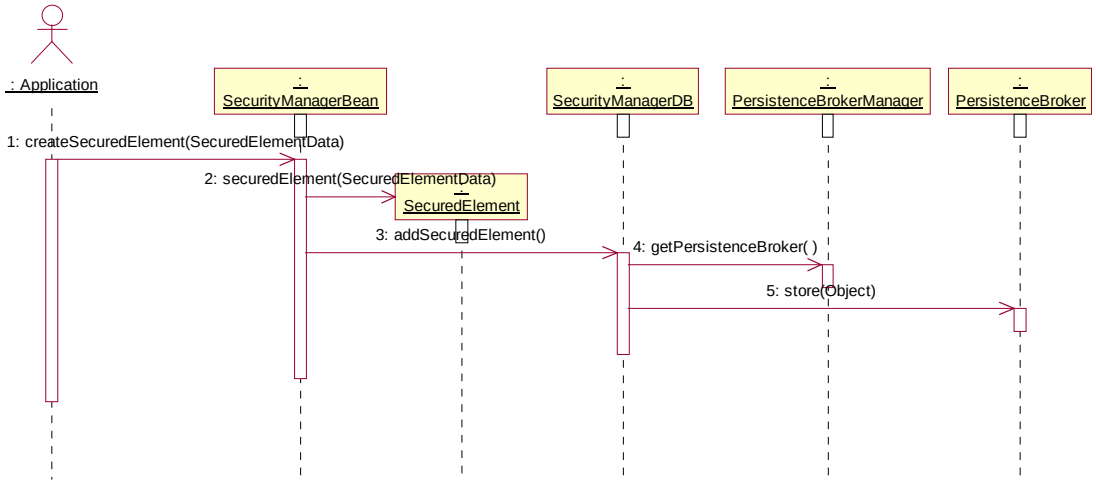
This diagram shows the processing performed to determine if a particular user is the owner of a secured element. First the SecuredElement object is retrieved from the database. Then the secured element is checked to see if it is owned by the specified user.

8. SecurityManagerBean.canUserAccessElement



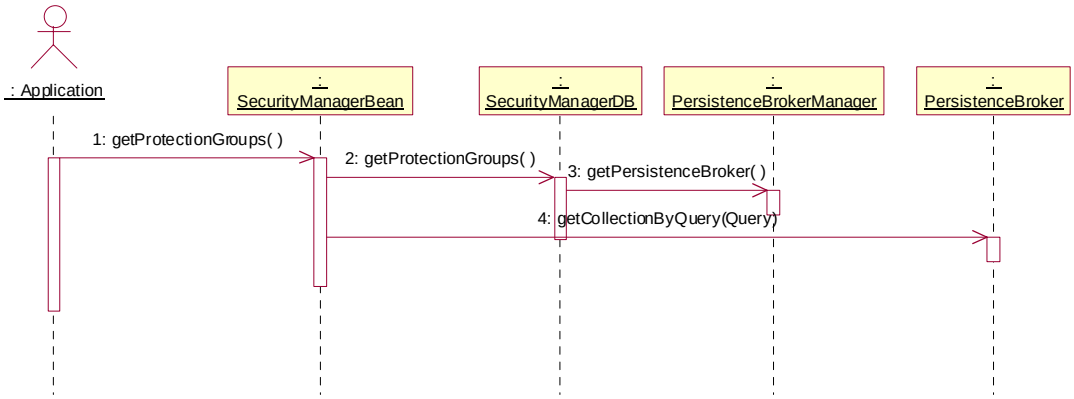
This diagram shows the processing involved in determining if a user can access a particular secured element under one or more roles. The role names are used to determine what type of operation the user wants to perform on the element. The SecurityManagerBean will first retrieve all role data for the user being checked. Then it will check if this user has any of the specified roles without group restriction. If so, the user will be granted access. If not, then another call will be made to the database to determine what protection groups the secured element belongs to. Then each of the roles is checked to see if the user has that role for any of the groups the element belongs to. If the user has any role for any of the groups the element belongs to, access will be granted, otherwise access will be denied.

9. SecurityManagerBean.createSecuredElement



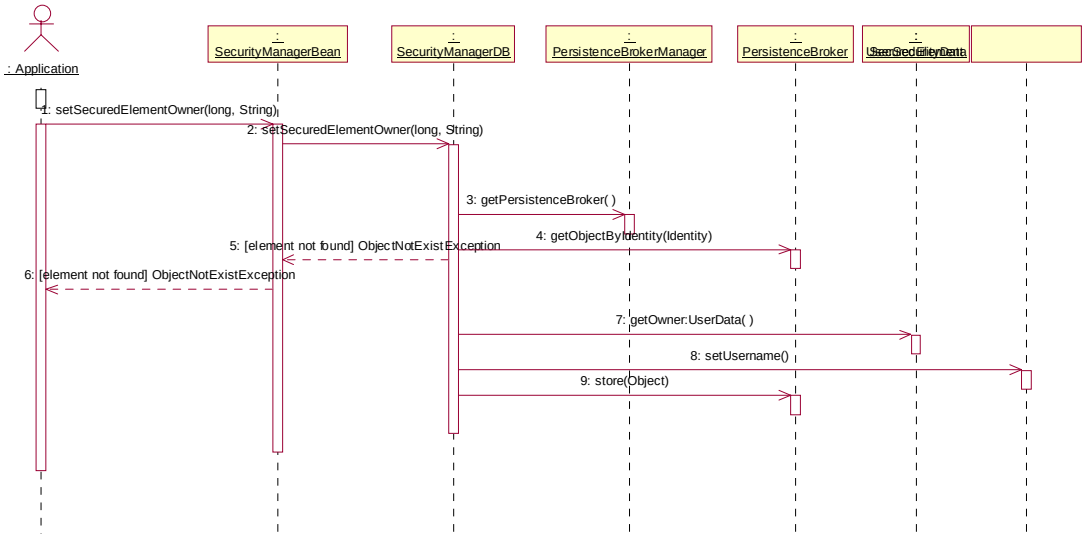
This diagram shows how a new SecuredElement is added to the system. A system component (such as the ExperimentManager) will create a new SecuredElementData object describing the new element (such as a new experiment) and will then call createSecuredElement(). The SecurityManagerBean will create a new SecuredElement object containing the passed data and will then store it in the database.

10. SecurityManagerBean.getProtectionGroups



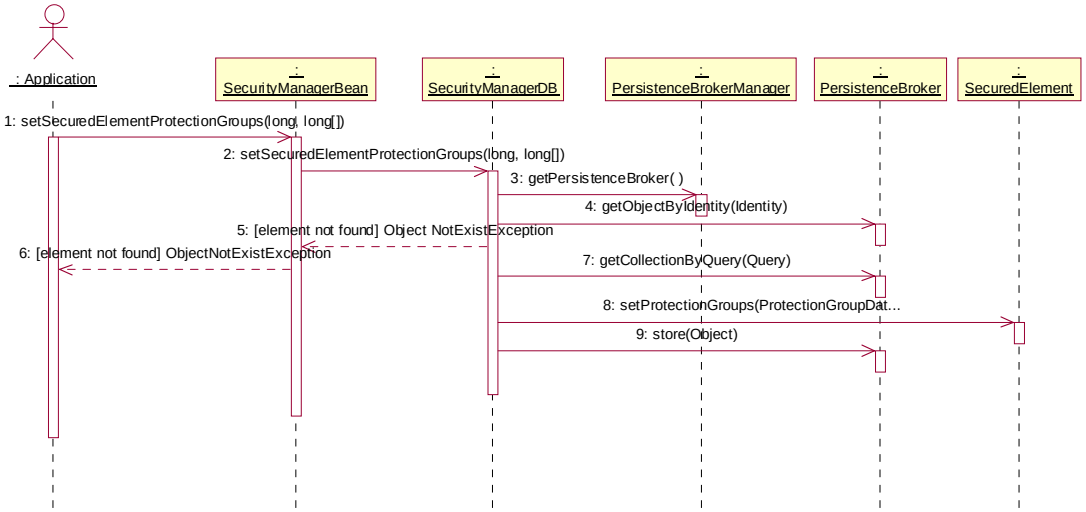
This method shows how the list of all ProtectionGroups in the system is retrieved. The SecurityManager delegates this call to the SecurityManagerDB which utilized the OJB PersistenceBroker to retrieve the objects from the database.

11. SecurityManagerBean.setSecuredElementOwner



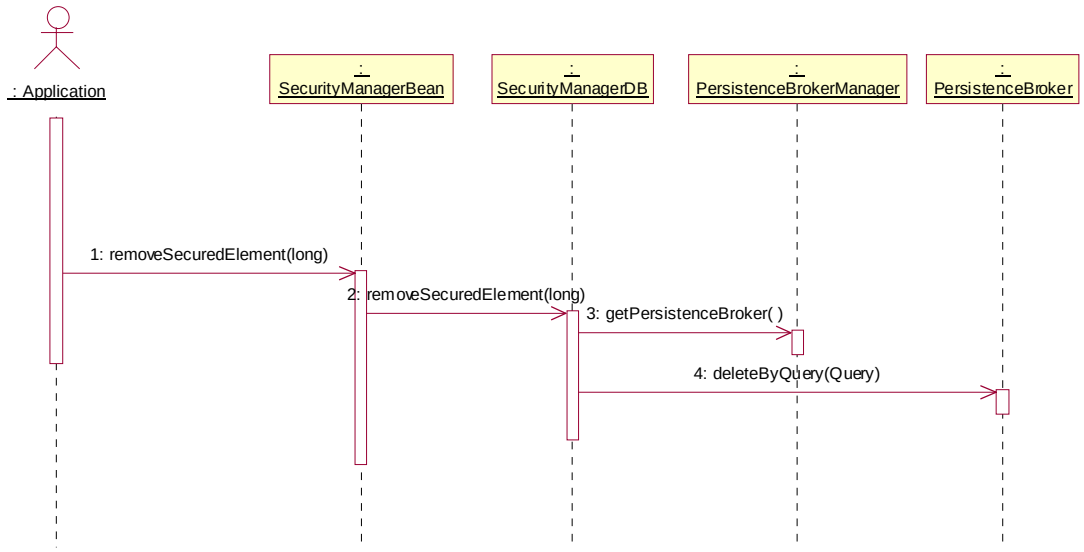
This diagram shows the processing performed when a call is made to change the owner of a secured element. The SecurityManagerBean delegates this call to the SecurityManagerDB class which utilizes the OJB PersistenceBroker to get the secured element from the database. If the secured element cannot be found in the database the caller will receive an exception indicating that the object does not exist. Otherwise the username of the owning UserData for the SecuredElement will be set to the specified username and the OJB PersistenceBroker will be used to store the object back into the database.

12. SecurityManagerBean.setSecuredElementProtectionGroups

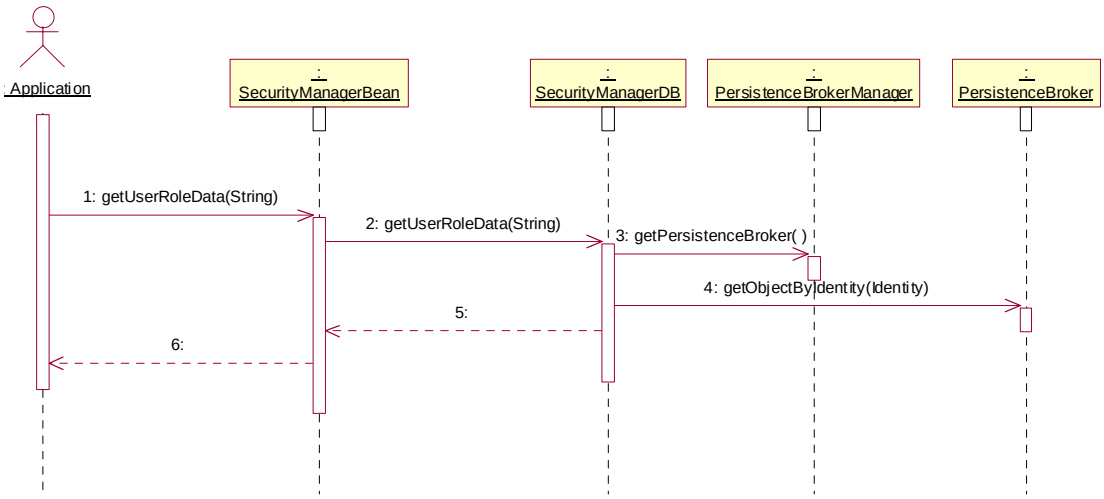


This diagram shows the processing performed when a call is made to change the visibility of a secured element. The SecurityManagerBean delegates this call to the SecurityManagerDB class which utilizes the OJB PersistenceBroker to get the secured element from the database. If the secured element cannot be found in the database the caller will receive an exception indicating that the object does not exist. Otherwise the specified collection of protection groups will also be retrieved from the database. The SecuredElement setProtectionGroups() method will be invoked to set its protection groups to the new collection of groups, and the OJB PersistenceBroker will be utilized to store the object back into the database.

13. SecurityManagerBean.removeSecuredElement



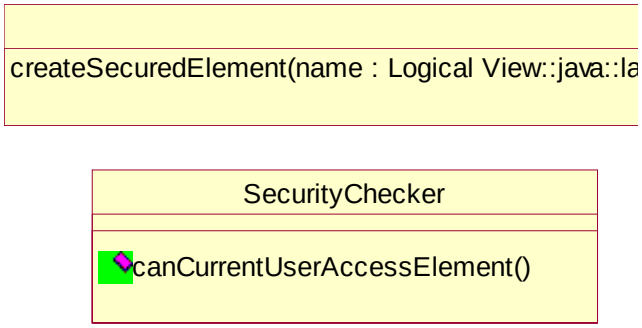
14. SecurityManagerBean.getUserRoleData



This diagram shows the processing performed to retrieve the user role data for a particular user.

15. Logical View::gov::nih::nci::caarray::services::util

16. Logical View::gov::nih::nci::caarray::services::util::security

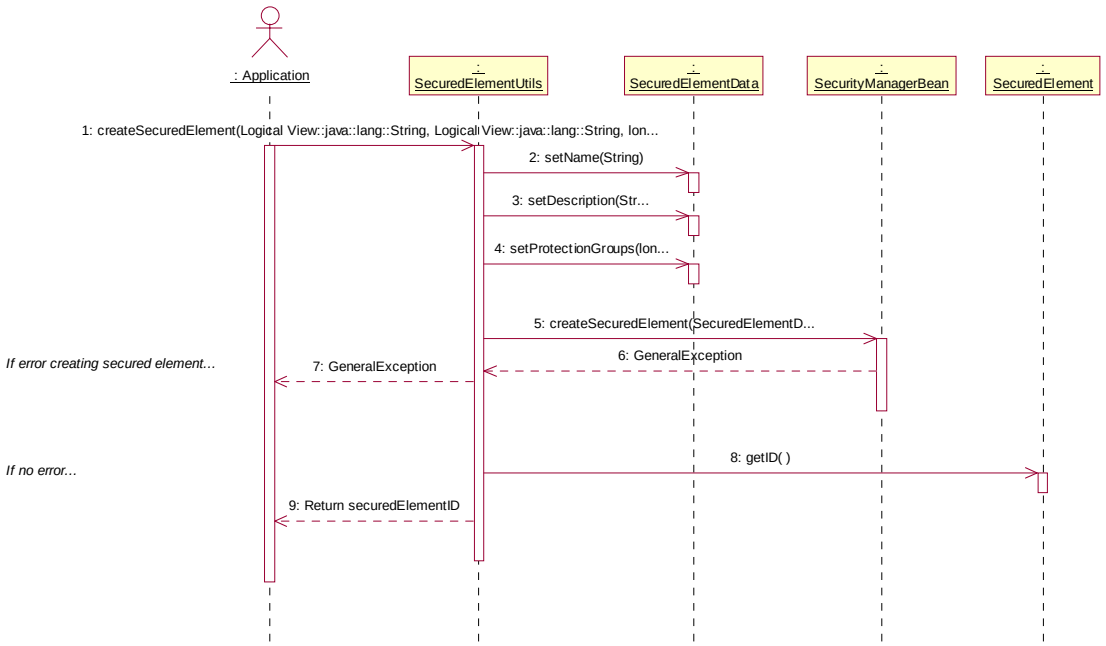


16.1 SecurityChecker

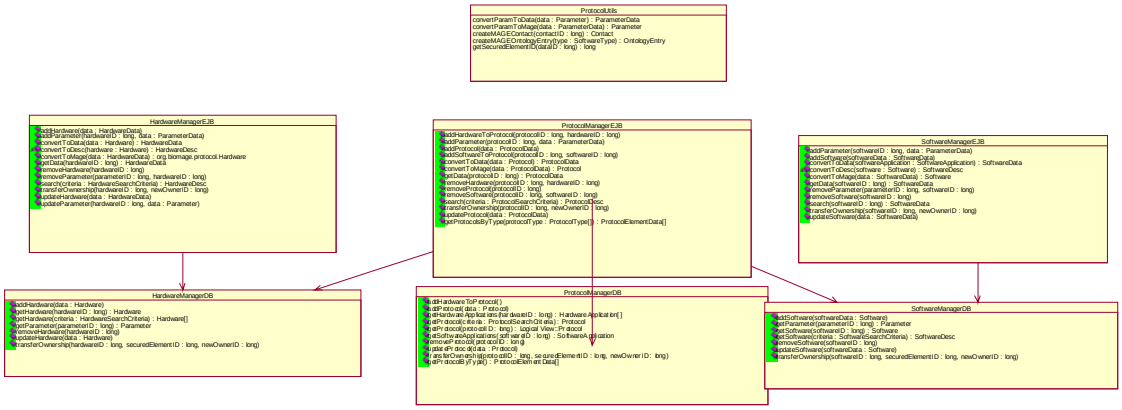
16.2 SecuredElementUtils

16.3

17. SecuredElementUtils.createSecuredElement



18. Logical View::gov::nih::nci::caarray::services::protocol


ProtocolUtils
convertParamToData(data : Parameter) : ParameterData
convertParamToMage(data : ParameterData) : Parameter
createMAGEContact(contactID : long) : Contact
createMAGEOntologyEntry(type : SoftwareType) : OntologyEntry
getSecuredElementID(dataID : long) : long
HardwareManagerEJB
addHardware(data : HardwareData)
addParameter(hardwareID : long, data : ParameterData)
convertToData(data : Hardware) : HardwareData
convertToDesc(hardware : Hardware) : HardwareDesc
convertToMage(data : HardwareData) : org.biomage.protocol.Hardware
getData(hardwareID : long) : HardwareData
removeHardware(hardwareID : long)
removeParameter(parameterID : long, hardwareID : long)
search(criteria : HardwareSearchCriteria) : HardwareDesc
transferOwnership(hardwareID : long, newOwnerID : long)
updateHardware(data : HardwareData)
updateParameter(hardwareID : long, data : Parameter)
ProtocolManagerEJB
addHardwareToProtocol(protocolID : long, hardwareID : long)
addParameter(protocolID : long, data : ParameterData)
addProtocol(data : ProtocolData)
addSoftwareToProtocol(protocolID : long, softwareID : long)
convertToData(data : Protocol) : ProtocolData
convertToMage(data : ProtocolData) : Protocol
getData(protocolID : long) : ProtocolData
removeHardware(protocolID : long, hardwareID : long)
removeProtocol(protocolID : long)
removeSoftware(protocolID : long, softwareID : long)
search(criteria : ProtocolSearchCriteria) : ProtocolDesc
transferOwnership(protocolID : long, newOwnerID : long)
updateProtocol(data : ProtocolData)
getProtocolsByType(protocolType : ProtocolType[]) : ProtocolElementData[]
SoftwareManagerEJB
addParameter(softwareID : long, data : ParameterData)
addSoftware(softwareData : SoftwareData)
convertToData(softwareApplication : SoftwareApplication) : SoftwareData
convertToDesc(software : Software) : SoftwareDesc
convertToMage(data : SoftwareData) : Software
getData(softwareID : long) : SoftwareData
removeParameter(parameterID : long, softwareID : long)
removeSoftware(softwareID : long)
search(softwareID : long) : SoftwareData
transferOwnership(softwareID : long, newOwnerID : long)
updateSoftware(data : SoftwareData)
HardwareManagerDB
addHardware(data : Hardware)
getHardware(hardwareID : long) : Hardware
getHardware(criteria : HardwareSearchCriteria) : Hardware[]
getParameter(parameterID : long) : Parameter
removeHardware(hardwareID : long)
updateHardware(data : Hardware)
transferOwnership(hardwareID : long, securedElementID : long, newOwnerID : long)
ProtocolManagerDB
addHardwareToProtocol()
addProtocol(data : Protocol)
getHardwareApplication(hardwareID : long) : HardwareApplication[]
getProtocol(criteria : ProtocolSearchCriteria) : Protocol
getProtocol(protocolID : long) : LogicalView::Protocol
getSoftwareApplication(softwareID : long) : SoftwareApplication
removeProtocol(protocolID : long)
updateProtocol(data : Protocol)
transferOwnership(protocolID : long, securedElementID : long, newOwnerID : long)
getProtocolsByType() : ProtocolElementData[]
SoftwareManagerDB
addSoftware(softwareData : Software)
getParameter(parameterID : long) : Parameter
getSoftware(softwareID : long) : Software
getSoftware(criteria : SoftwareSearchCriteria) : SoftwareDesc
removeSoftware(softwareID : long)
updateSoftware(softwareData : Software)
transferOwnership(softwareID : long, securedElementID : long, newOwnerID : long)

18.1 HardwareManagerDB

18.2 ProtocolManagerDBProtocolManagerDB is responsible for retrieving all protocol-related data from the database.

18.3 SoftwareManagerDB

18.4 HardwareManagerEJB

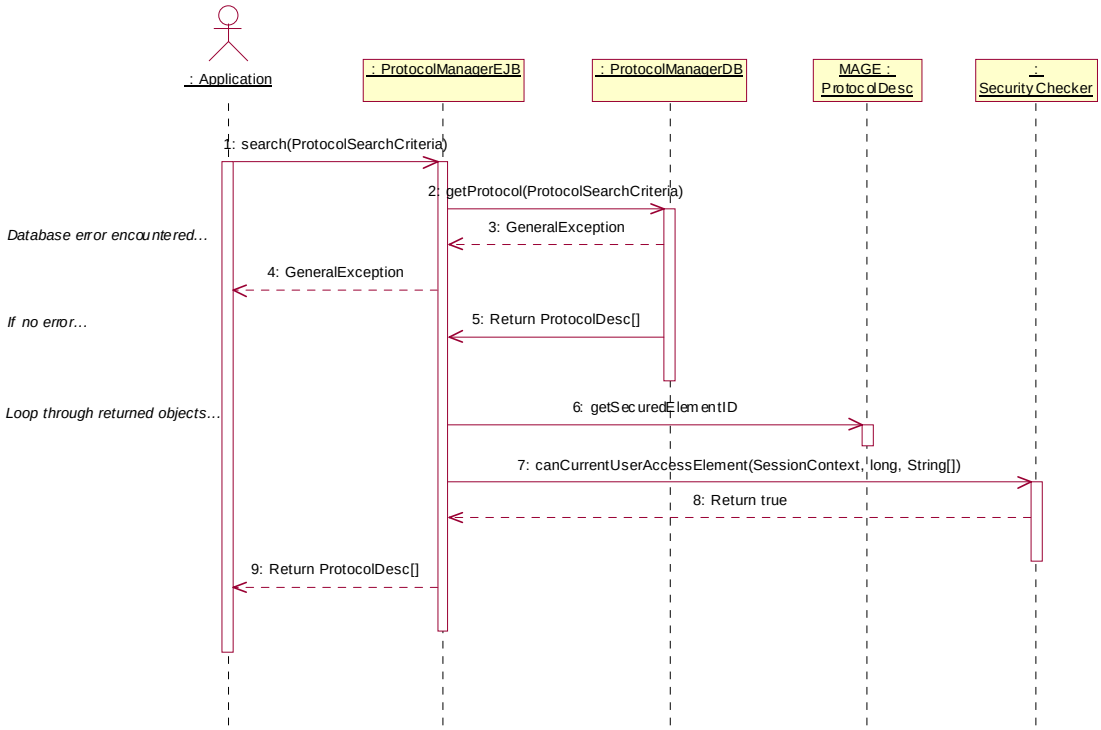
18.5 ProtocolManagerEJB

18.6 SoftwareManagerEJB

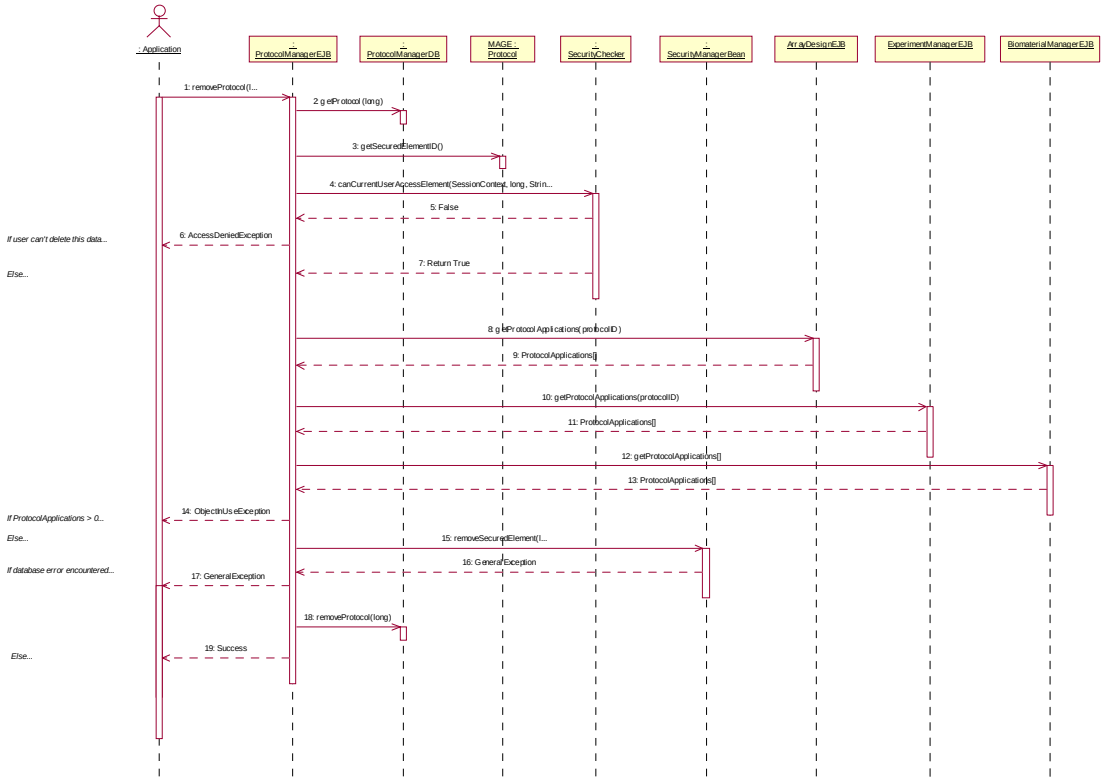
18.7 ProtocolUtils

18.8

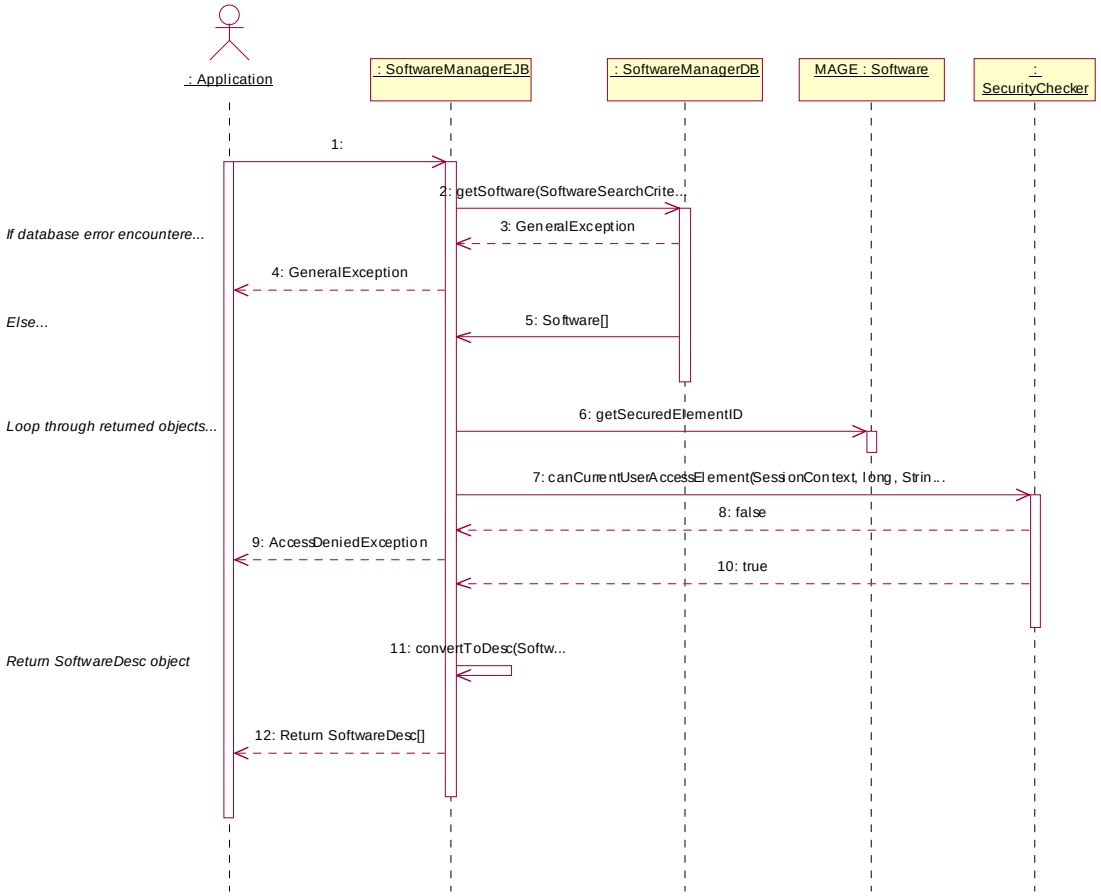
19. ProtocolManager.getProtocol



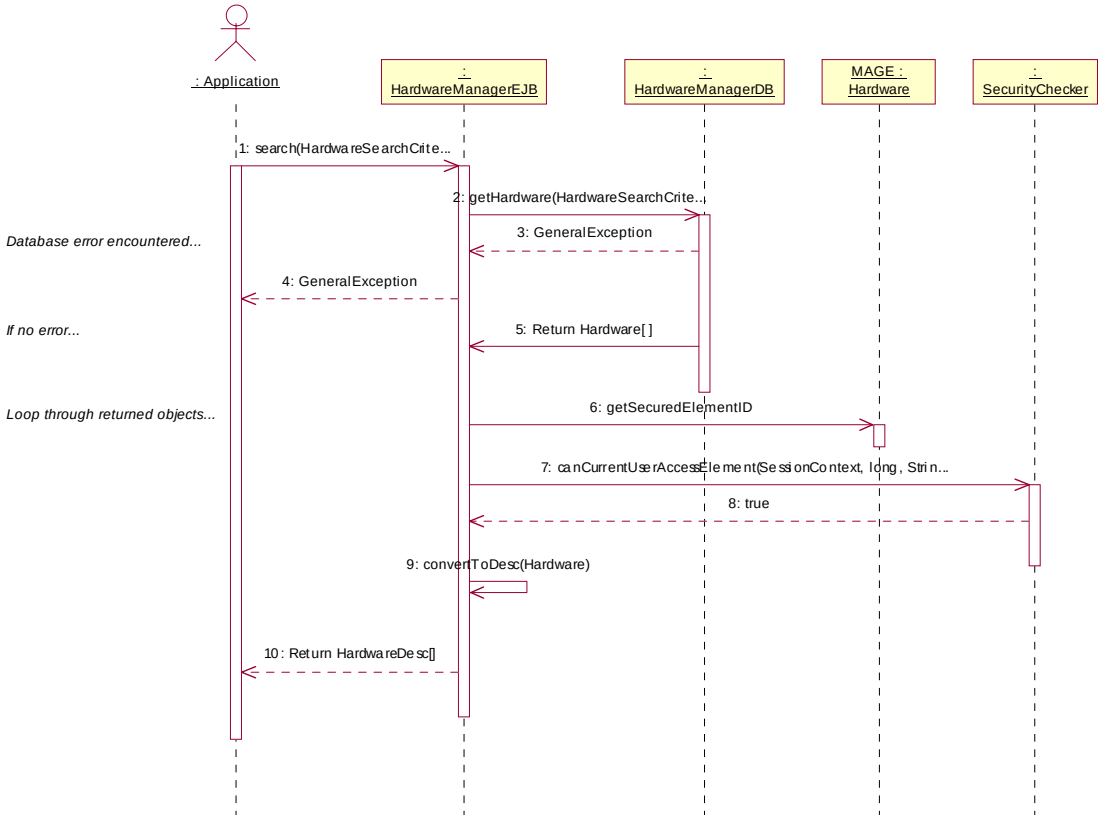
20. ProtocolManager.removeProtocol



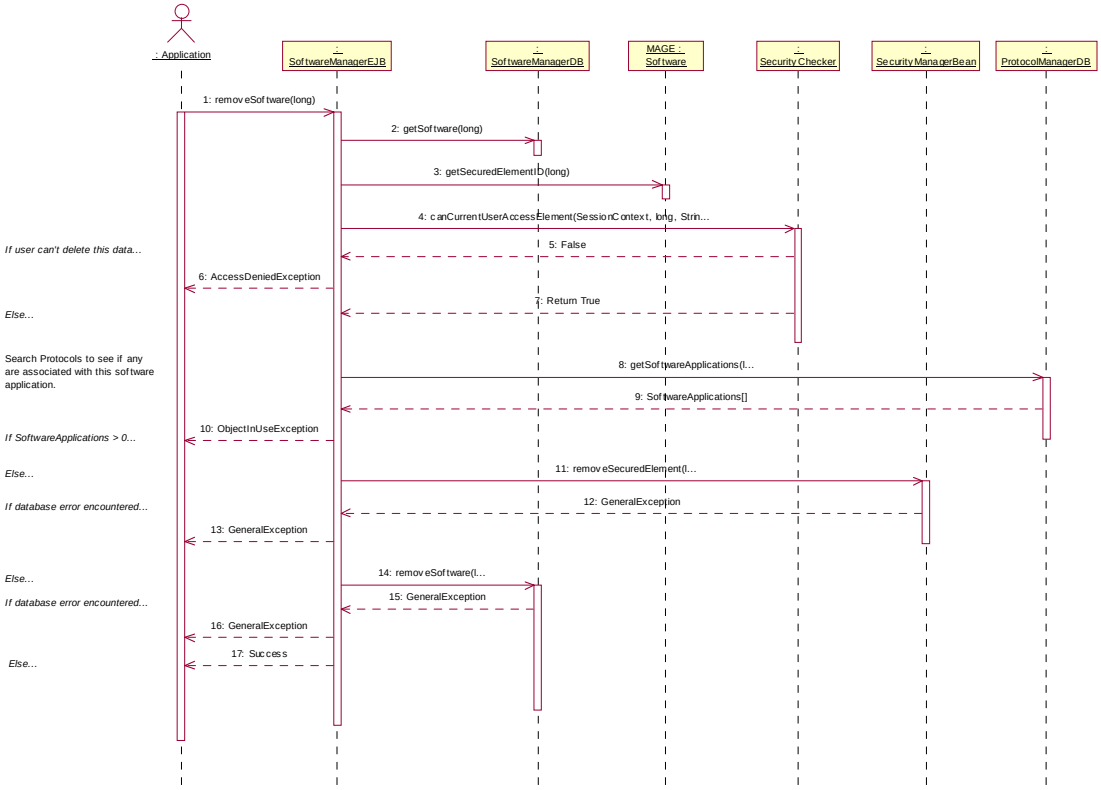
21. SoftwareManager.getSoftware



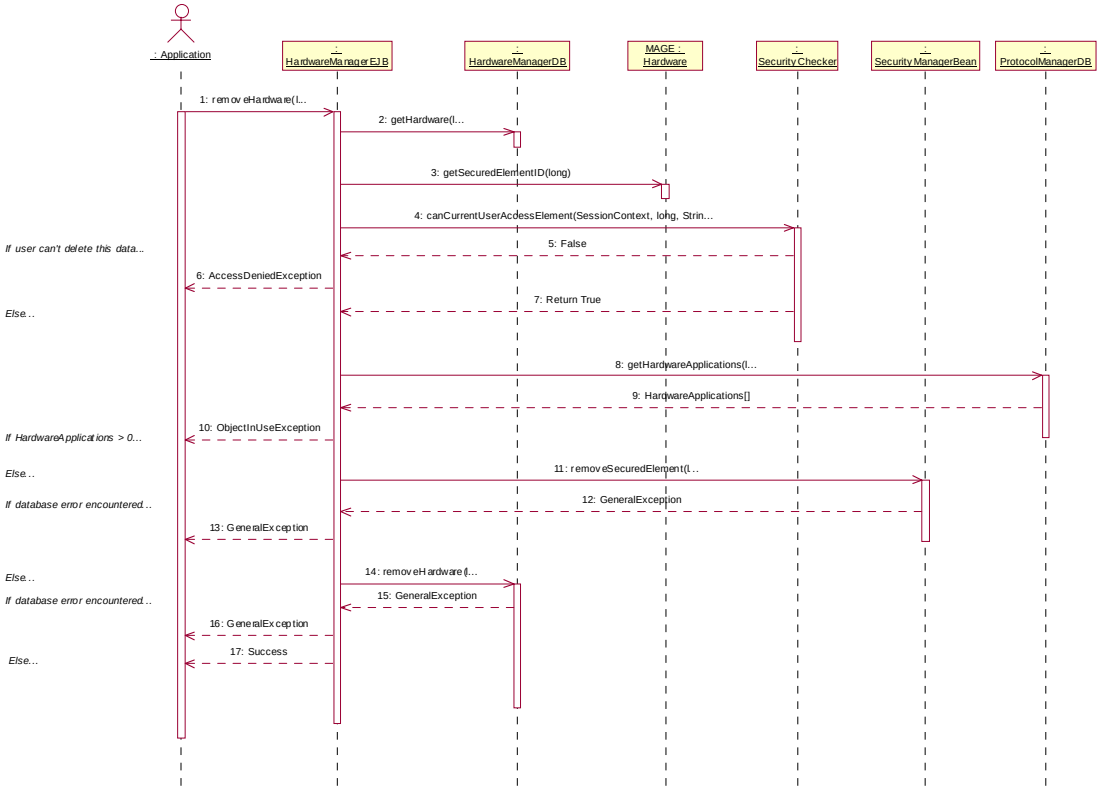
22. HardwareManager.getHardware



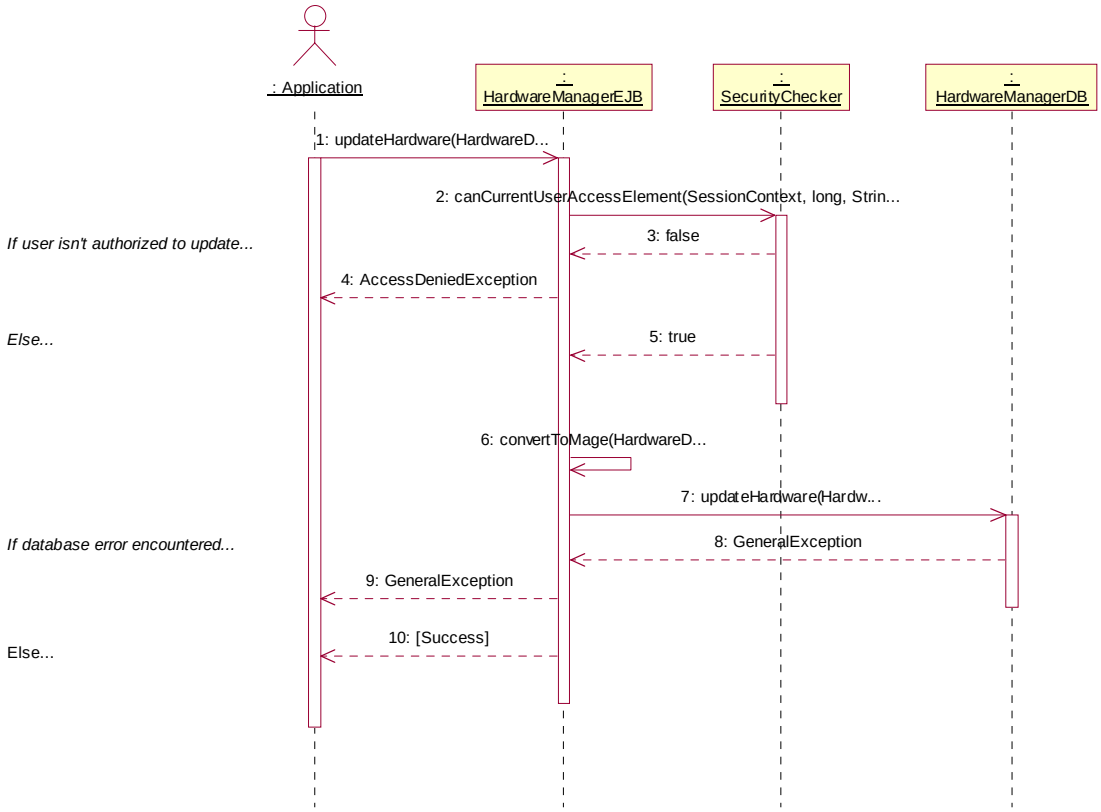
23. SoftwareManager.removeSoftware



24. HardwareManager.removeHardware

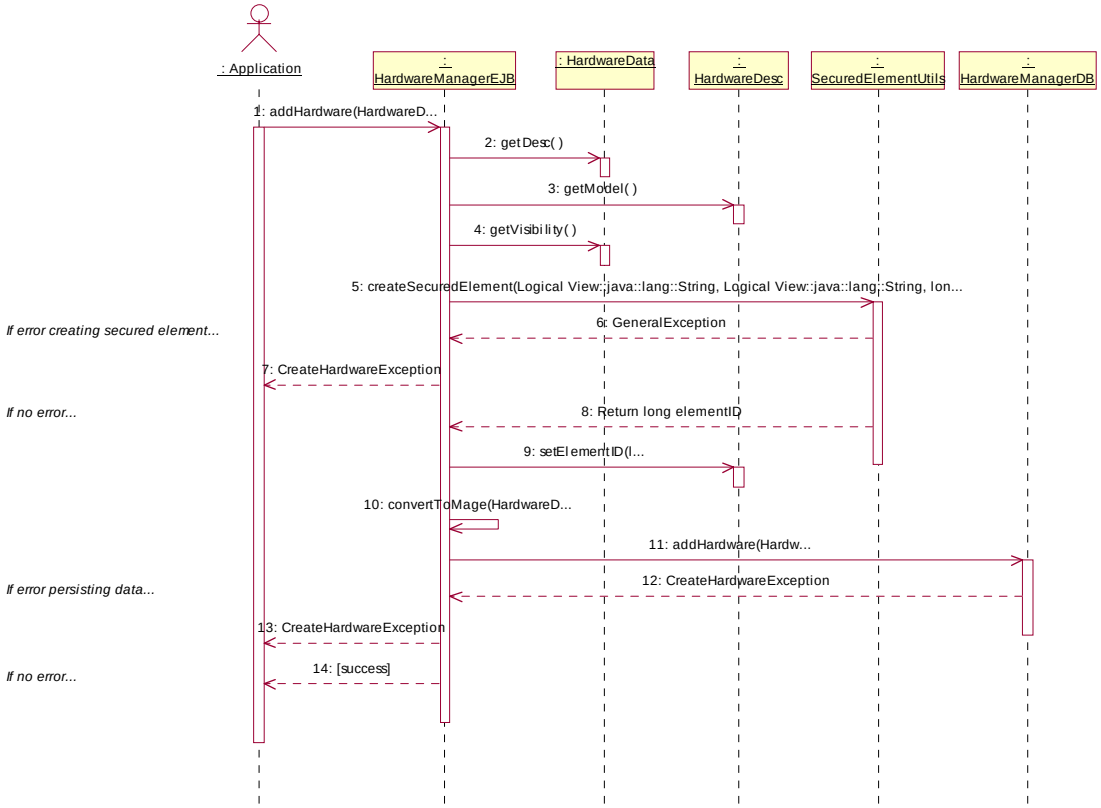


25. HardwareManager.updateHardware

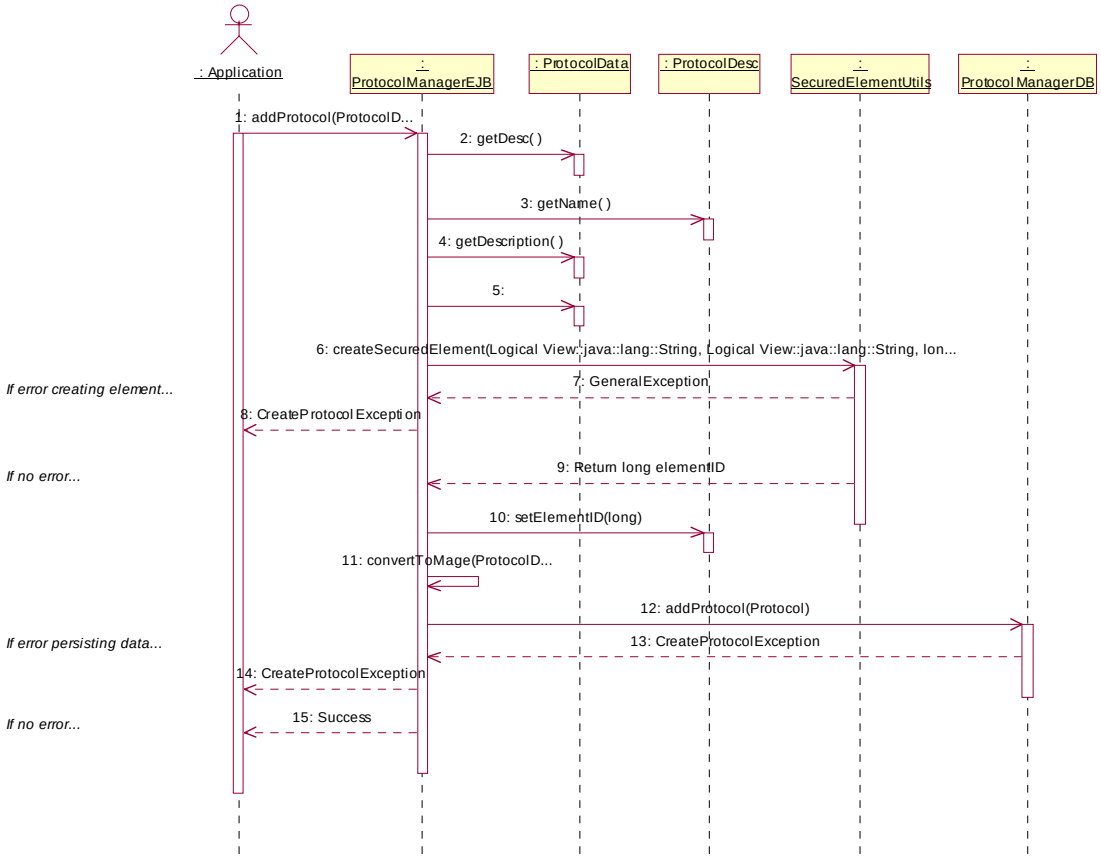


This sequence shows the processing involved to update an existing piece of hardware. If the user has UPDATE privileges, they will be allowed to invoke the updateHardware method. The EJB then checks to verify that the user has UPDATE privileges for this particular hardware. If they do not, the EJB throws an AccessDeniedException. If they do, the incoming data is transformed to a MAGEstk Hardware object and given to the DB class to persist in the database. If any error occurs during persistence, a general exception is thrown.

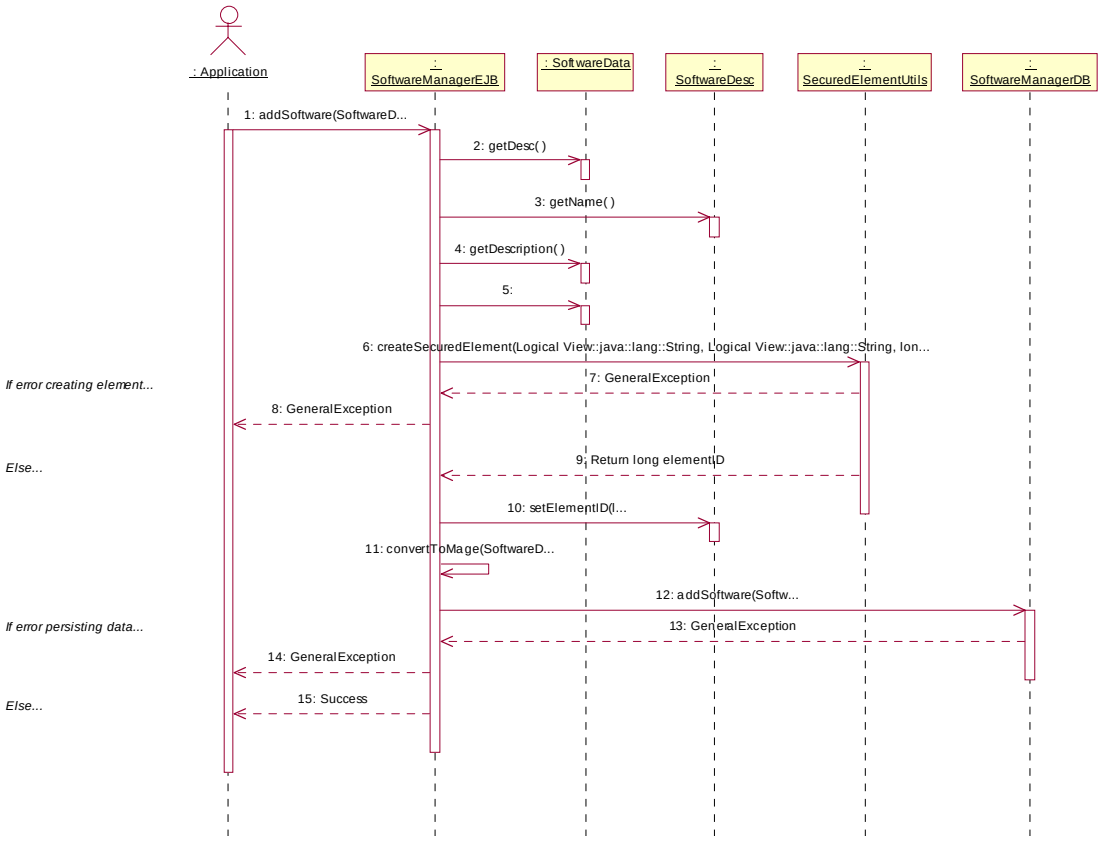
26. HardwareManager.addHardware



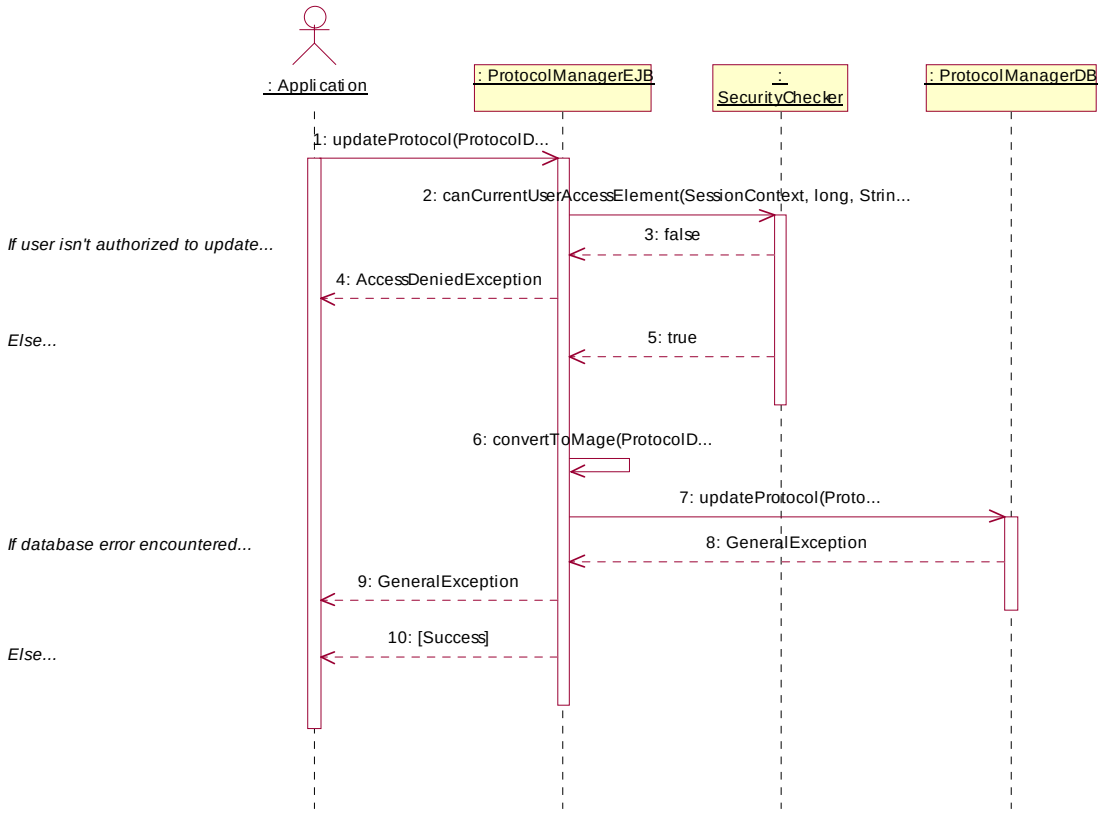
27. ProtocolManager.addProtocol



28. SoftwareManager.addSoftware

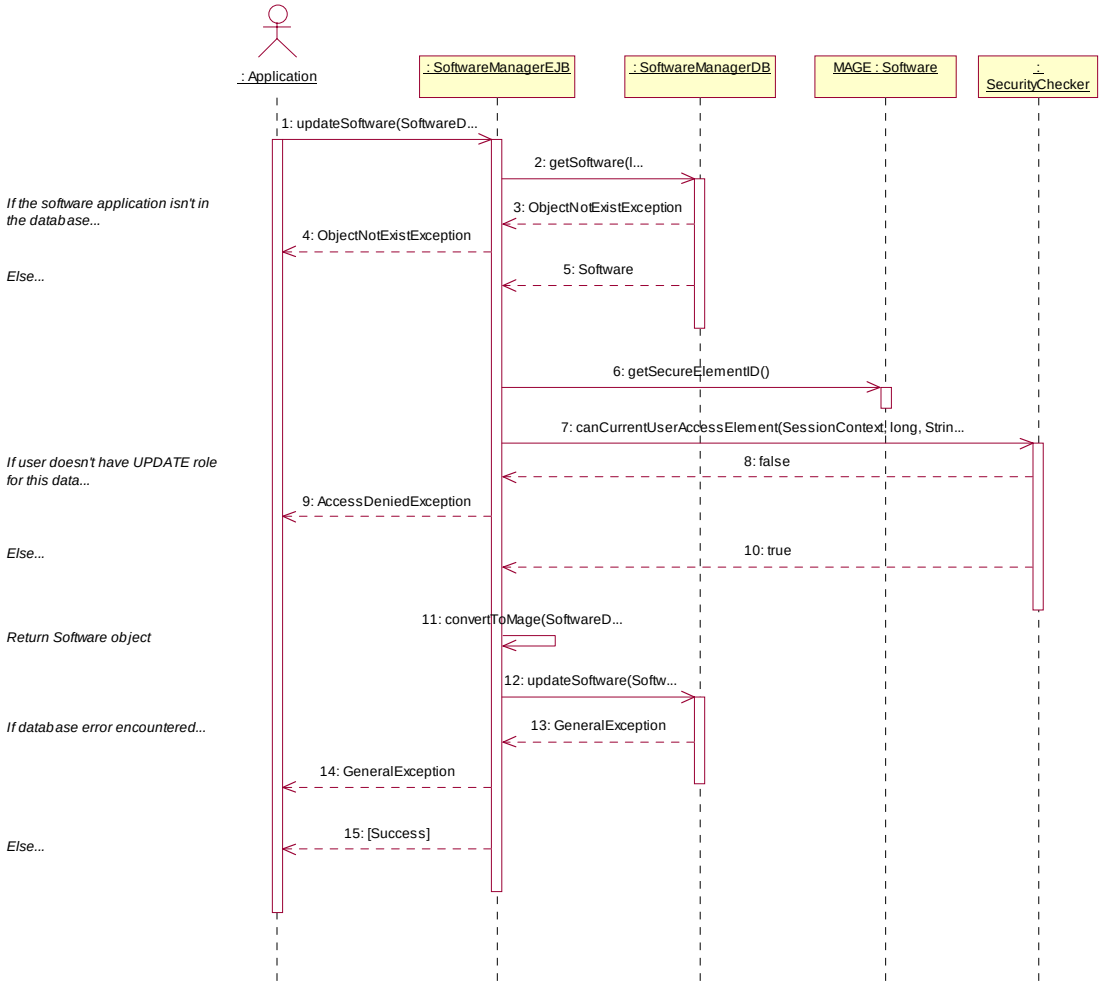


29. ProtocolManager.updateProtocol



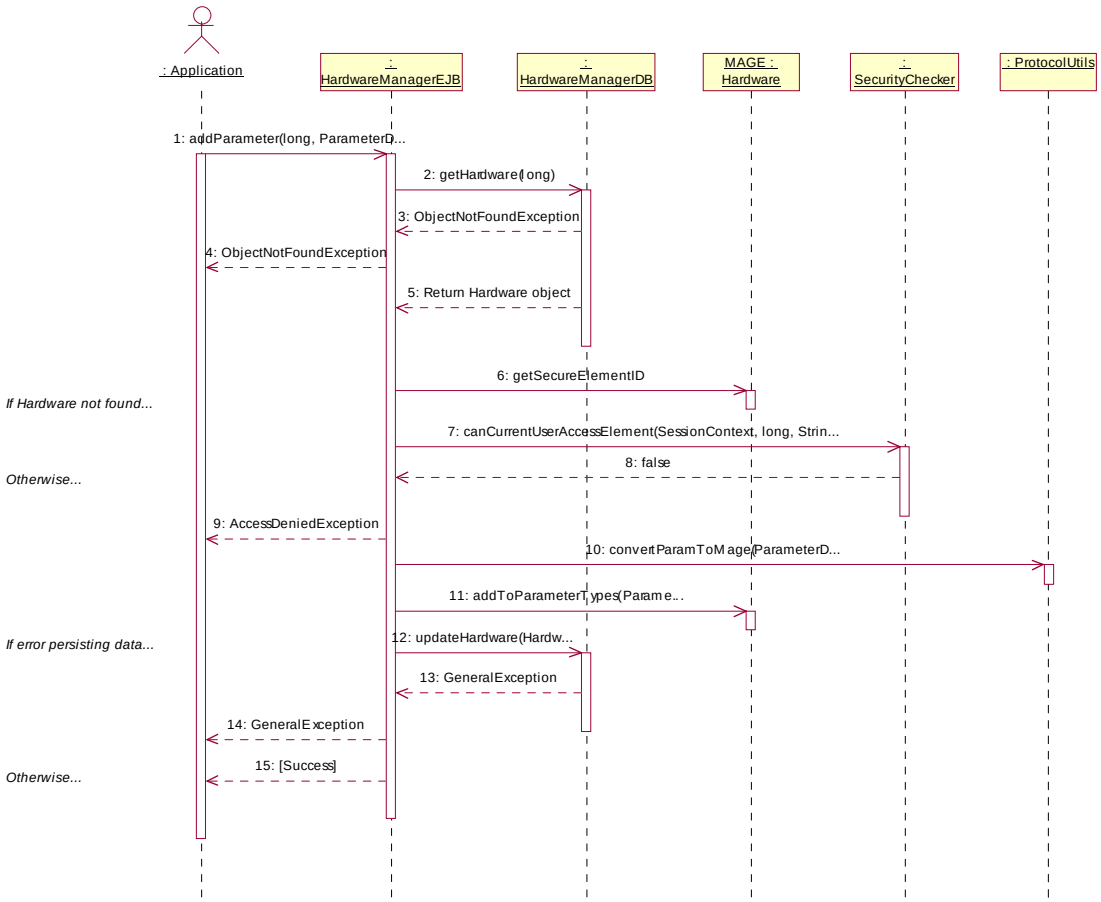
This sequence shows the processing involved to update an existing protocol. If the user has UPDATE privileges, they will be allowed to invoke the updateProtocol method. The EJB then checks to verify that the user has UPDATE privileges for this particular protocol. If they do not, the EJB throws an AccessDeniedException. If they do, the incoming data is transformed to a MAGEstk Protocol object and given to the DB class to persist in the database. If any error occurs during persistence, a general exception is thrown.

30. SoftwareManager.updateSoftware

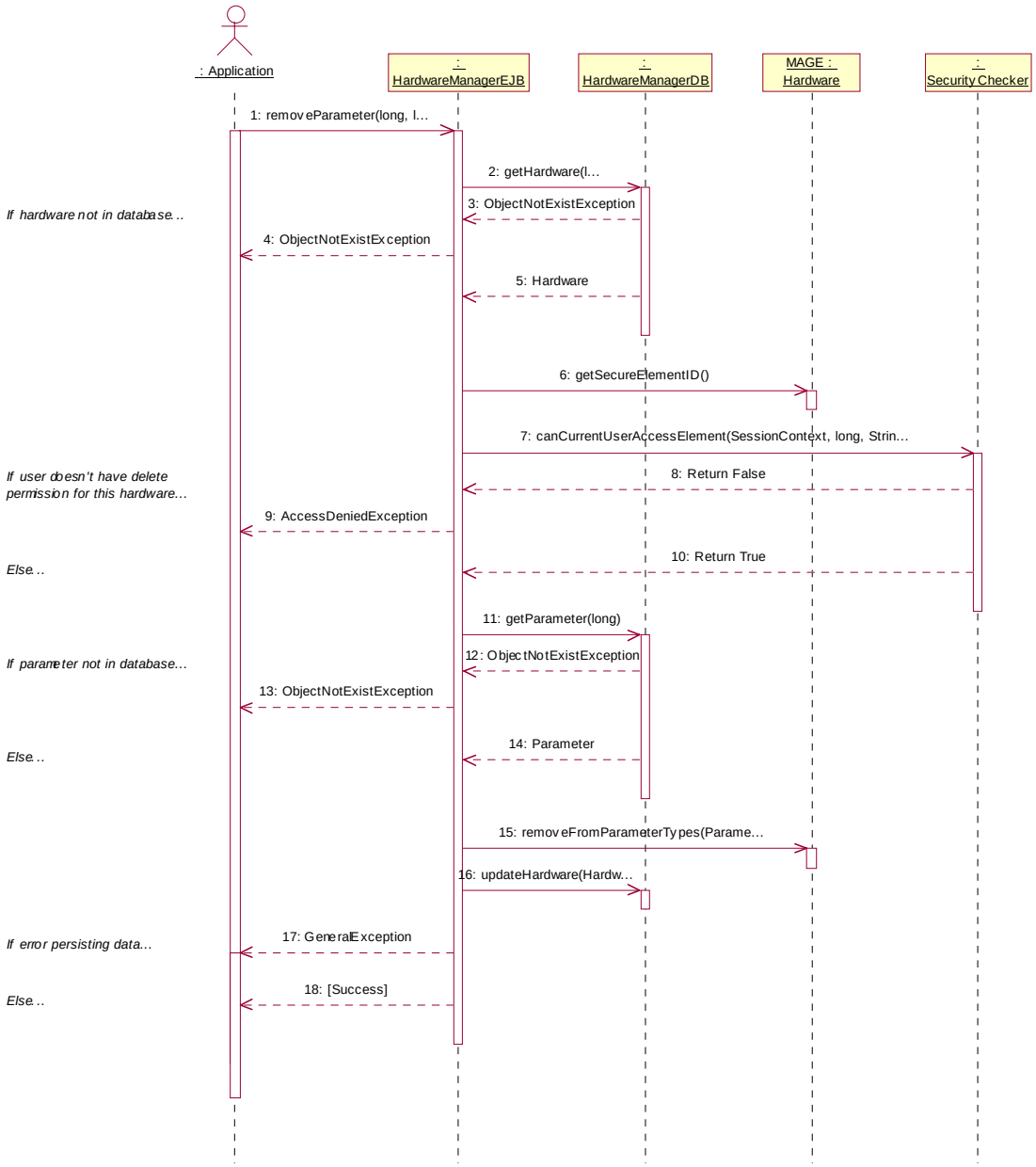


This sequence shows the processing involved to update existing piece of software. If the user has UPDATE privileges, they will be allowed to invoke the updateSoftware method. The EJB then checks to verify that the user has UPDATE privileges for this particular software. If they do not, the EJB throws an AccessDeniedException. If they do, the incoming data is transformed to a MAGEstk Software object and given to the DB class to persist in the database. If any error occurs during persistence, a general exception is thrown.

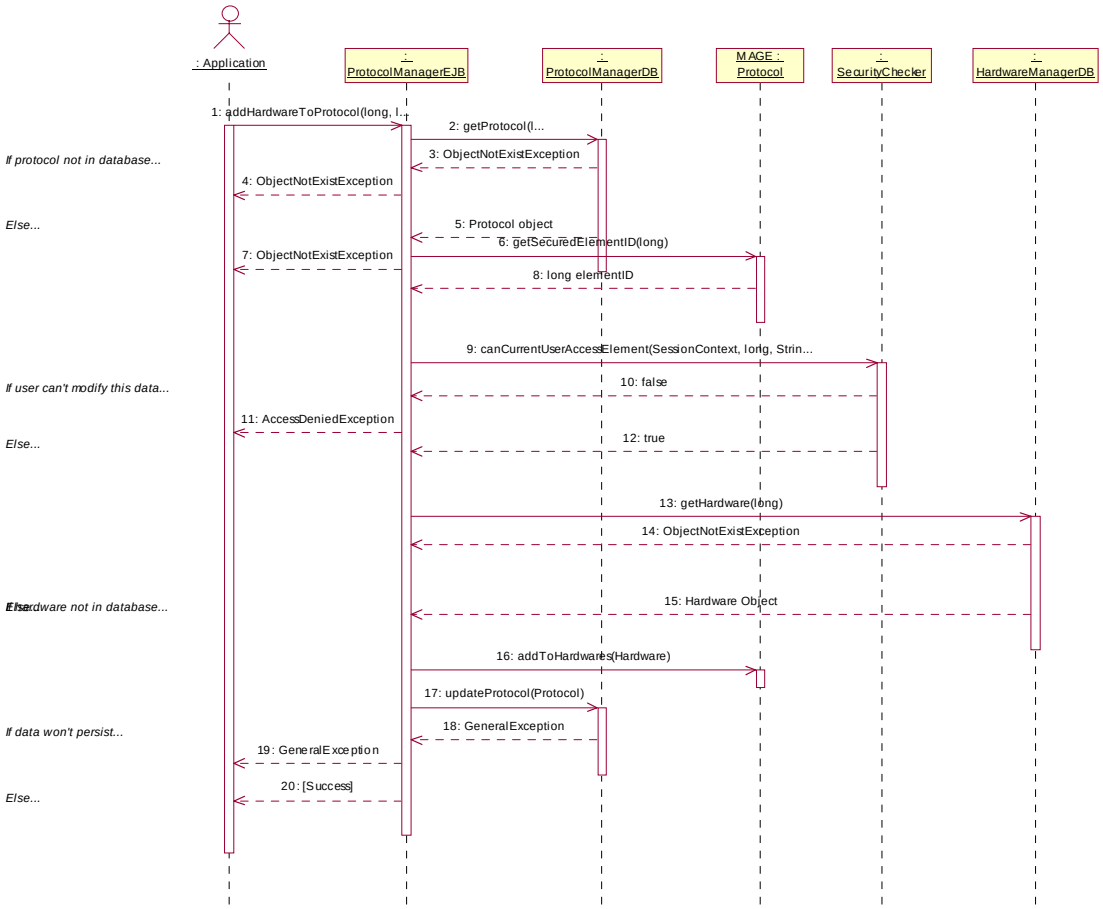
31. HardwareManager.addParameter



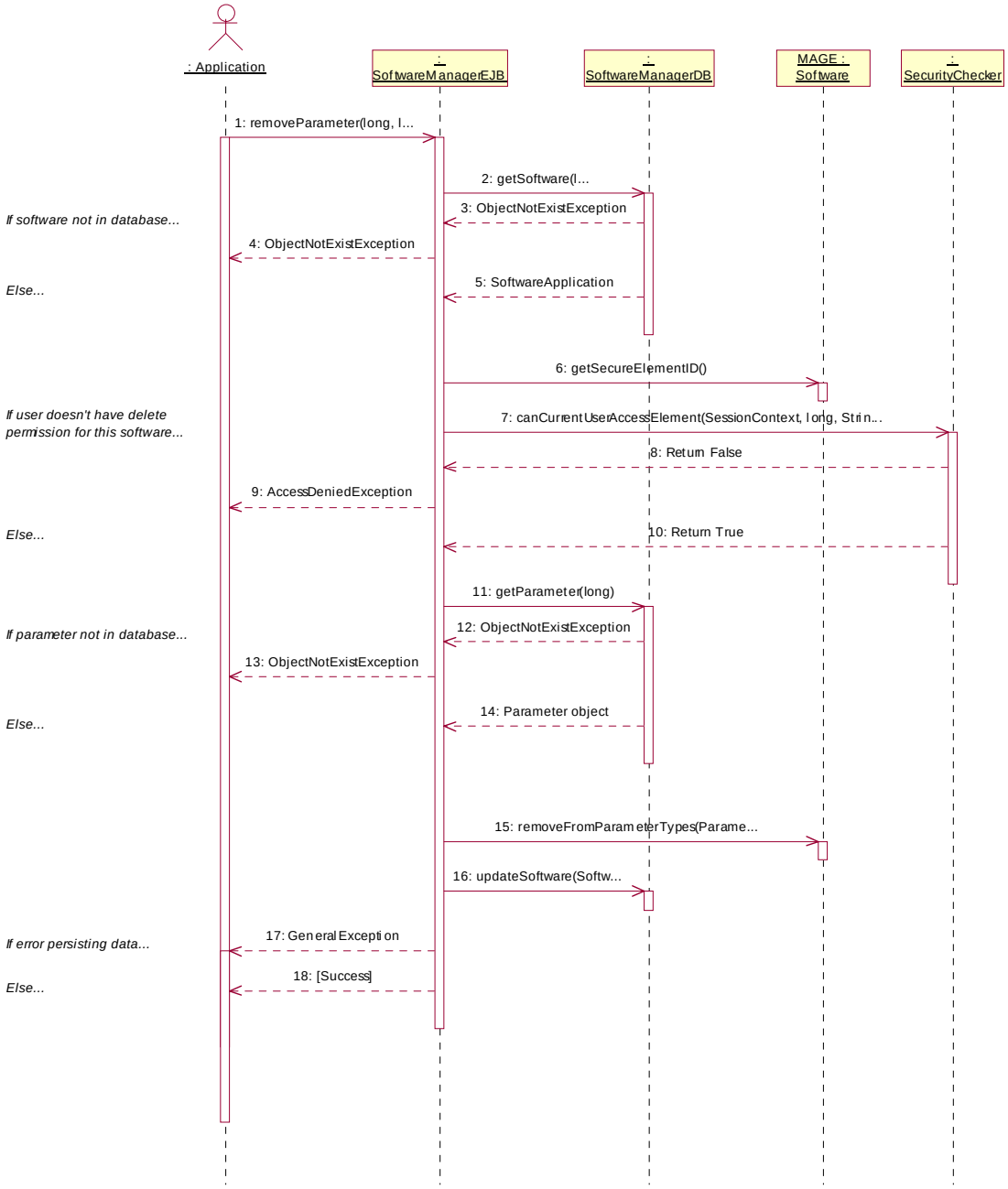
32. HardwareManager.removeParameter



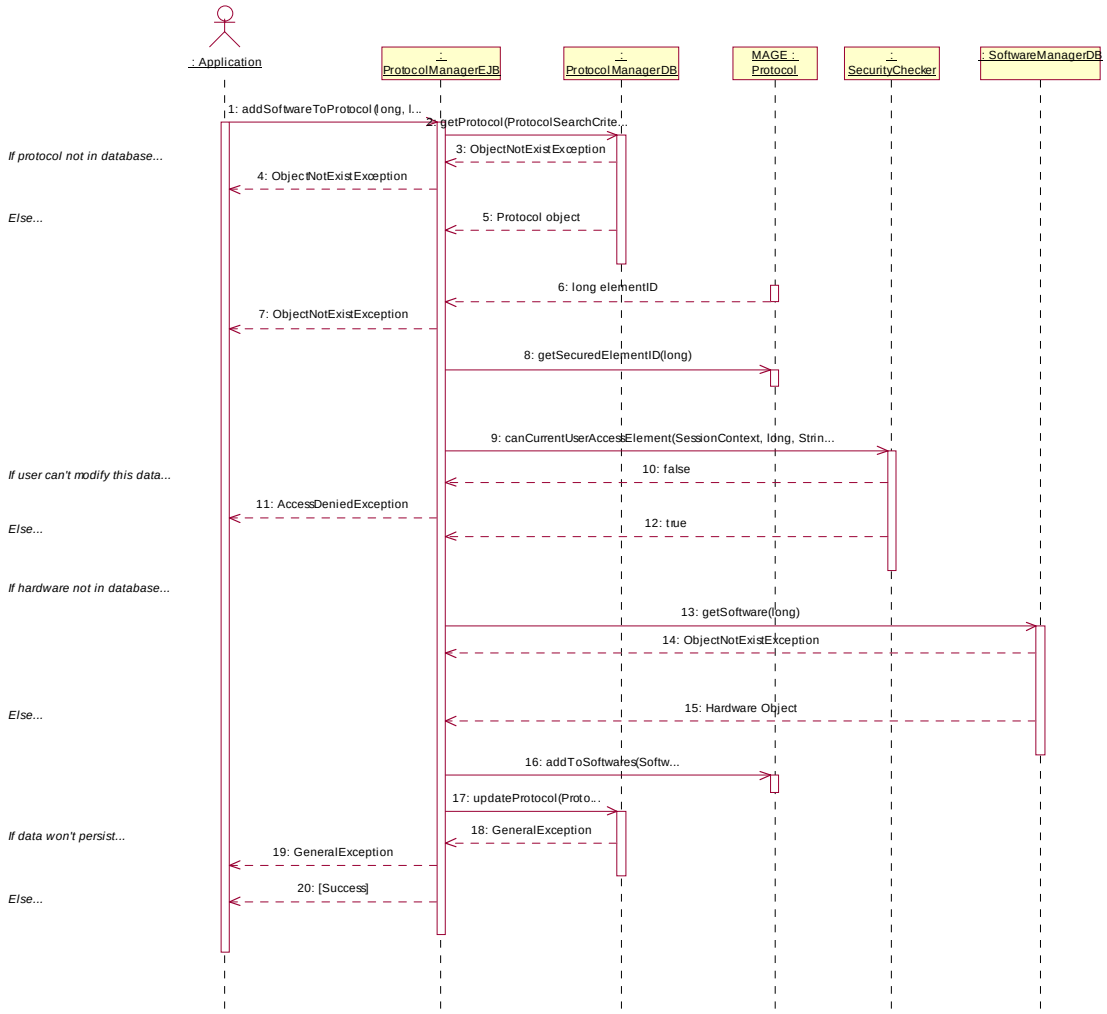
33. ProtocolManager.addHardware



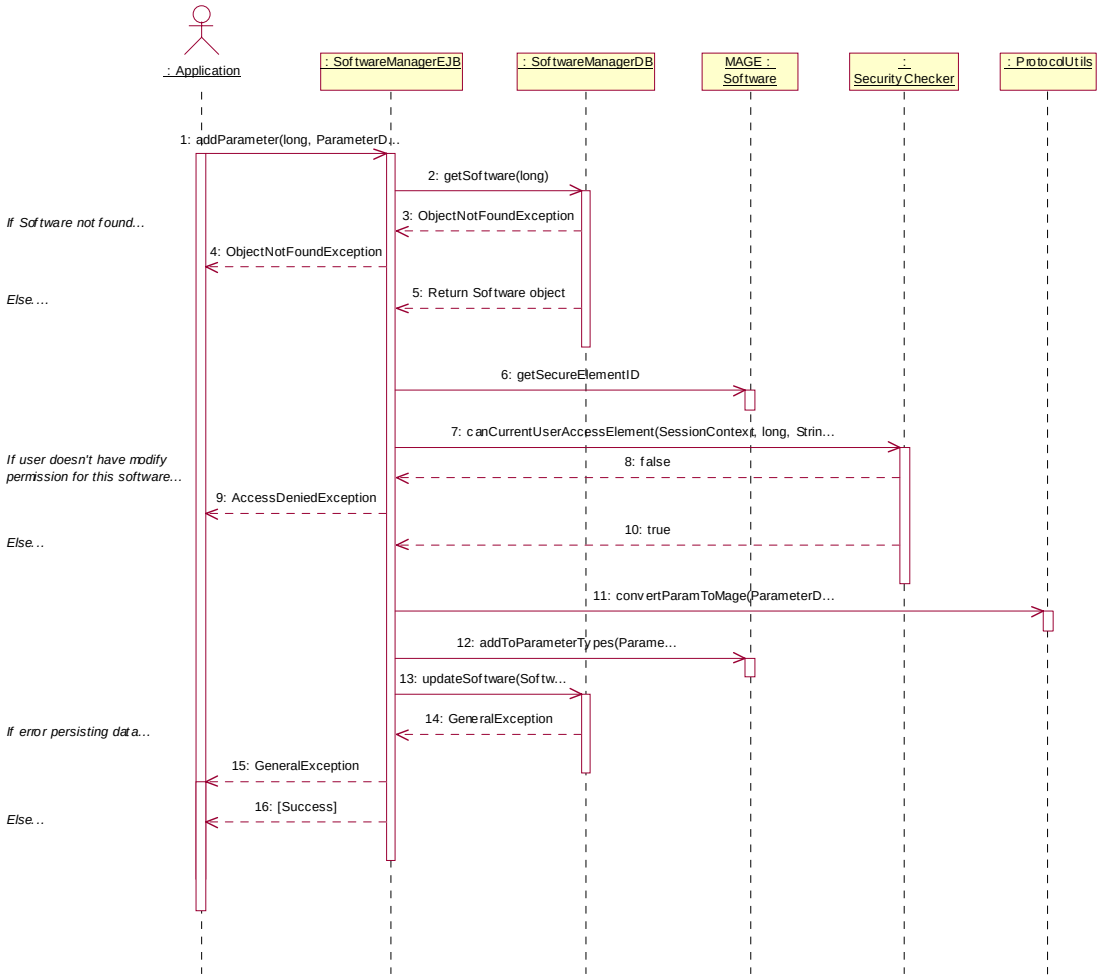
34. SoftwareManager.removeParameter



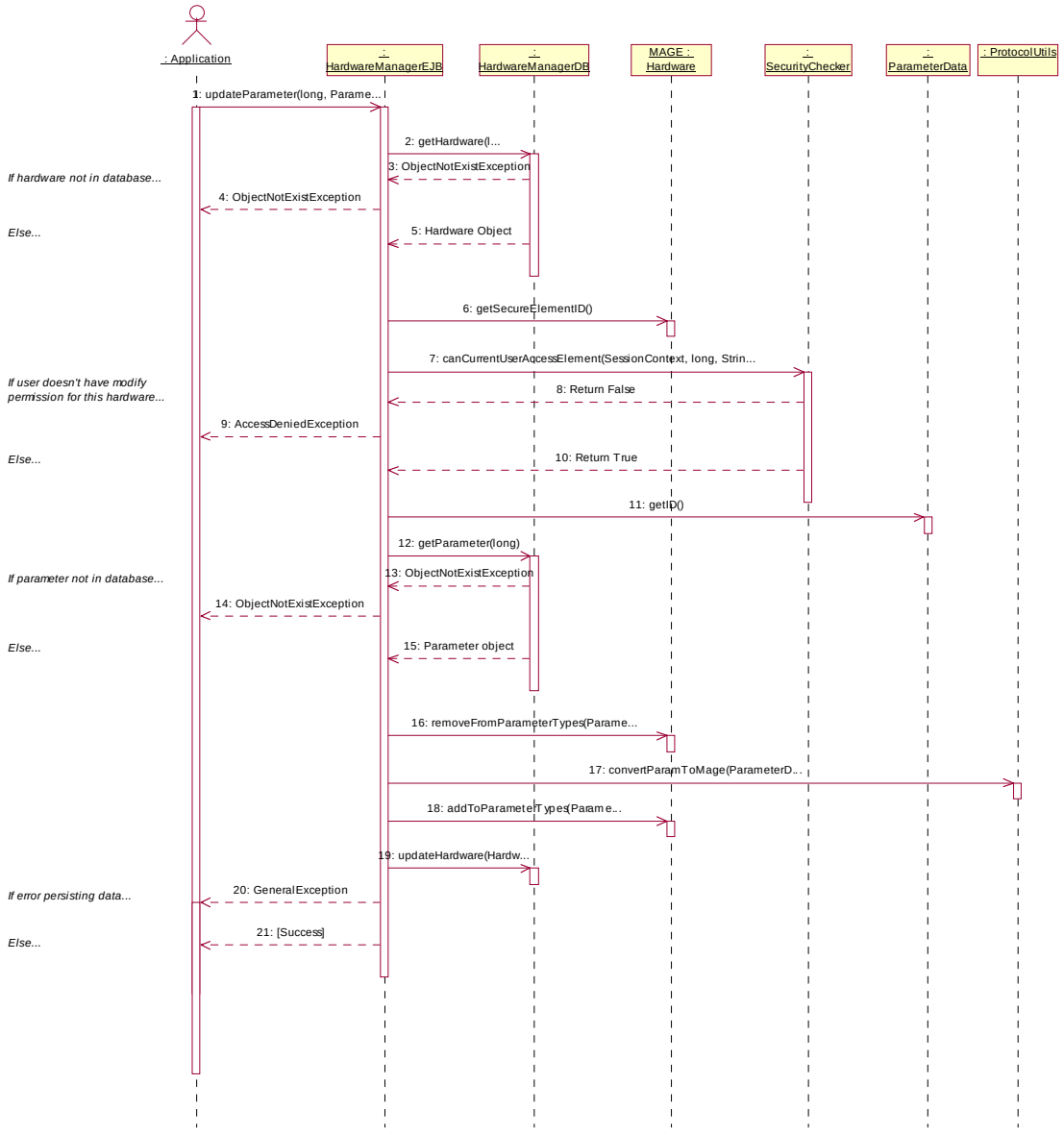
35. ProtocolManager.addSoftware



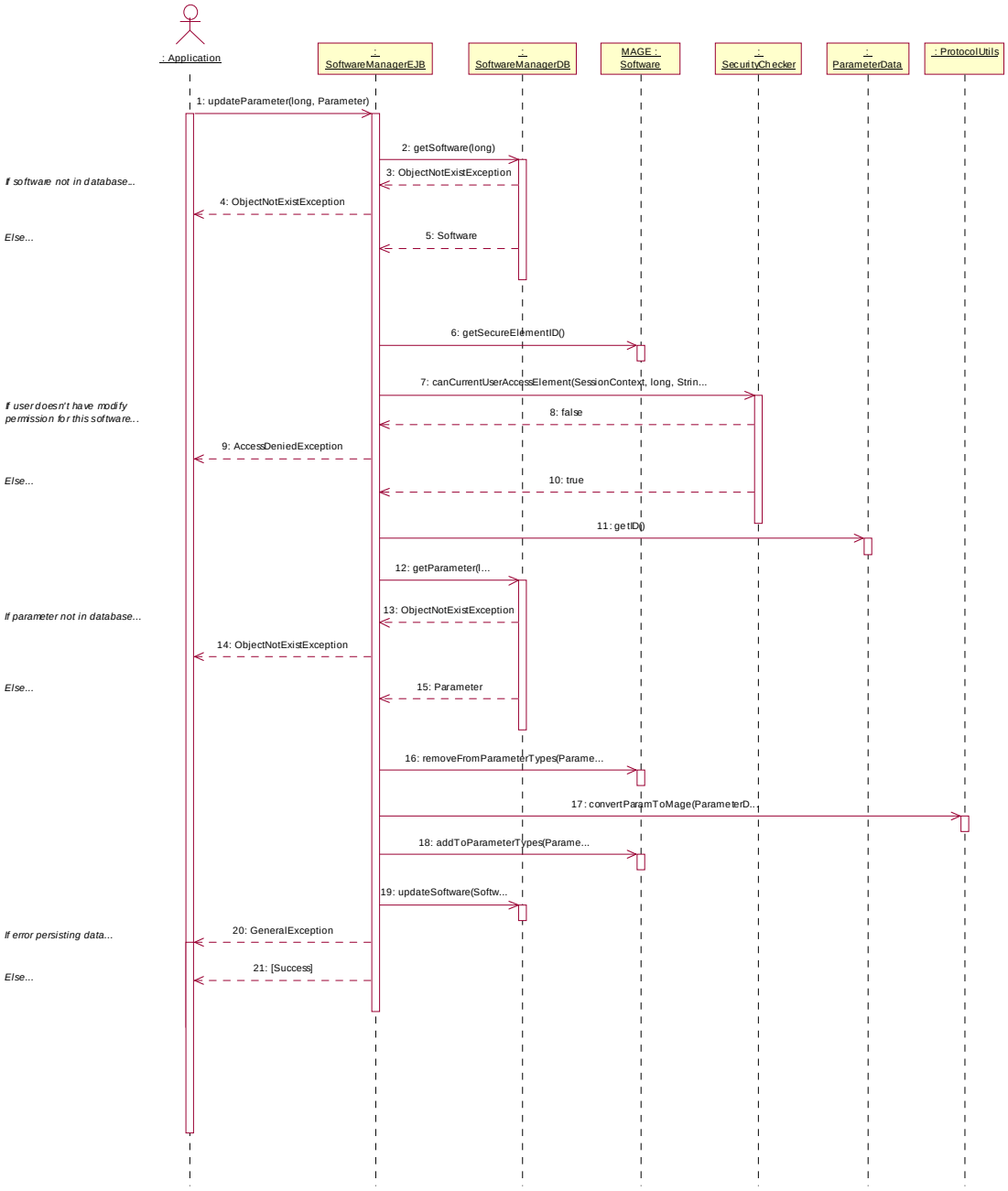
36. SoftwareManager.addParameter



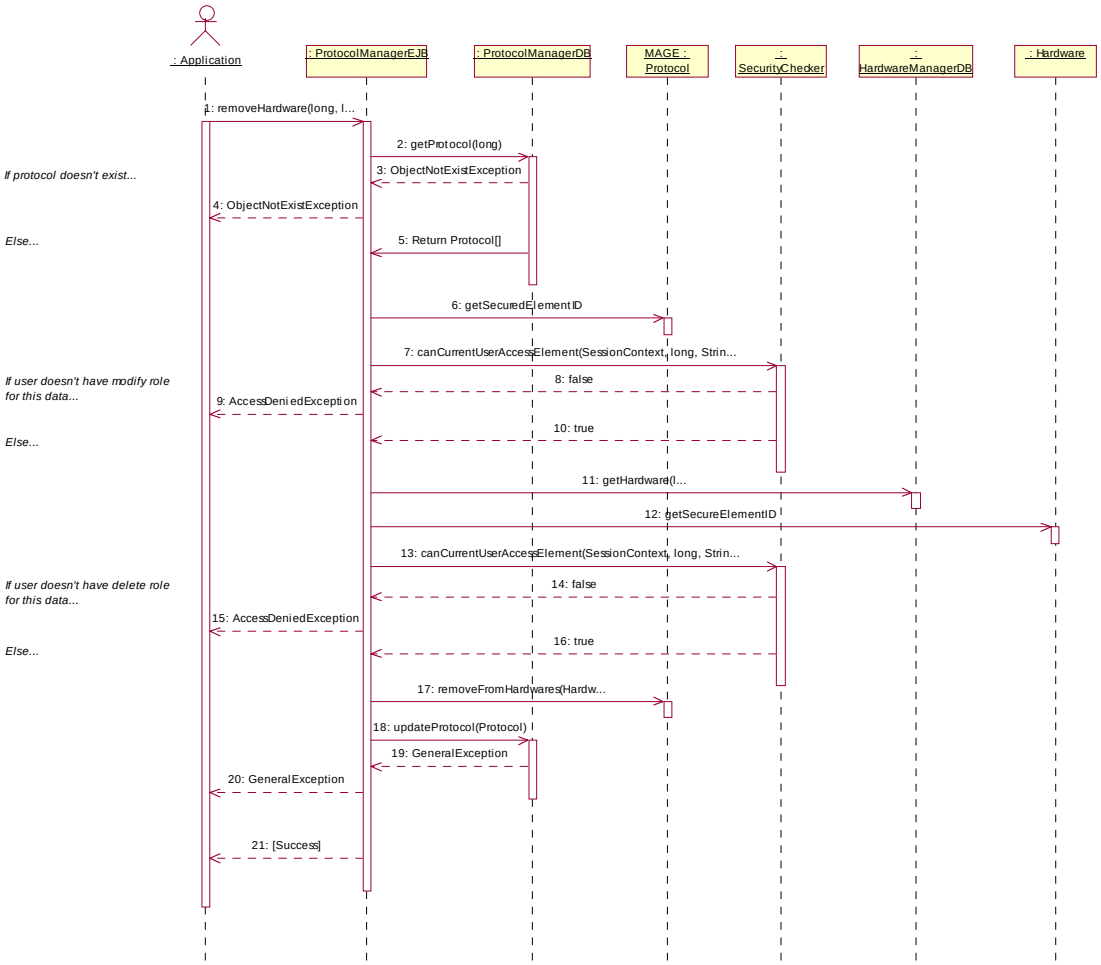
37. HardwareManager.updateParameter



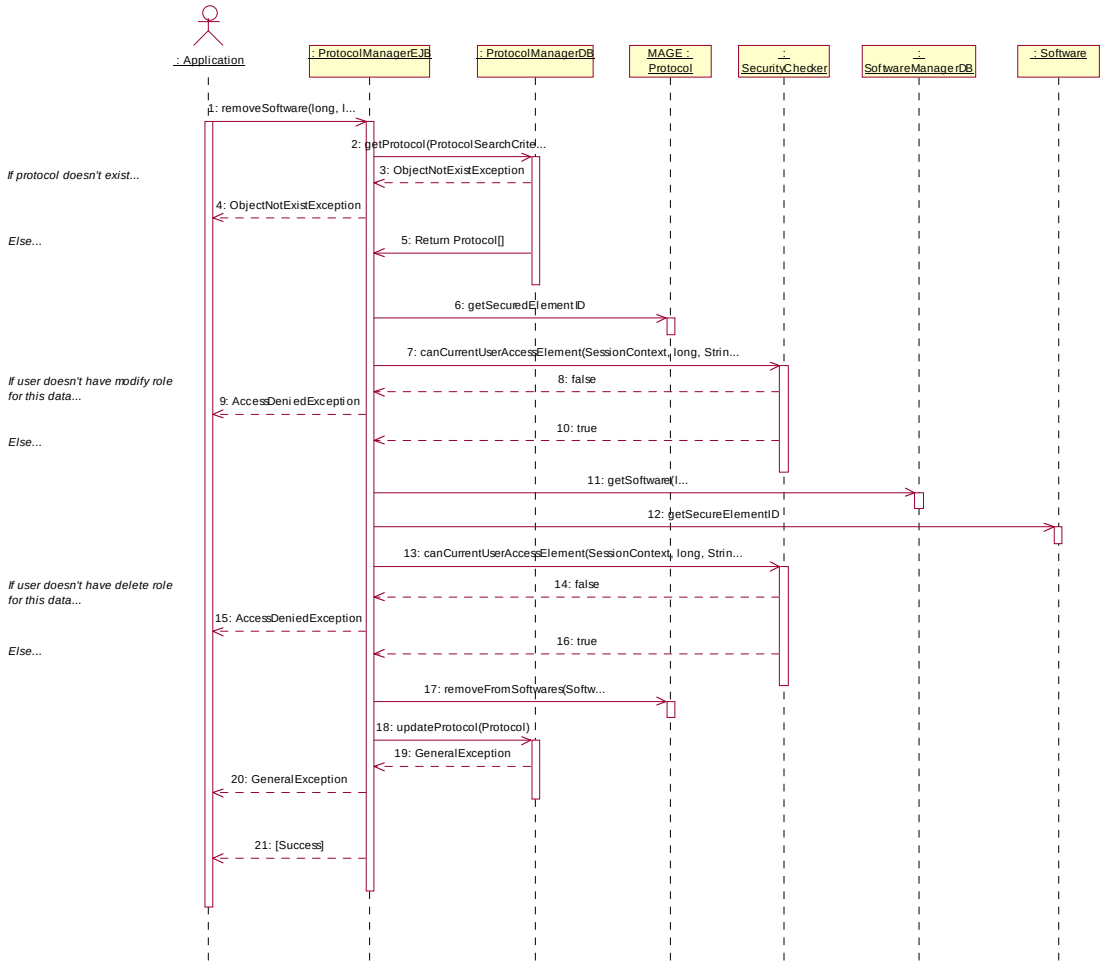
38. SoftwareManager.updateParameter



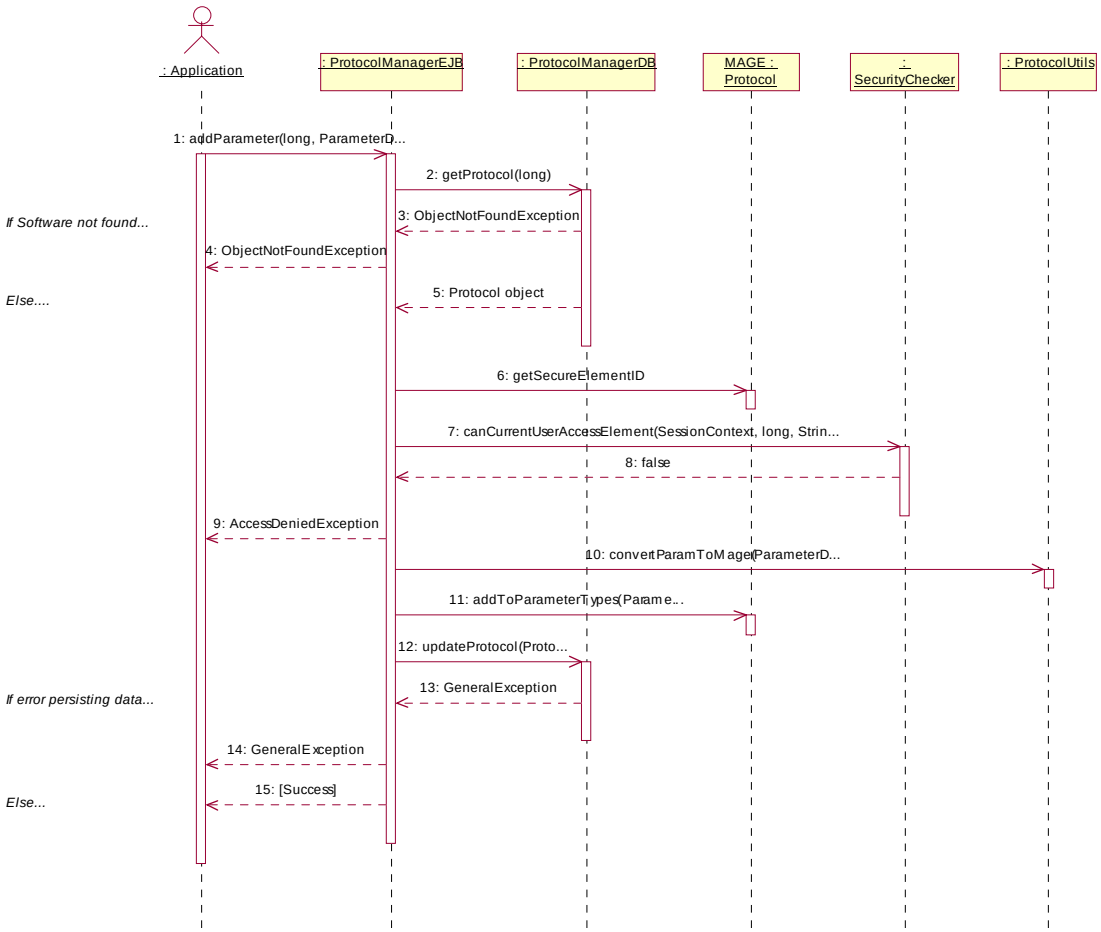
39. ProtocolManager.removeHardware



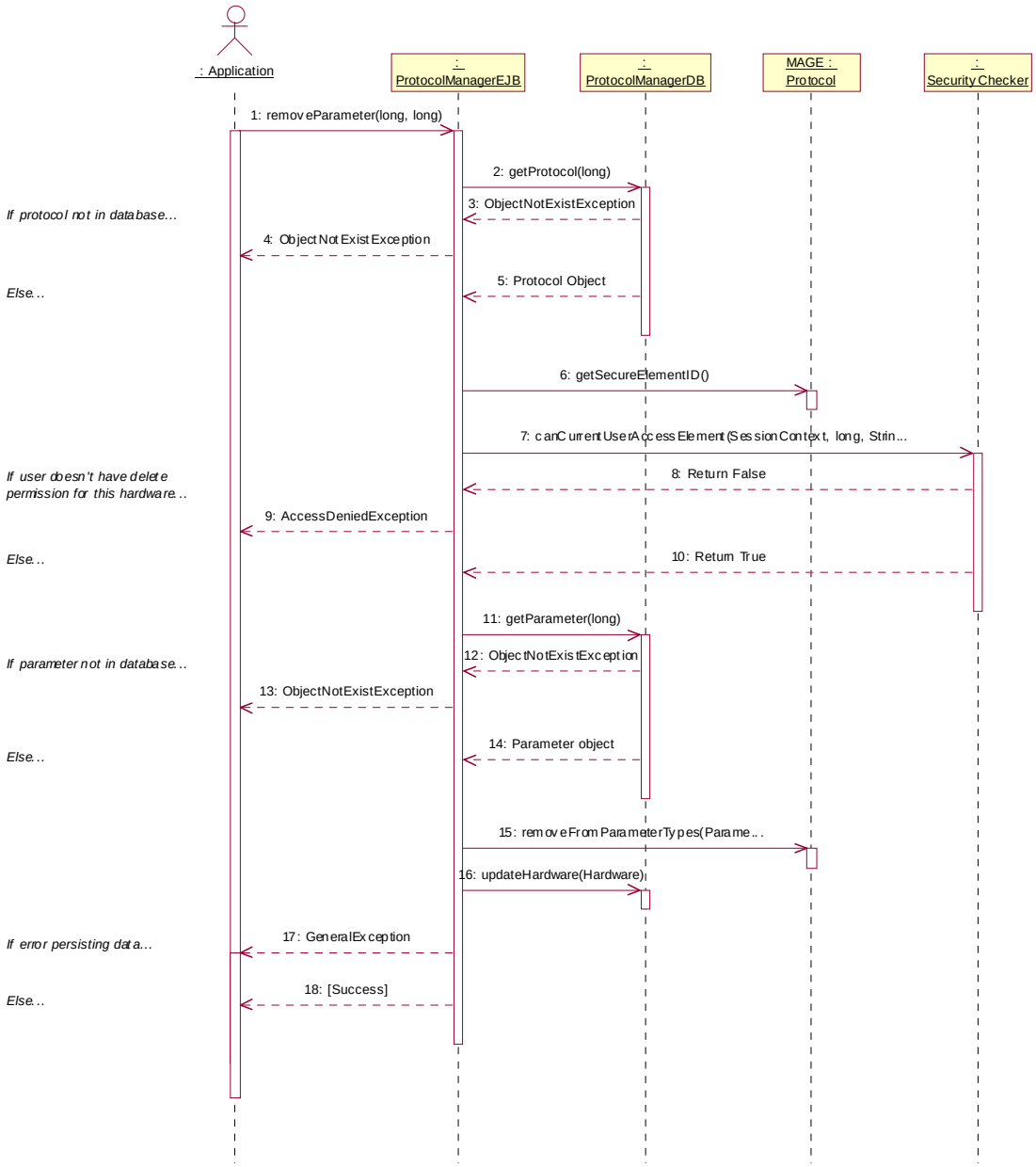
40. ProtocolManager.removeSoftware



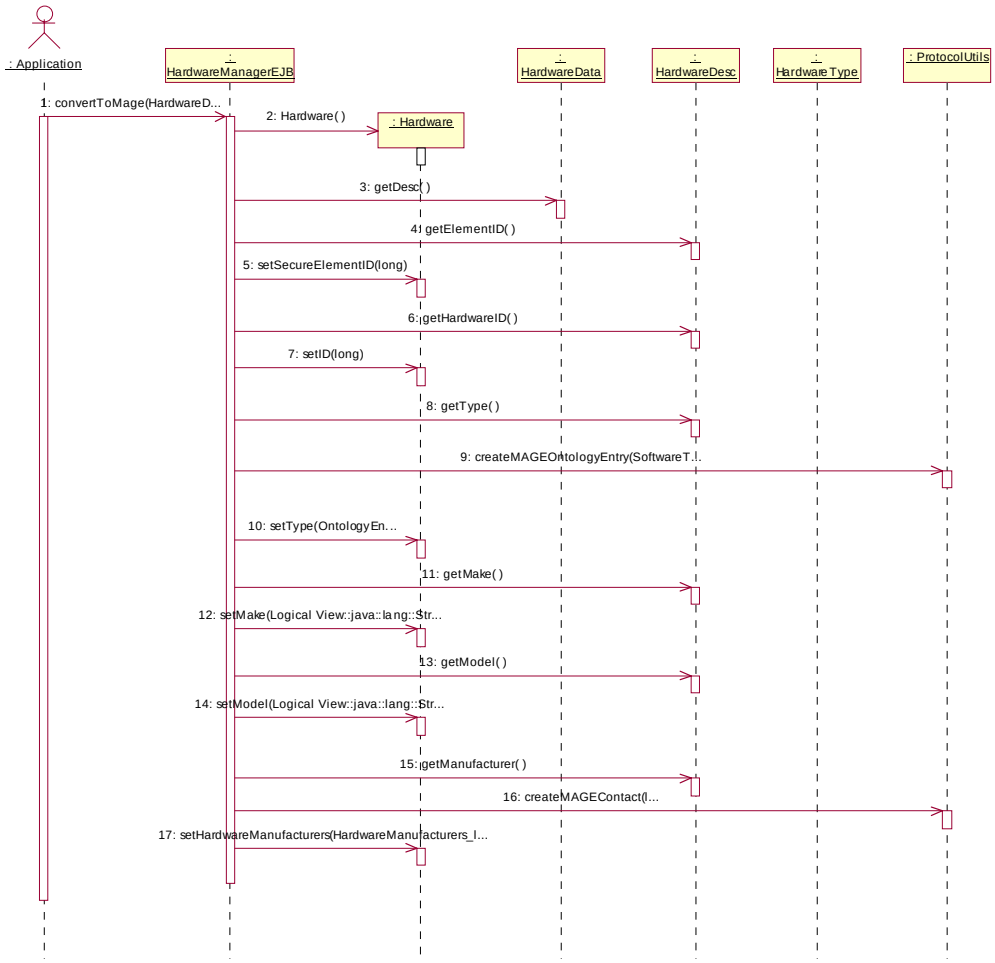
41. ProtocolManager.addParameter



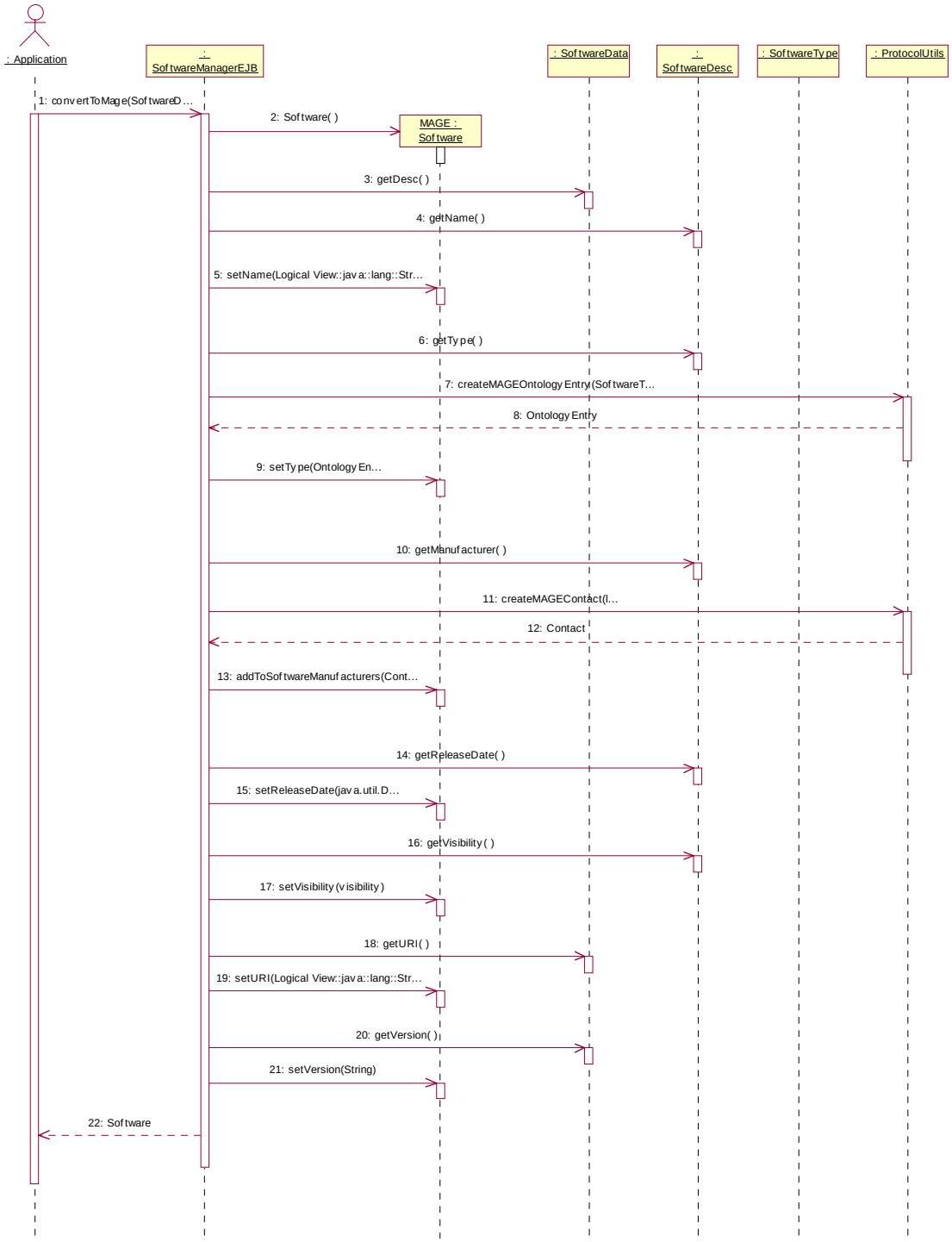
42. ProtocolManager.removeParameter



43. HardwareManager.convertToMage

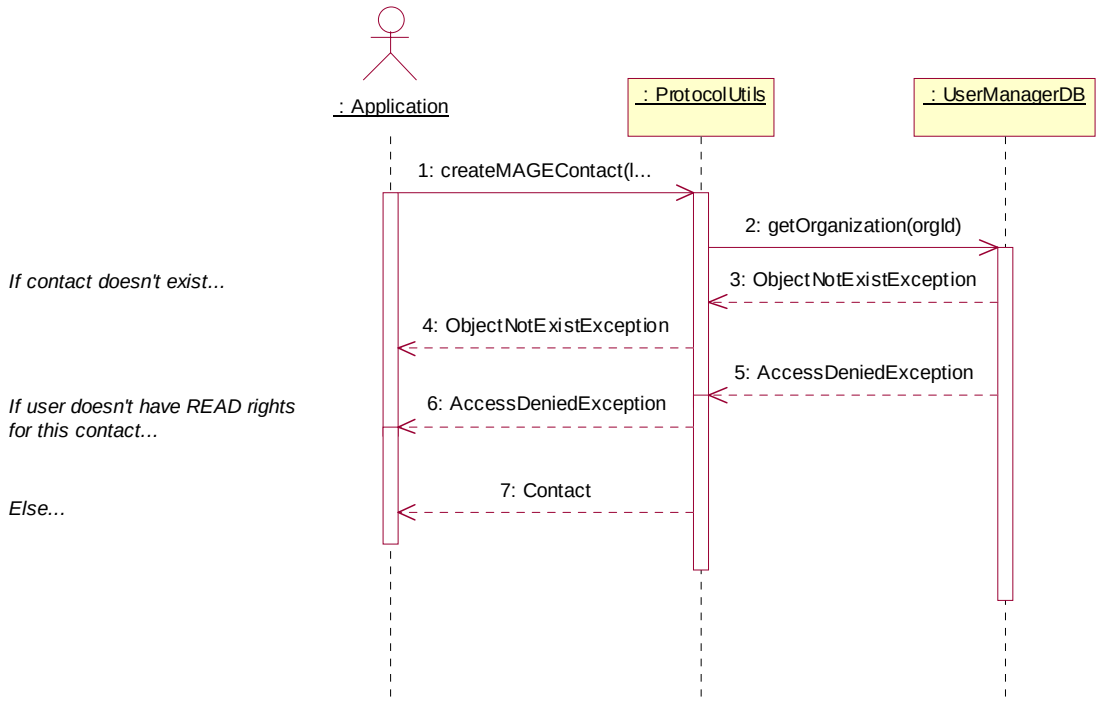


44. SoftwareManager.convertToMage



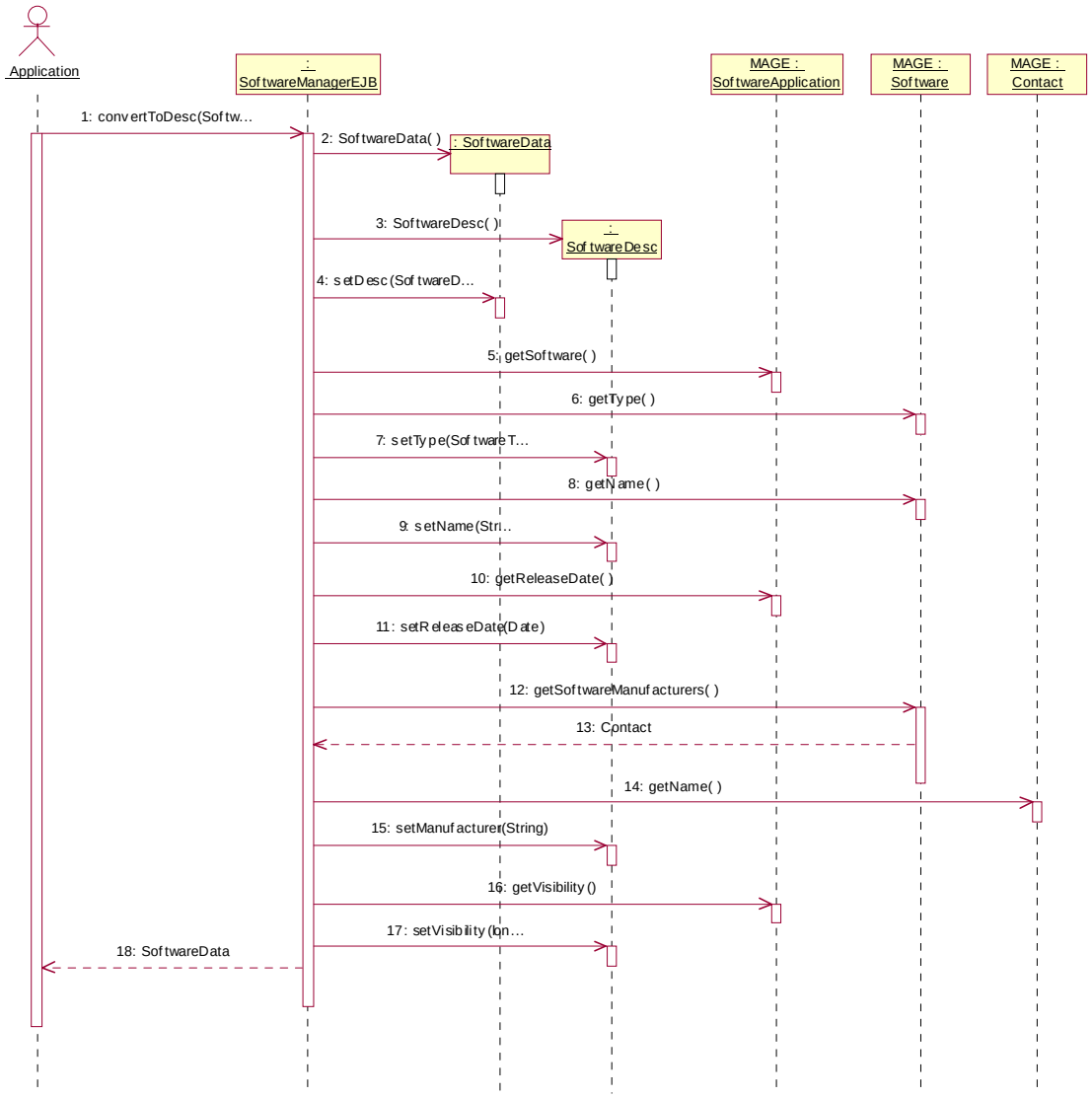
This diagram demonstrates the process of mapping data in a SoftwareDesc object to a MAGEstk SoftwareApplication object, which is then persisted to the database.

45. ProtocolUtils.createMAGEContact



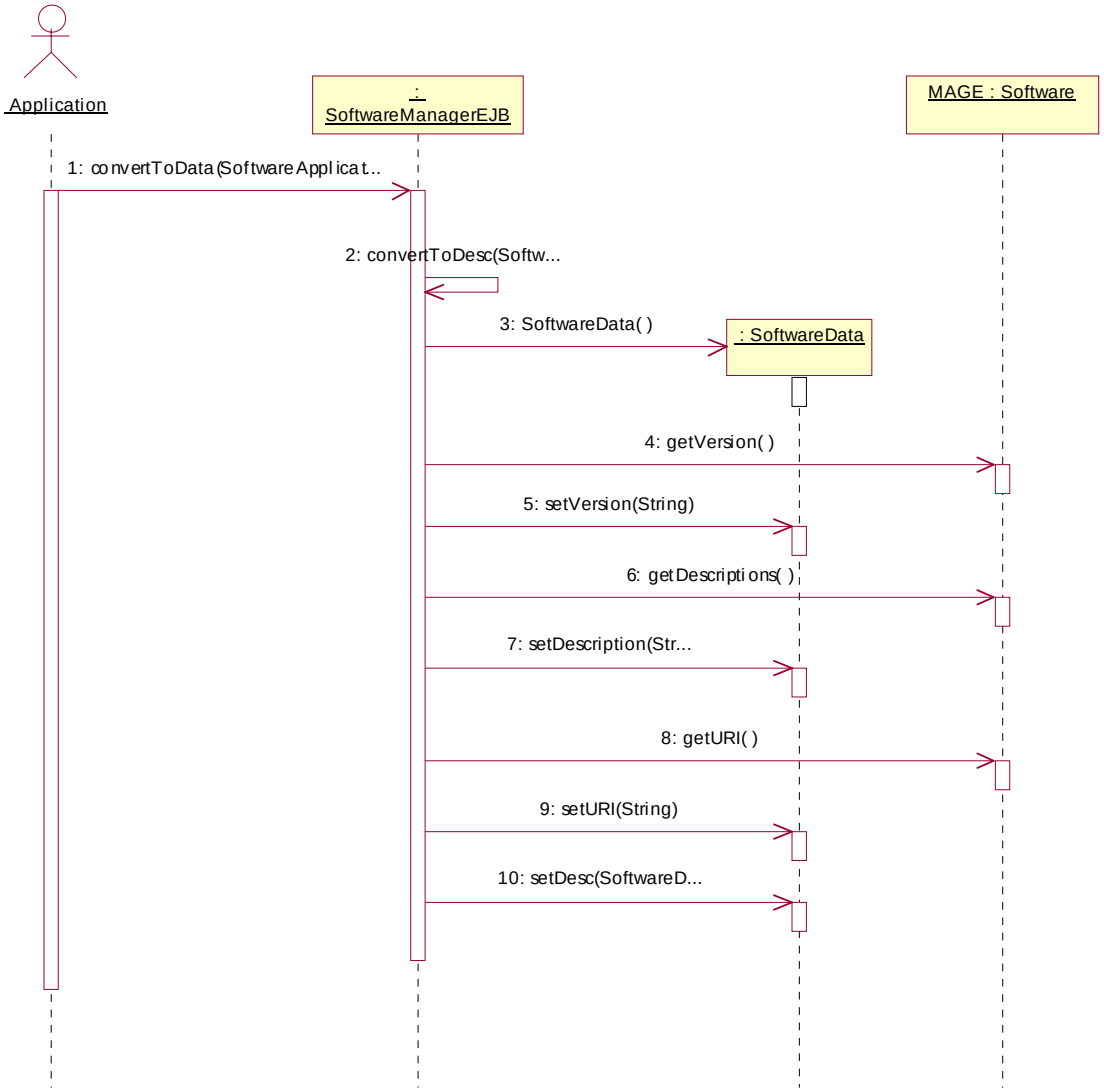
The sequence demonstrates a utility method that retrieves MAGE contact from the database.

46. SoftwareManager.convertToDesc



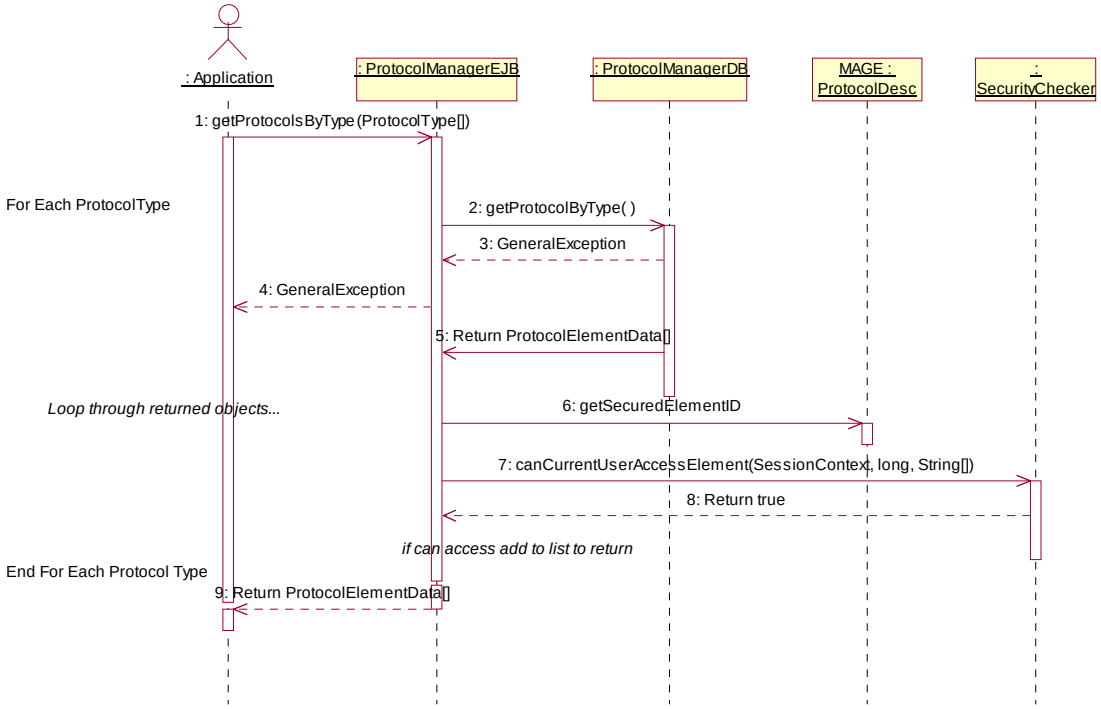
This diagram demonstrates the process of mapping data from a MAGEstk SoftwareApplication to object to a SoftwareDesc object, which is then used by the application. The Desc contains only a summary of data need to display in search results.

47. SoftwareManager.convertToData

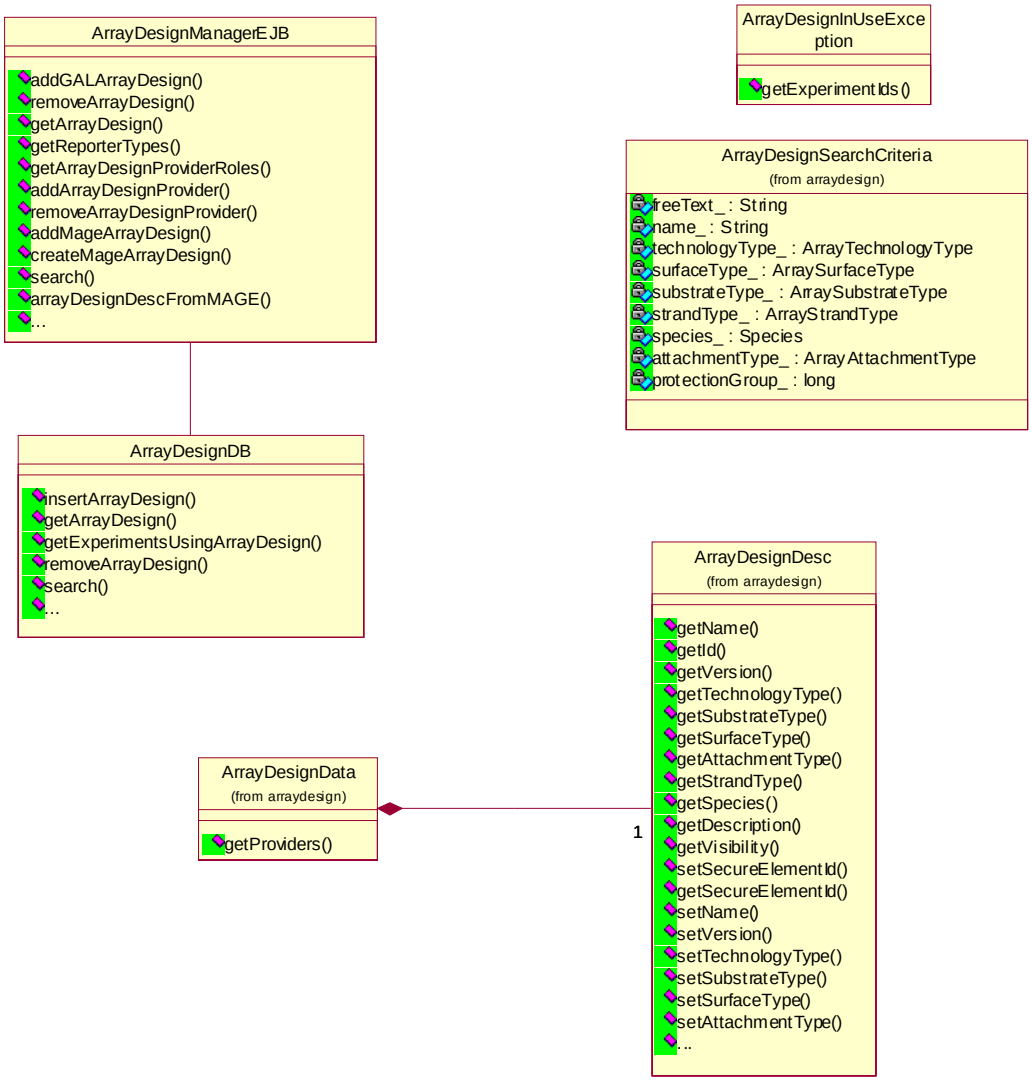


This diagram demonstrates the process of mapping data from a MAGEstk SoftwareApplication to object to a SoftwareData object, which is then used by the application. The contains only a summary of data need to display in search results.

48. ProtocolManager.getProtocolsByType



49. Logical View::gov::nih::nci::caarray::services::arraydesign



This diagram shows the static relationships between classes in the caarray.services.arraydesign package.

49.1 ArrayDesignManagerEJBThis class provides the public API for dealing with array design objects in the caarray system.

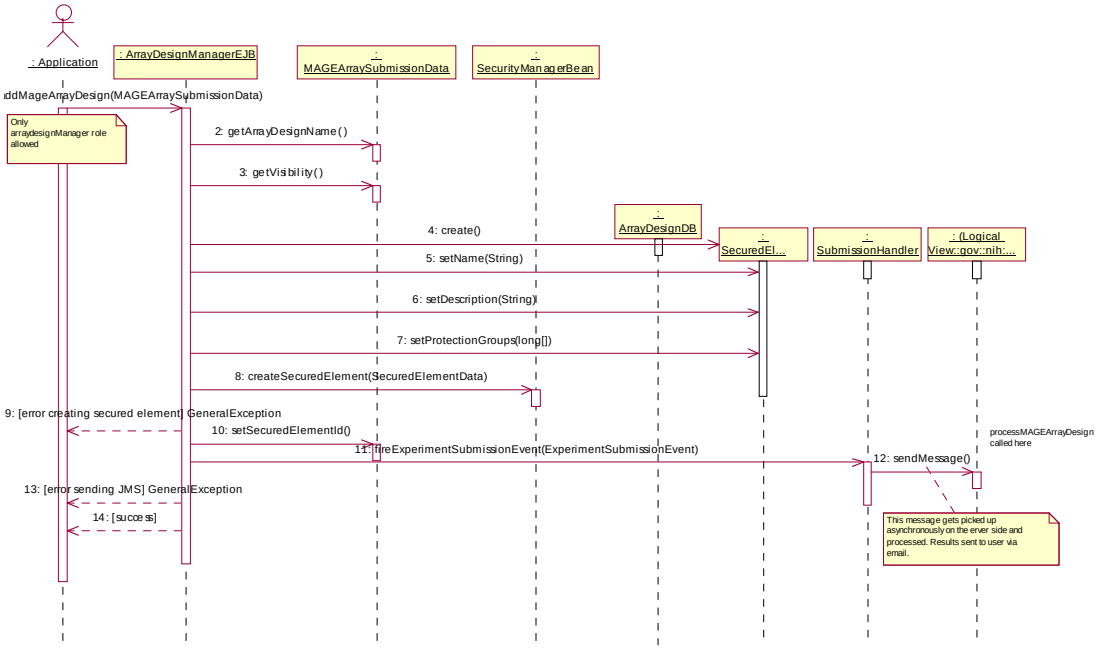
49.2 ArrayDesignDBThis class is responsible for all database operations needed within the array design service.

49.3 ArrayDesignInUseExceptionThis exception is thrown when an operation cannot be performed on an ArrayDesign object because it is currently in use.

49.4

49.5

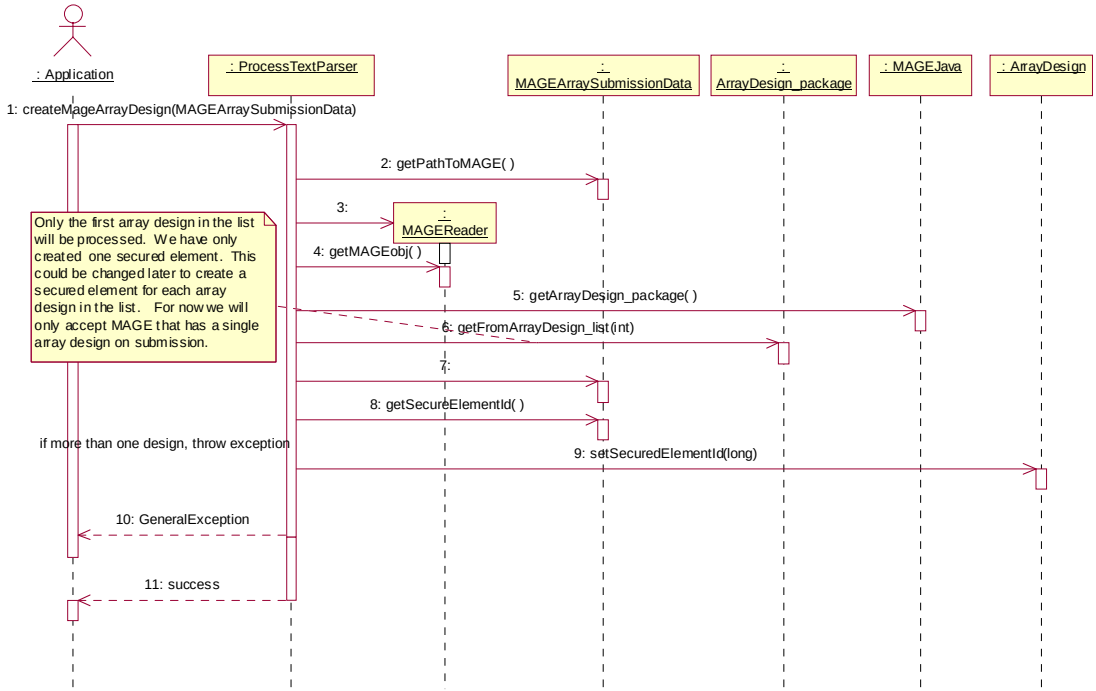
50. ArrayDesignManager.addMAGEArrayDesign



This diagram shows the processing involved in submitting a new MAGE-ML array design. The ArrayDesignManagerEJB will get the array design name and visibility from the submission data and use it to create a new SecuredElement via the SecurityManager. If the secured element cannot be created, the array design submission will fail and the caller will get an exception describing the problem. Once the secured element has been created, the submission data will be converted into a

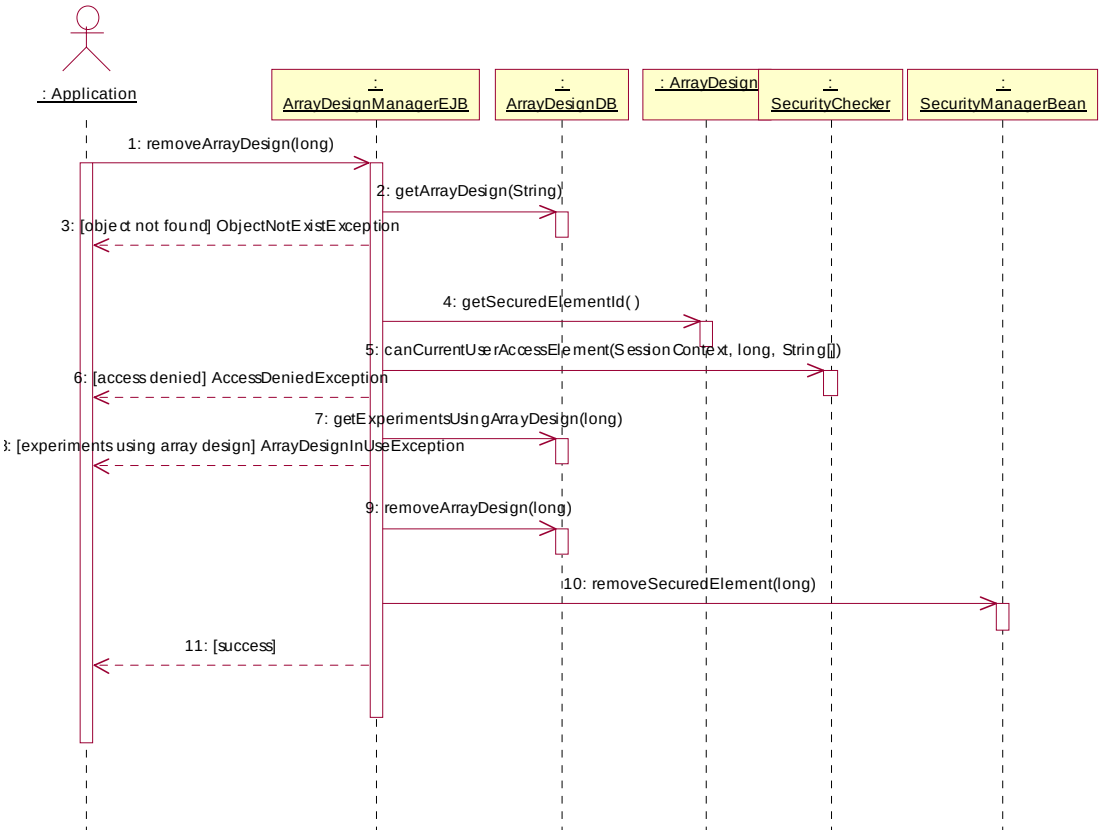
MAGE-stk ArrayDesign object and persisted to the database. If persistence to the database fails and exception will be thrown.

51. ArrayDesignManager.createMageArrayDesign



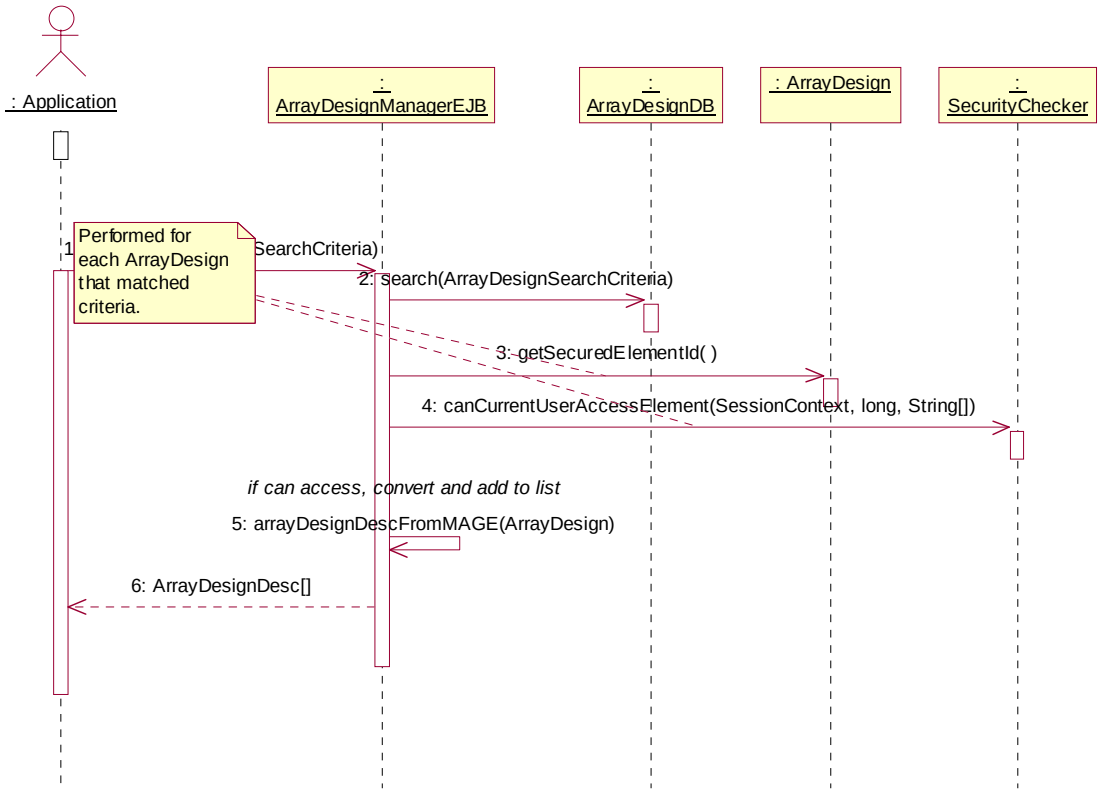
This diagram shows how a MAGE array design is created from a MAGEArraySubmissionData object.

52. ArrayDesignManager.removeArrayDesign



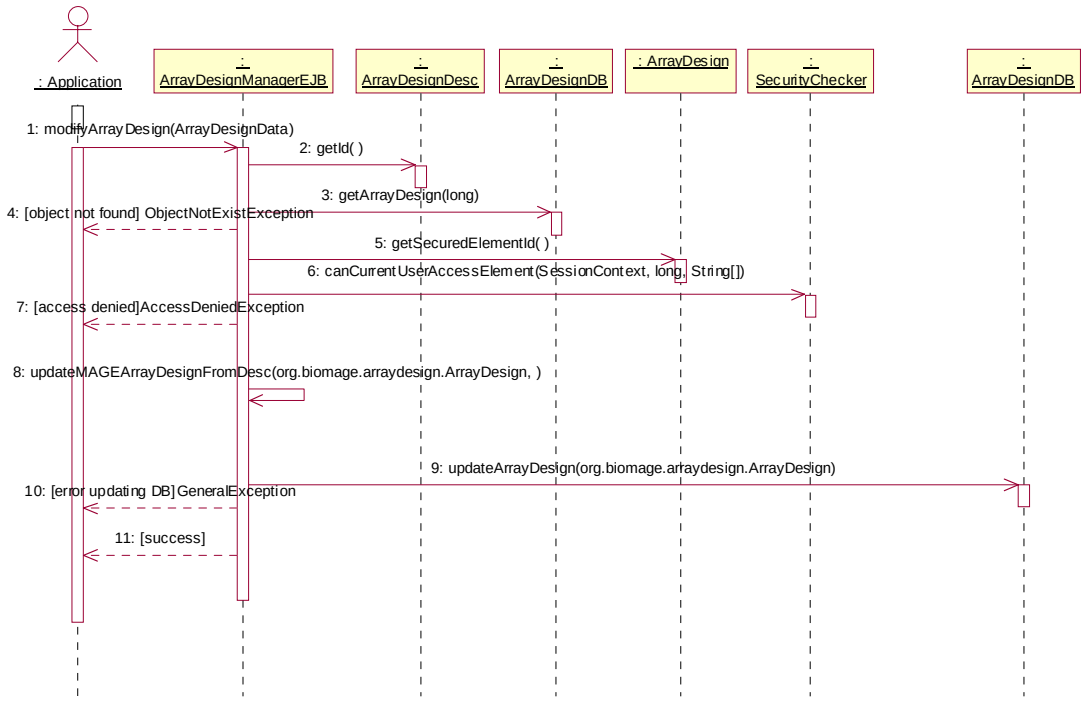
This diagram show the processing performed to remove an array design from the system. The ArrayDesign is retrieved from the database to obtain its security id. If the ArrayDesign does not exist, an exception is thrown. Next the security service is called to determine if the current user can perform this operation on this array design. If not, an AccessDeniedException is thrown. If the user can remove this array design, a final check is made to see if any experiments in the database reference this array design. If they do an ArrayDesignInUse exception is thrown. This exception contains the identifiers of the experiments that are using the array design. If the array design is not in use, it is removed from the security manager, then finally from the database.

53. ArrayDesignManager.search



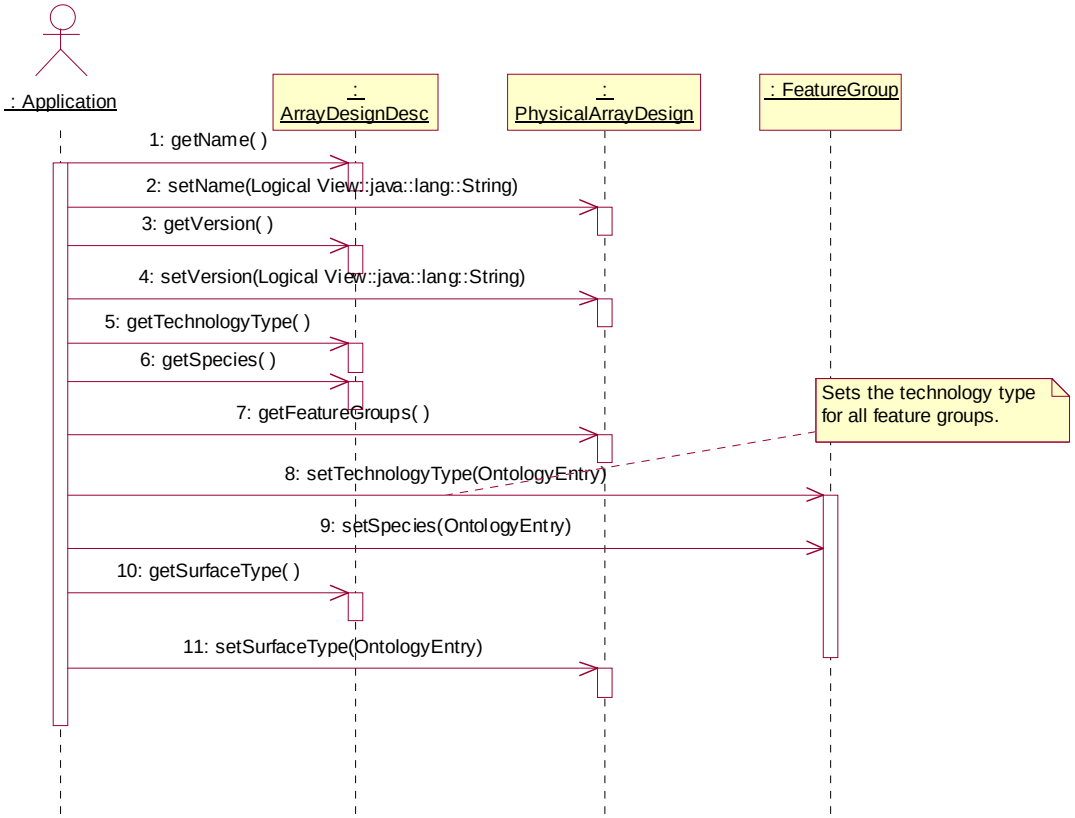
This diagram shows the processing performed when a user searches for array designs. The specified search criteria are passed to the database. The database returns only those ArrayDesign objects that meet the specified criteria. Each returned array design is checked to verify that the calling user can see it. An ArrayDesignDesc object is created for each ArrayDesign that the caller is allowed to access. The collection of these ArrayDesignDesc objects is then returned to the caller.

54. ArrayDesignManager.modifyArrayDesign



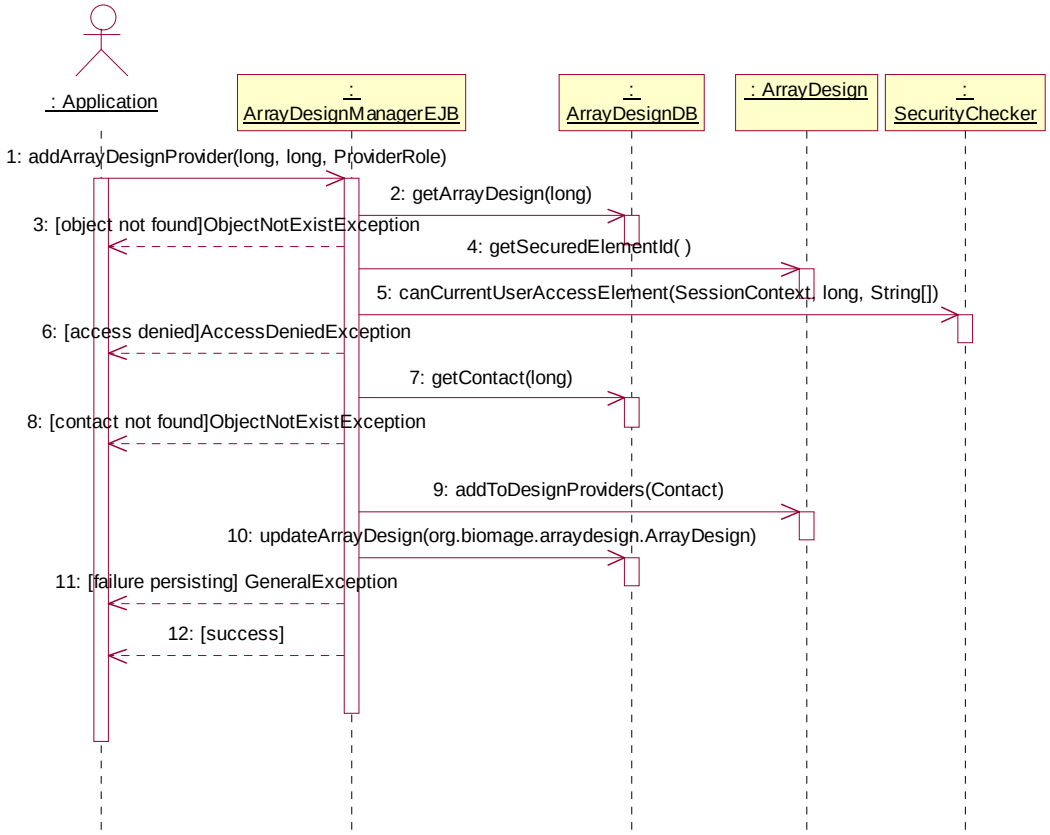
This diagram shows the processing performed when a user alters the general information of an array design. First the array design is obtained from the database. Then a check is made to verify that this user is allowed to alter this array design. If the user is allowed to alter the design, the data will be copied from the new ArrayDesignDesc object to the ArrayDesign and the ArrayDesign object will be updated in the database.

55. ArrayDesignManager.updateMAGEArrayDesignFromDesc



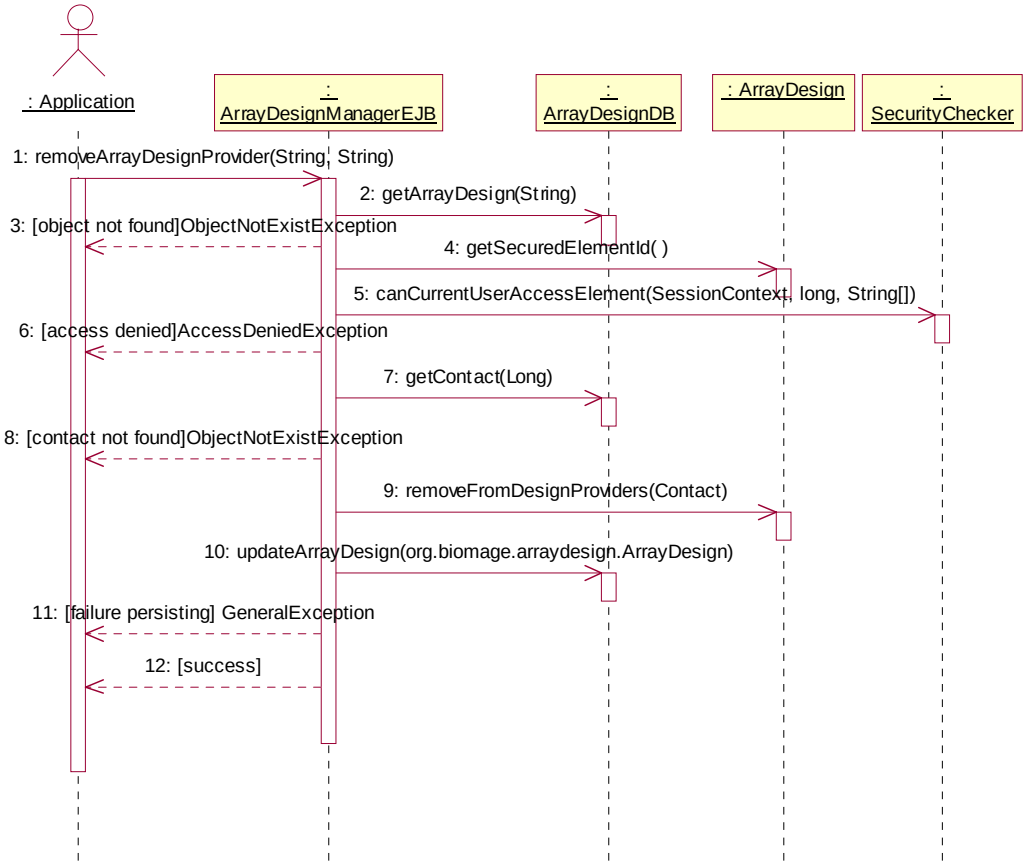
This diagram shows the processing performed to copy the data from a caarray ArrayDesignDesc object into a MAGE ArrayDesign object.

56. ArrayDesignManager.addArrayDesignProvider



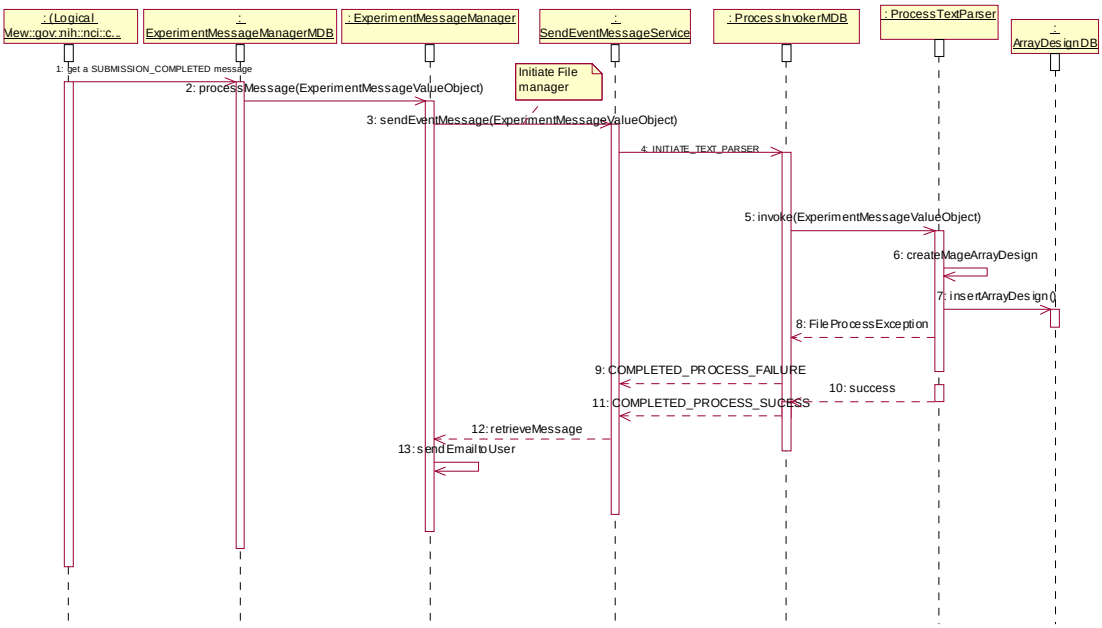
This sequence shows the processing performed to add a new array design provider to an existing array. The array is retrieved from the database. Then a check is made to verify that this user can alter this array design. If the user is allowed to alter the design the contact will be retrieved from the database. If the contact with the specified ID can be retrieved, his/her contact information will be associated with the array design and the array design will be updated in the database.

57. ArrayDesignManager.removeArrayDesingProvider



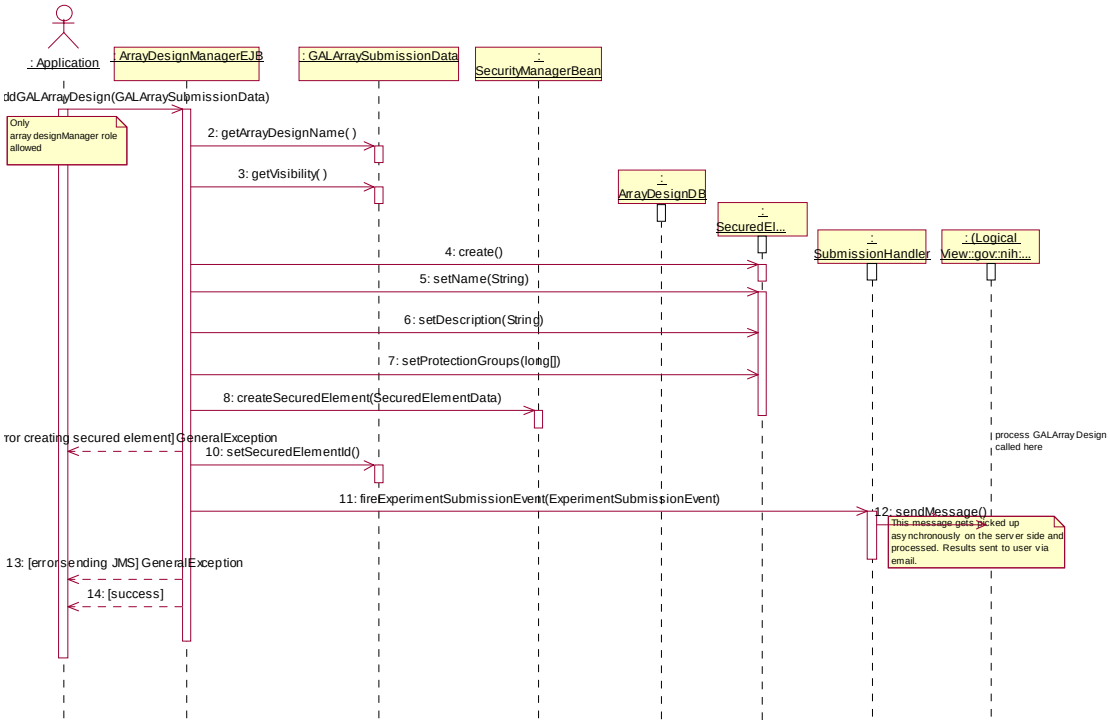
This diagram shows the processing performed when a provider is removed from an array design.

58. ArrayDesignManager.processMAGEArrayDesign

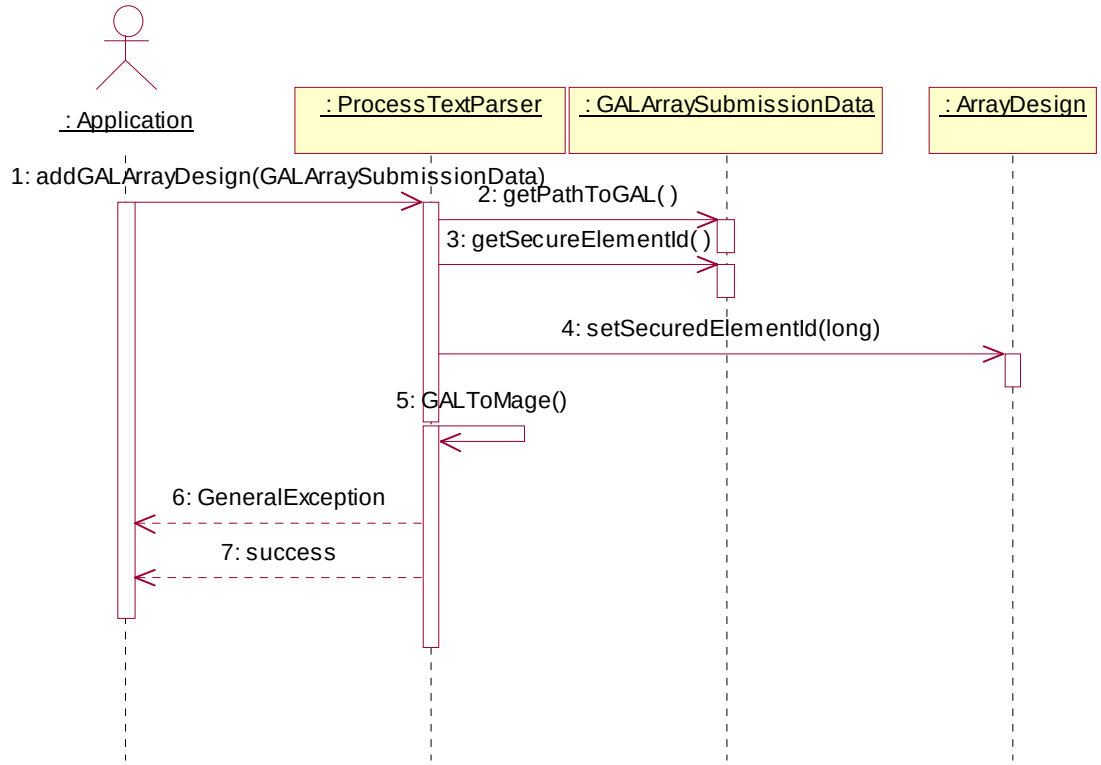


This diagram shows asynchronous processing of Mage design files on the server side.

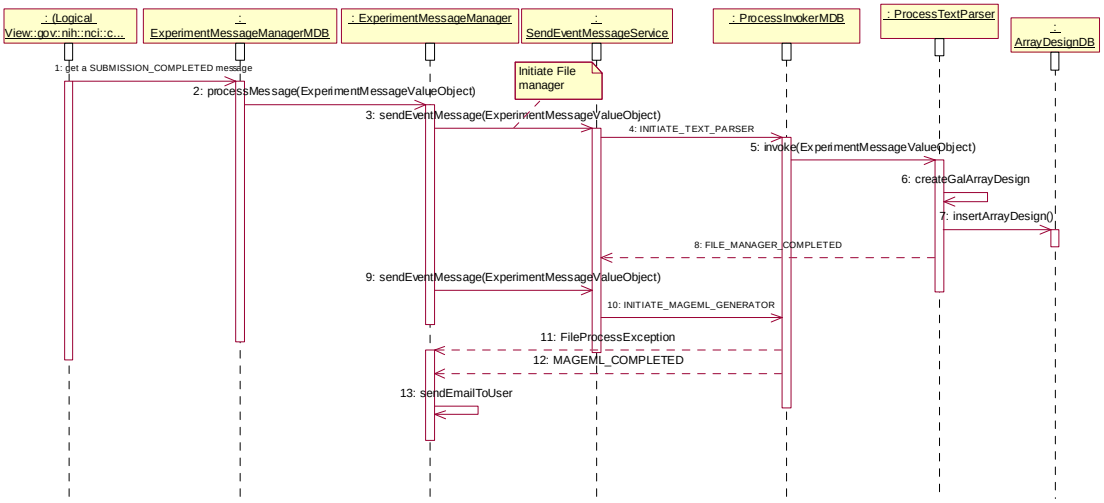
59. ArrayDesignManager.addGALArrayDesign



60. ArrayDesignManager.createGALArrayDesign



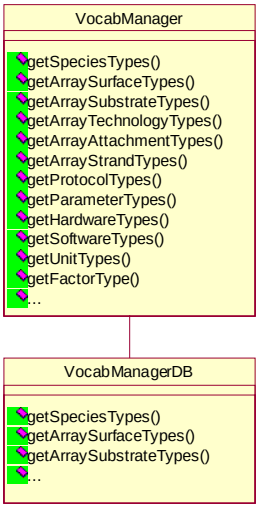
61. ArrayDesignManager.processGALArrayDesign

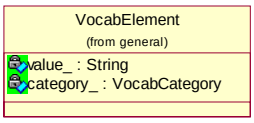


This diagram shows the sequence for processing GAL array design files. The parsing of

GAL files and mapping onto MAGE-OM objects would follow the GAL to MAGE-ML mapping created as part of the project. CreateGAL arrayDesign would accomplish that. The details of it could involve other delegate classes which is left as an implementation detail.

62. Logical View::gov::nih::nci::caarray::services::vocab





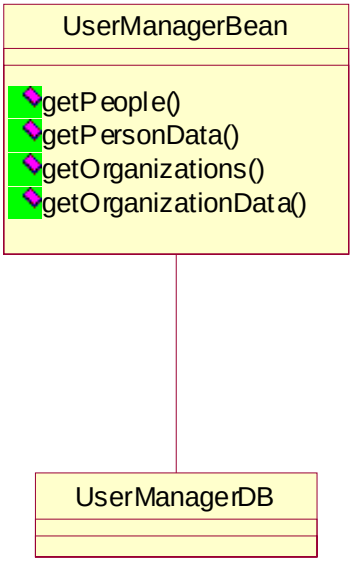
This class is used to model the available types of units of measurement. Examples are TemperatureUnit, MassUnit, etc.

62.1 VocabManager

62.2 VocabManagerDB

62.3

63. Logical View::gov::nih::nci::caarray::services::usermgmt

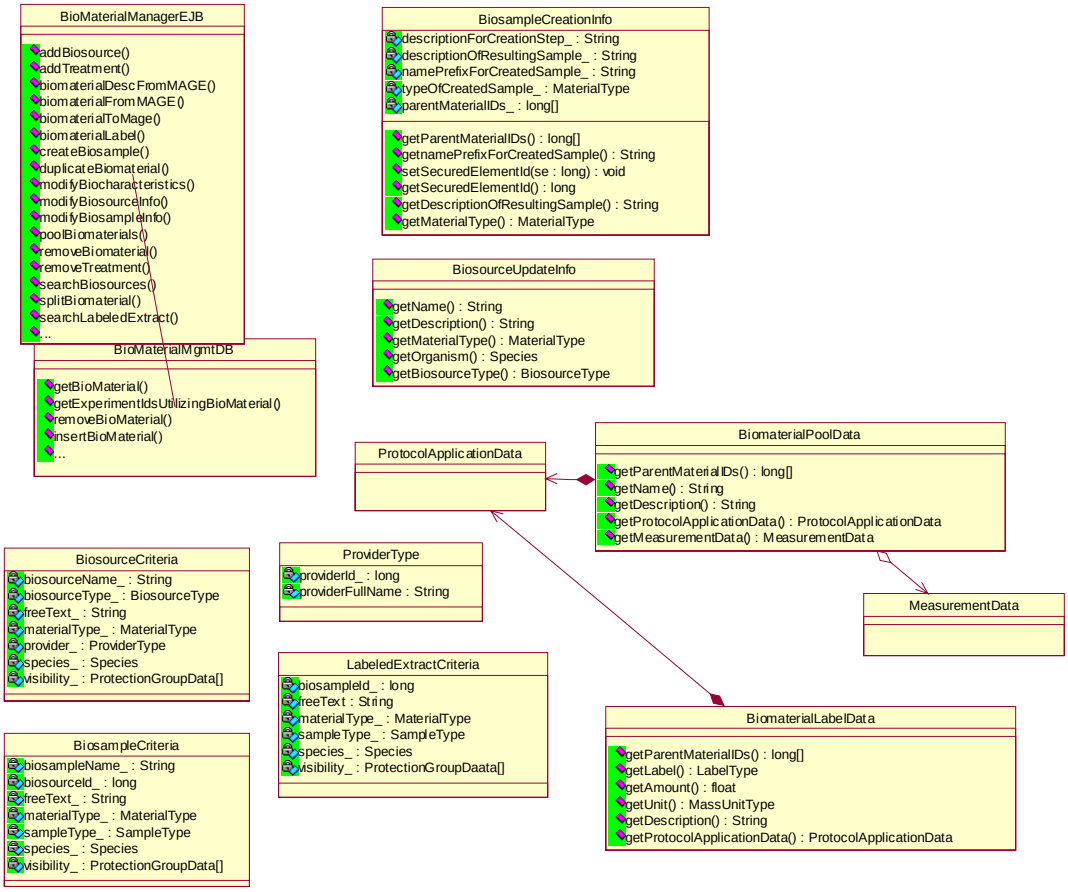


63.1 UserManagerBean

63.2 UserManagerDB

63.3

64. Logical View::gov::nih::nci::caarray::services::biomaterial



This diagram shows the static relationships between classes in the caarray.services.biomaterial package.

64.1 BioMaterialManagerEJB

64.2 BioMaterialMgmtDB

64.3 BiosourceCriteria

64.4 ProviderType

64.5 BioMaterialSearchCriteria

64.6 BiosampleCriteria

64.7 LabeledExtractCriteria

64.8 BiosampleCreationInfo

64.9 BiosourceUpdateInfo

64.10 ProtocolApplicationData

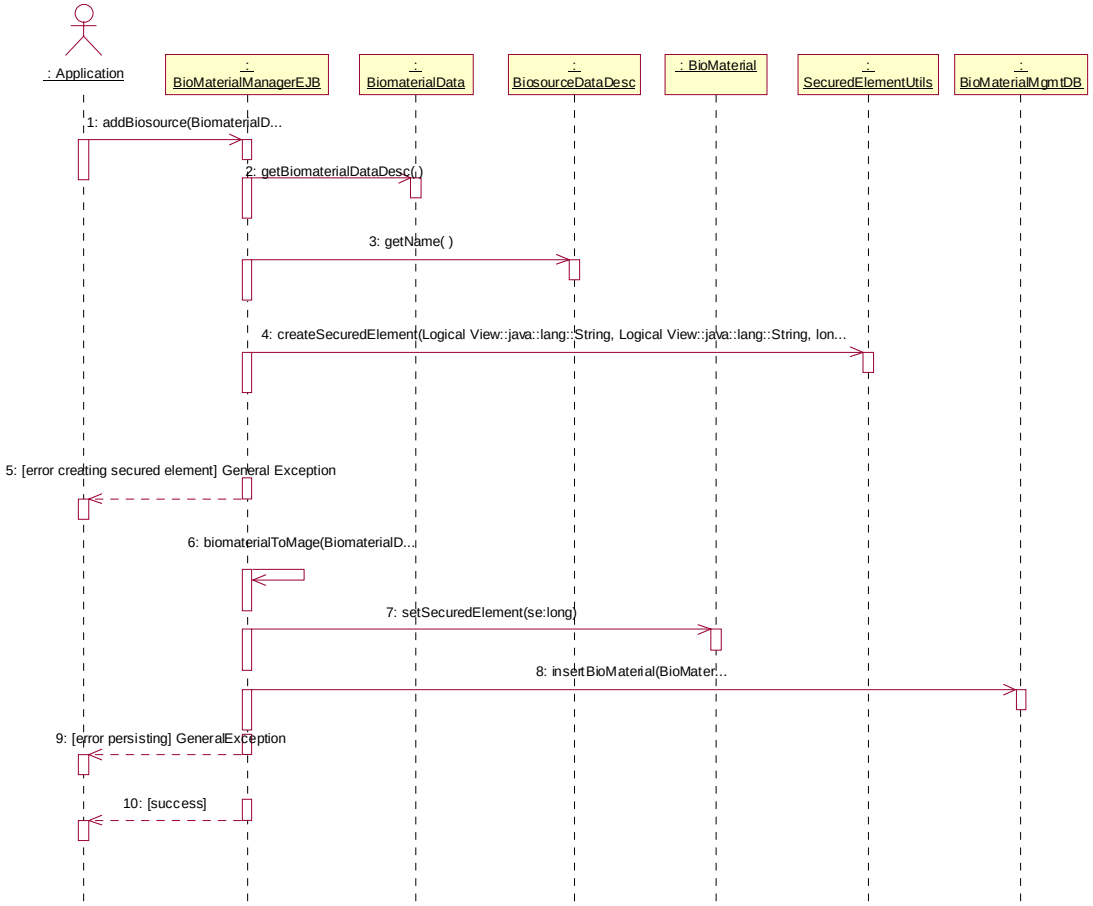
64.11 BiomaterialPoolData

64.12 MeasurementData

64.13 BiomaterialLabelData

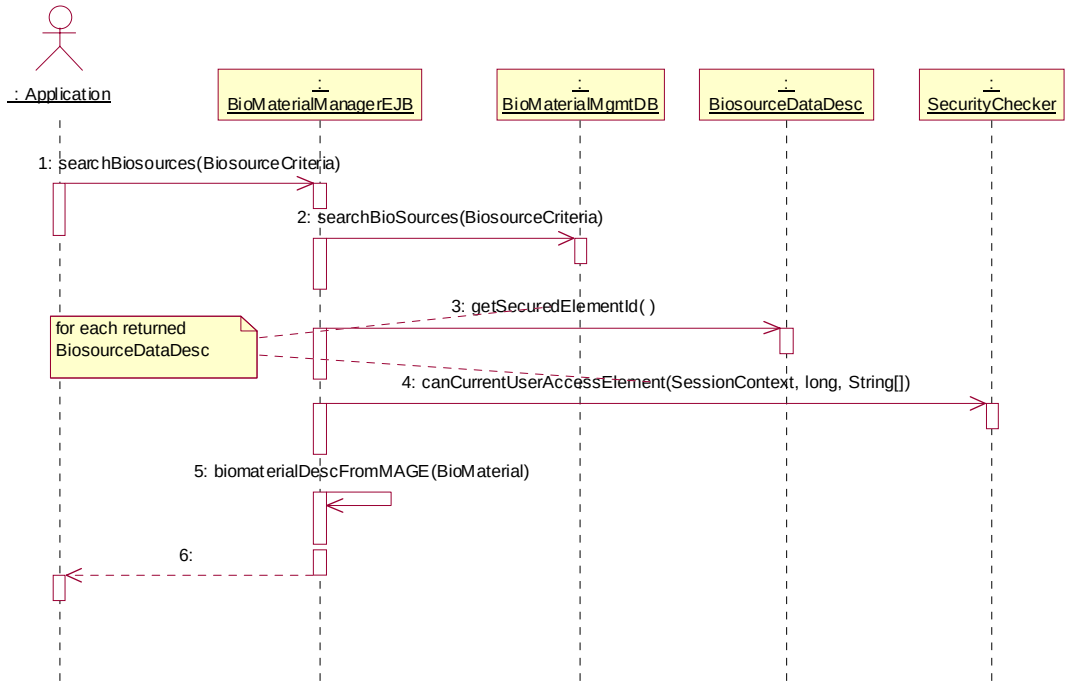
64.14

65. BiomaterialManager.addBiosource



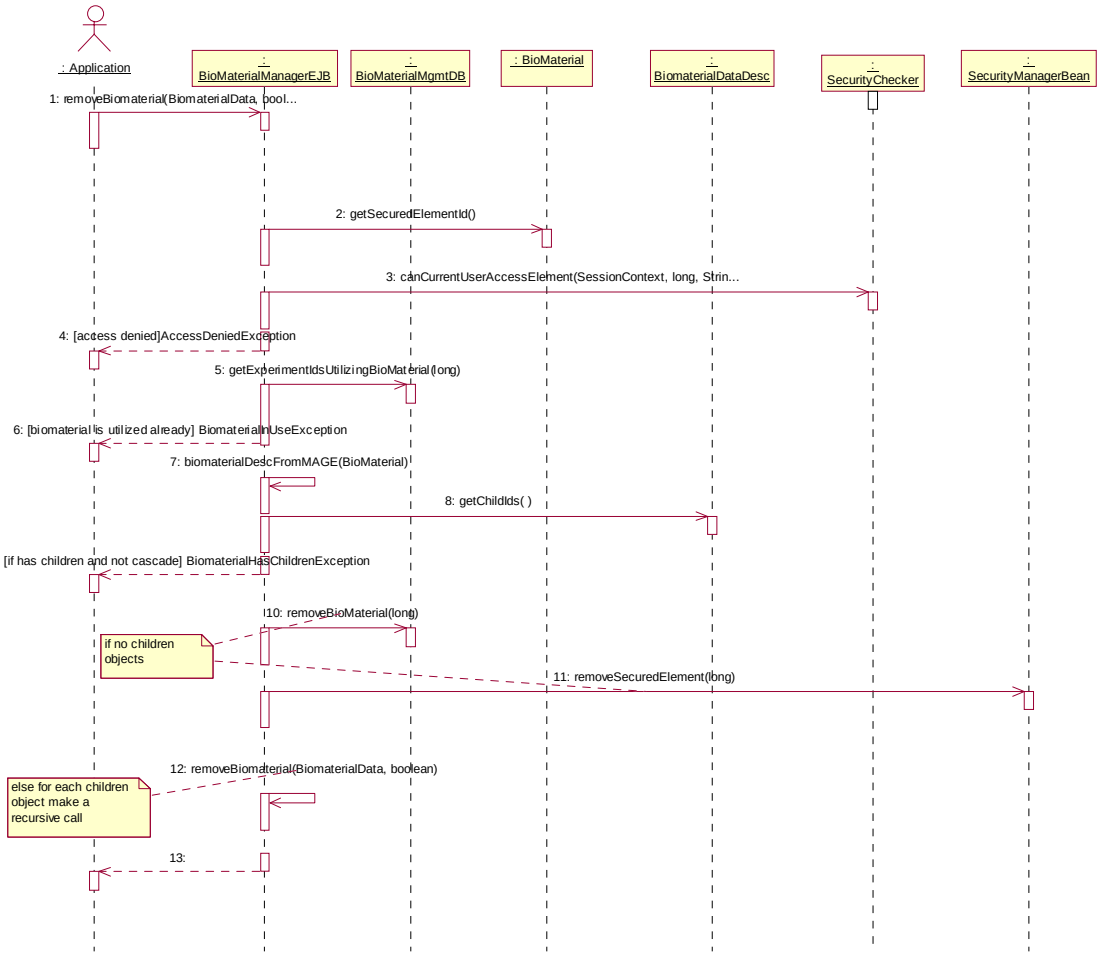
This diagram shows how a biosource is added to the system. Note: biosamples and labeled extracts are not added to the system directly; these are created from initial biosources by treatment events.

66. BiomaterialManager.searchBiomaterials



This diagram shows the processing performed when a user searches for biomaterials (biosources, biosamples, and labeled extracts). The specified search criteria are passed to the database. The database returns only those BioMaterial (BioSource, BioSample, or LabeledExtract) bioMAGE objects that meet the specified criteria. Each returned BioMaterial (BioSource, BioSample, or LabeledExtract) is checked to verify that the calling user can see it. A BiomaterialDataDesc (BiosourceDataDesc, BiosampleDataDesc, or LabeledExtractDataDesc) object is created for each BioMaterial that the caller is allowed to access. The collection of these objects is then returned to the caller. Since BioMaterial is an abstract class this sequence diagram shows a concrete class BioSource; sequence diagrams for BioSample and LabeledExtract are similar with a replacemenent of objects related to biosources with the corresponding objects for biosamples and labeled extracts.

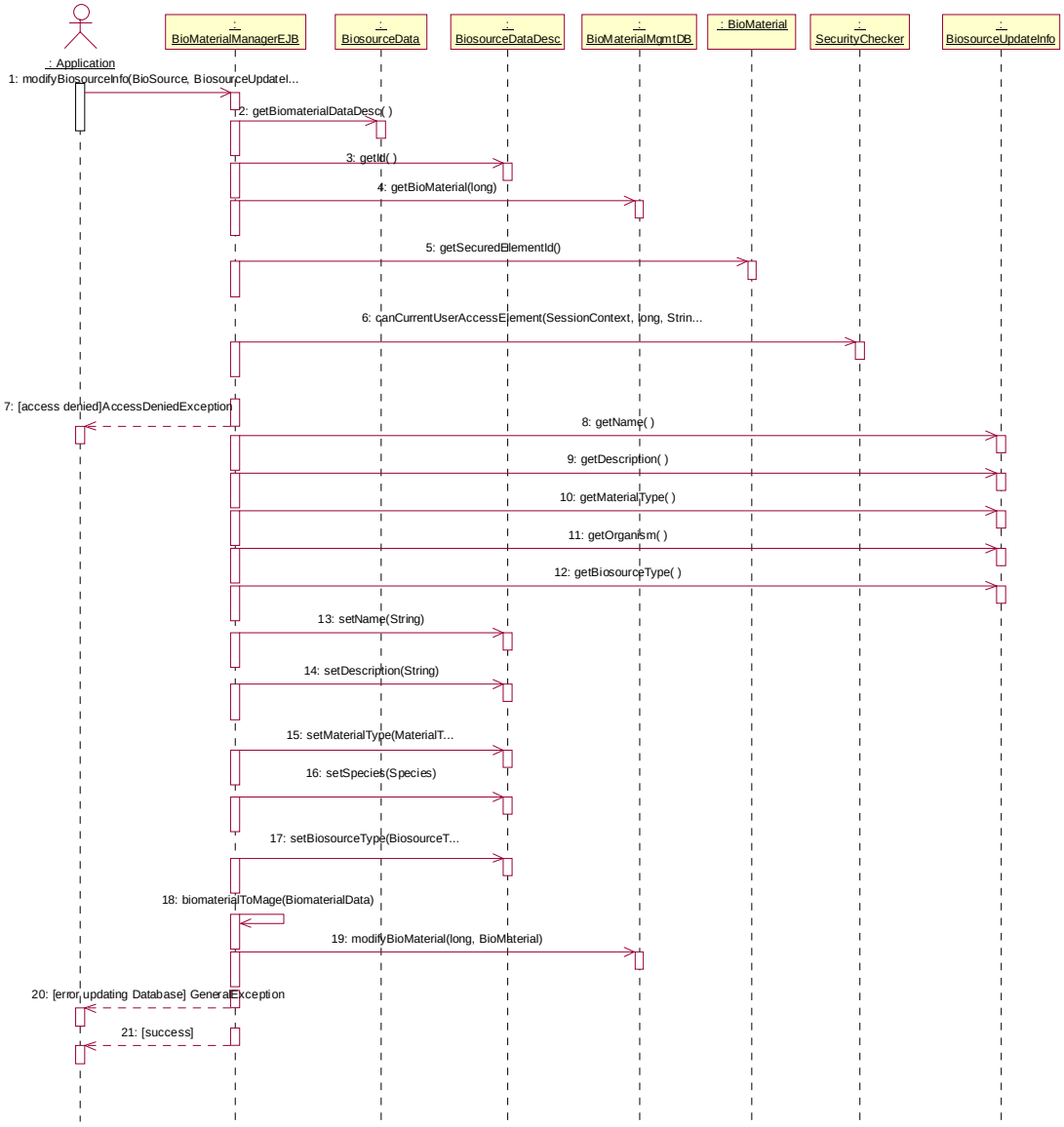
67. BiomaterialManager.removeBiomaterial



This diagram show the processing performed to remove a biomaterial from the system. A BioMaterial (BioSource, BioSample, or LabeledExtract) is retrieved from the database to obtain its security id. If the BioMaterial does not exist, an exception is thrown. Next the security service is called to determine if the current user can perform this operation on this biomaterial. If not, an AccessDeniedException is thrown. If the user can remove this biomaterial, a check is made to see if any experiments in the database reference this biomaterial. If they do an BiomaterialInUseException exception is thrown. This exception contains the identifiers of the experiments that are using the biomaterial. If the biomaterial is not in use, the final check is made if the biomaterial has child objects. If not it is removed from the database along with its secured element id. If child biomaterials exist and cascade is set to false

BiomaterialHasChildrenException will be thrown that will contain ids of child bimaterials. If cascade is set to true, removeBiomaterial will be recursively called on each of the child biomaterials, resulting in the cascade removal of all child biomaterials.

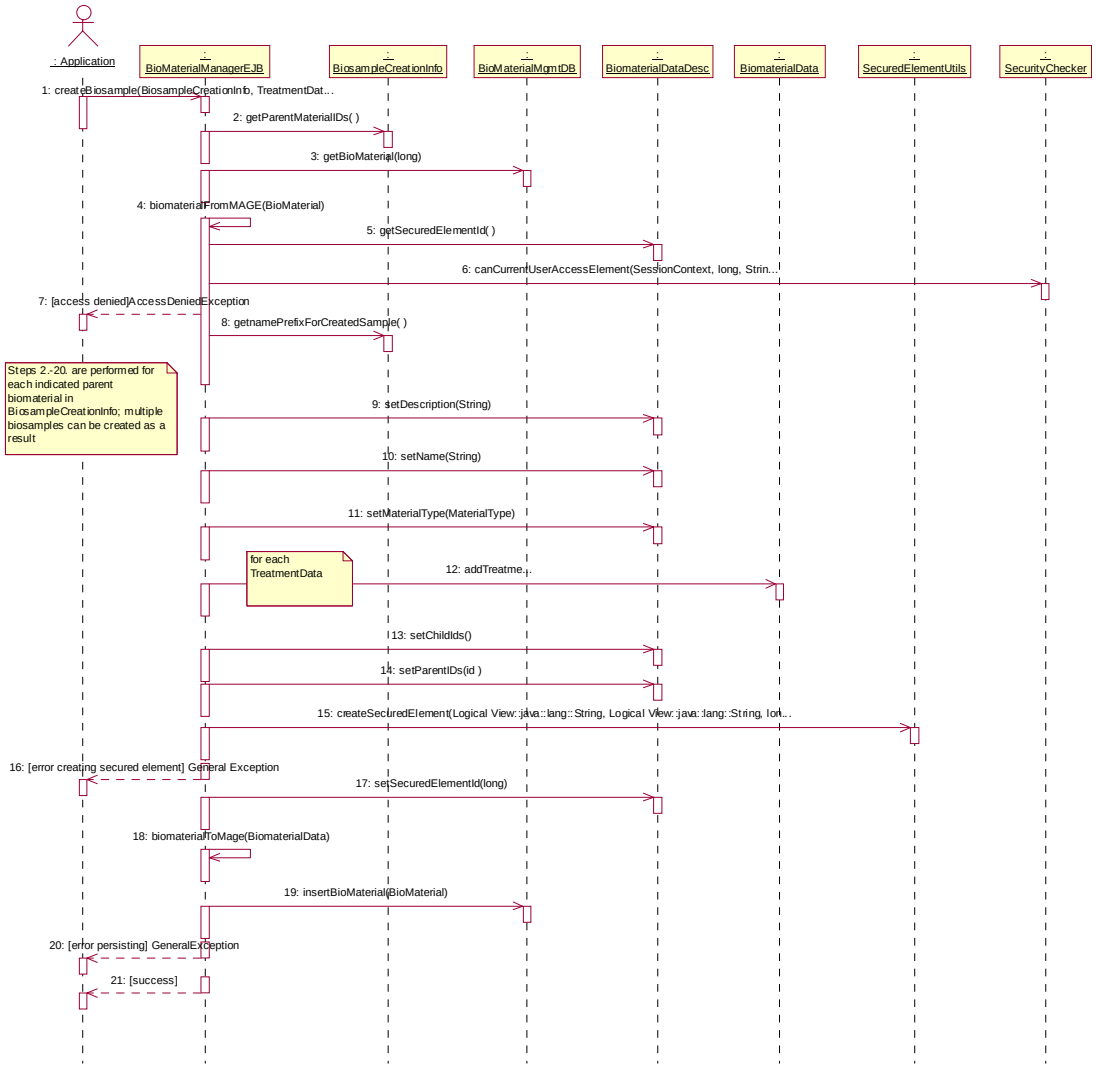
68. BiomaterialManager.modifyBiosourceInfo



This diagram shows how biomaterial information, including general biomaterial information, visibility, and biocharacteristics

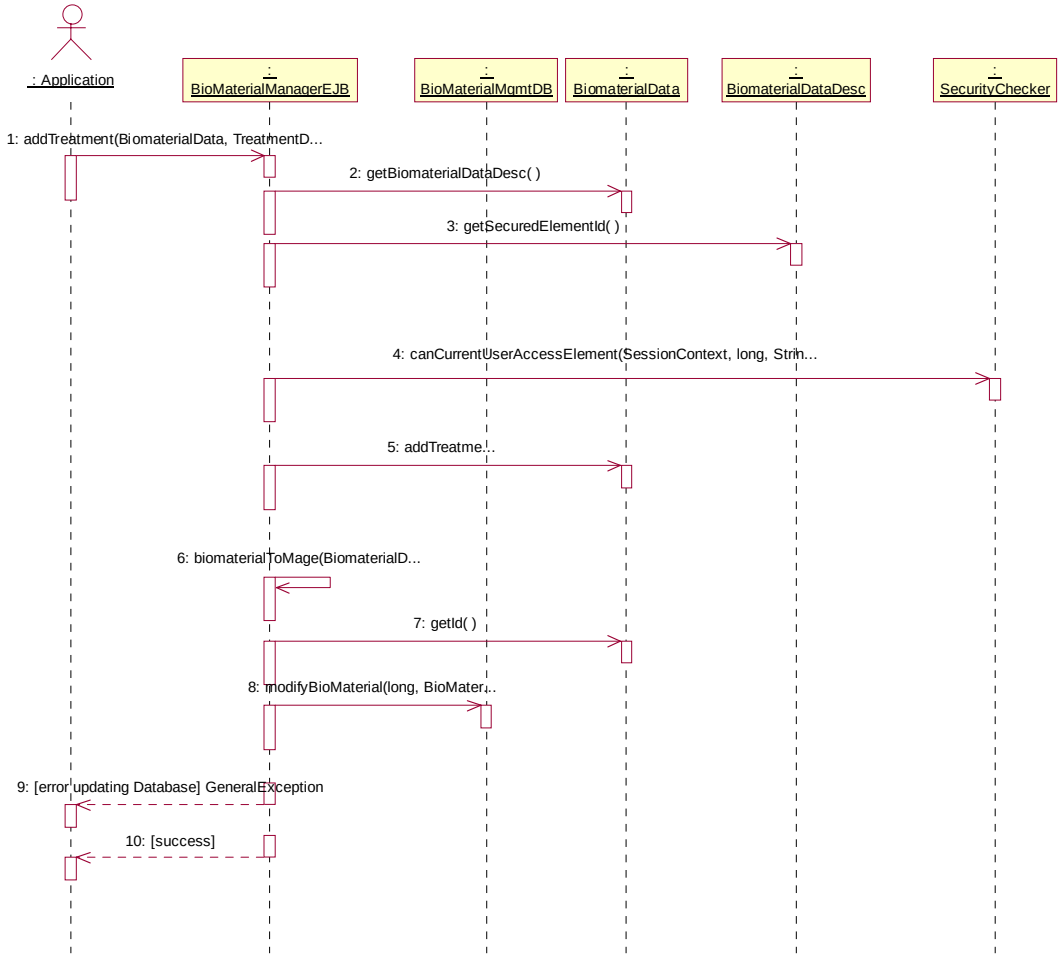
are modified. This sequence diagram shows the sequence for modification of biosource information, but the same sequence would be applicable for biosamples and labeled extracts.

69. BiomaterialManager.createBiosample

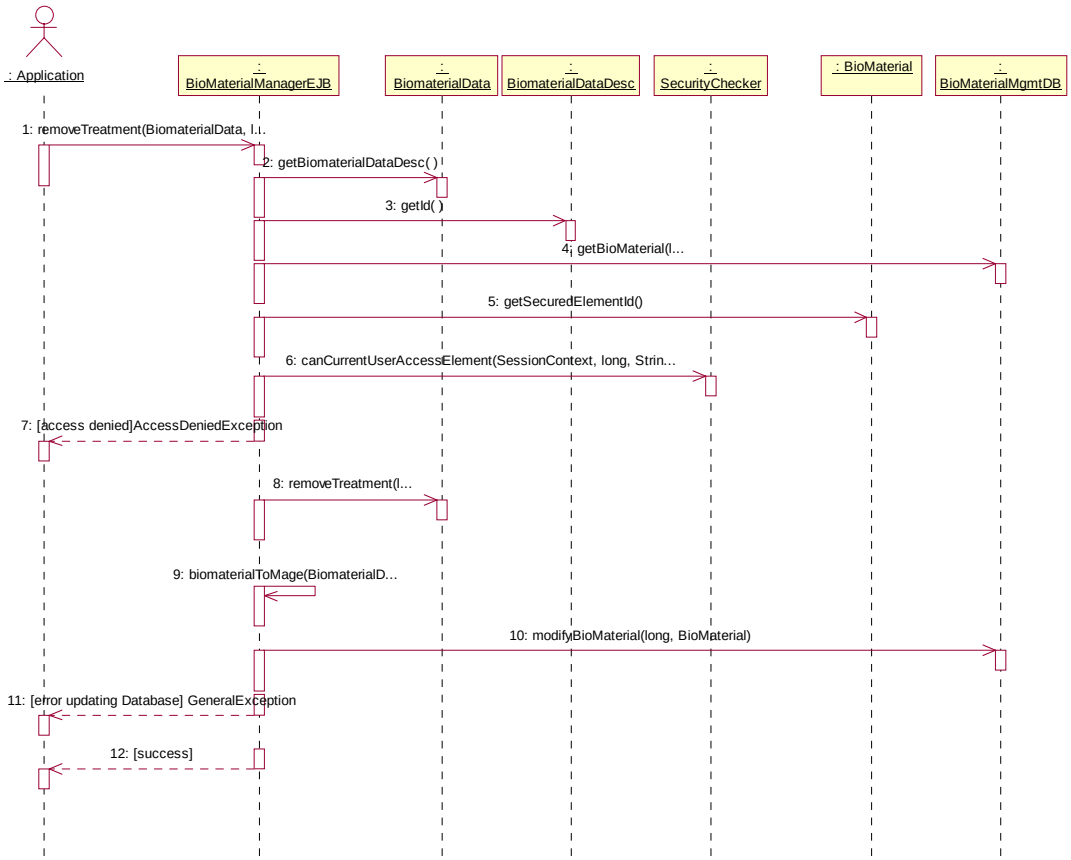


This diagram shows how a biosamle is created in the system given parent biomaterials. Biocharacteristics are copied from the parent biomaterial.

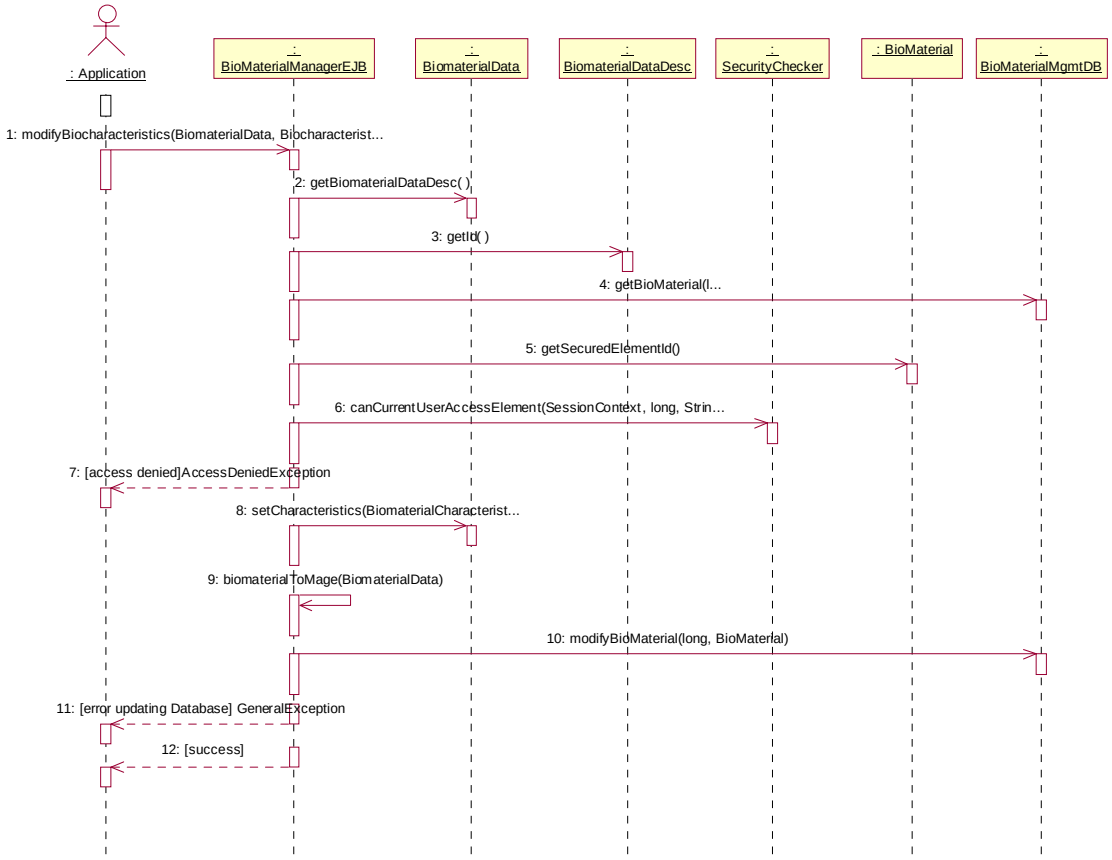
70. BiomaterialManager.addTreatment



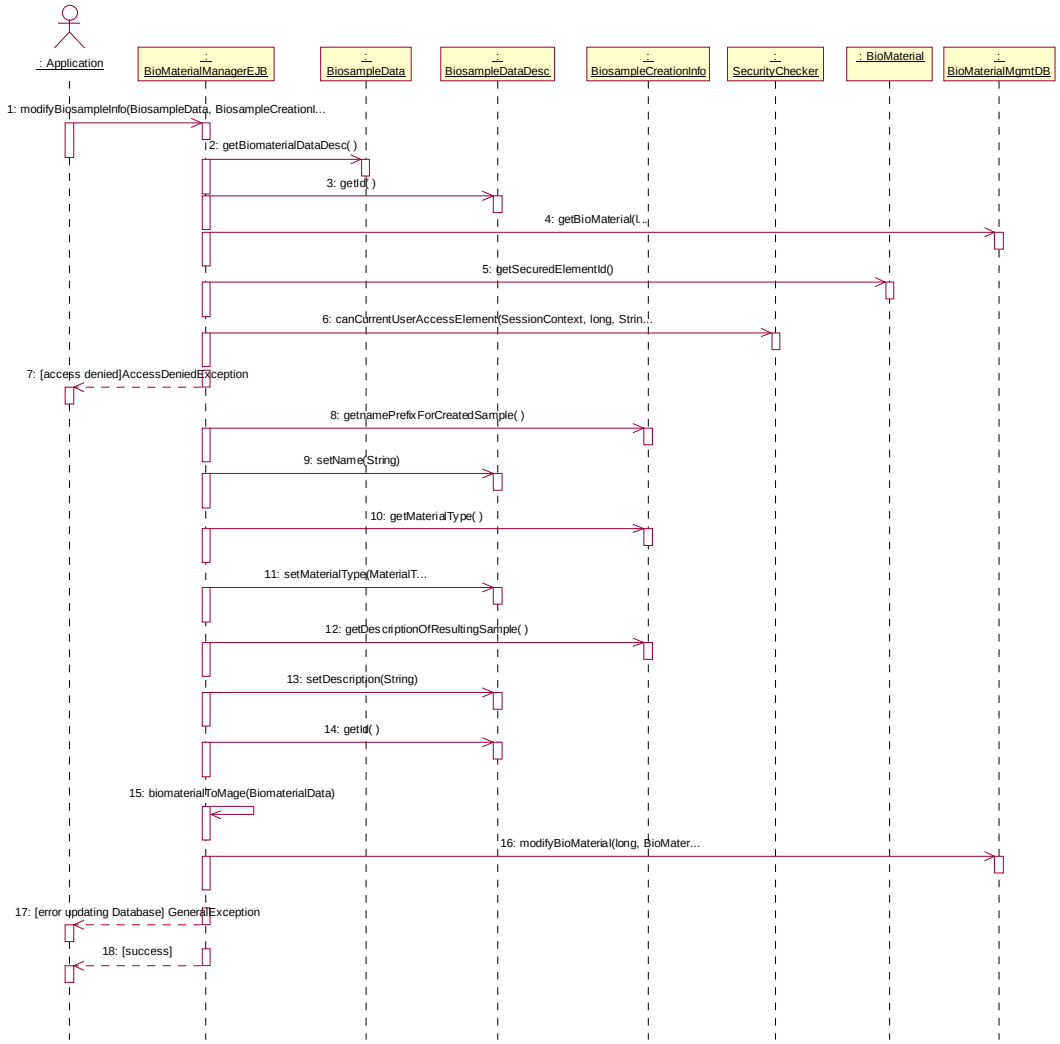
71. BiomaterialManager.removeTreatment



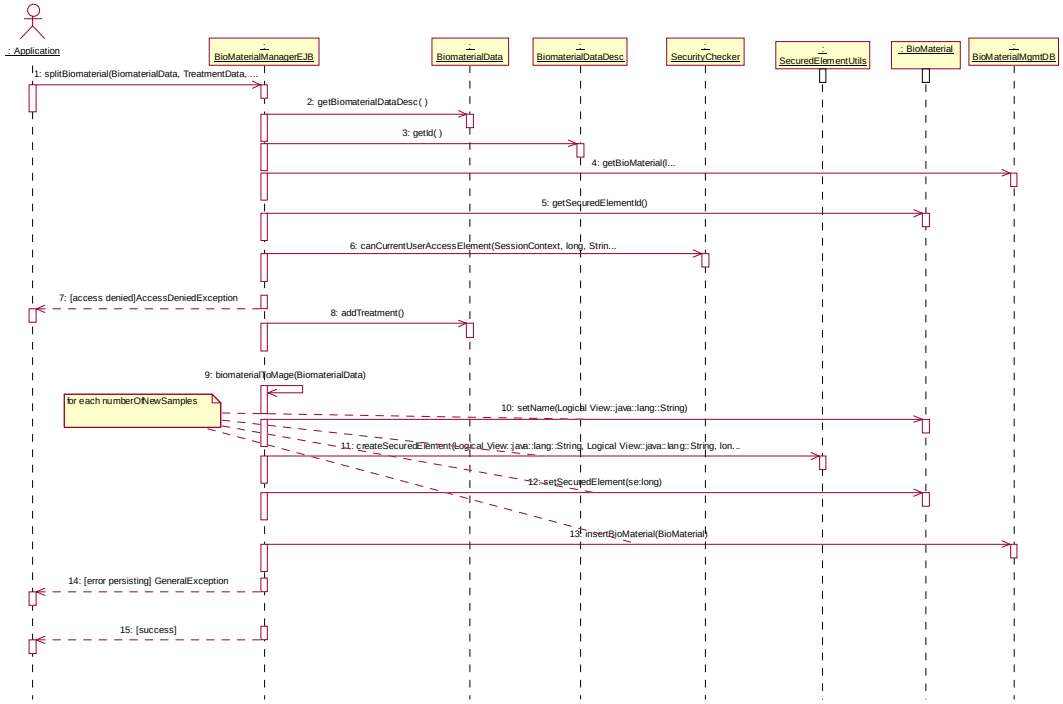
72. BiomaterialManager.modifyBiocharacteristics



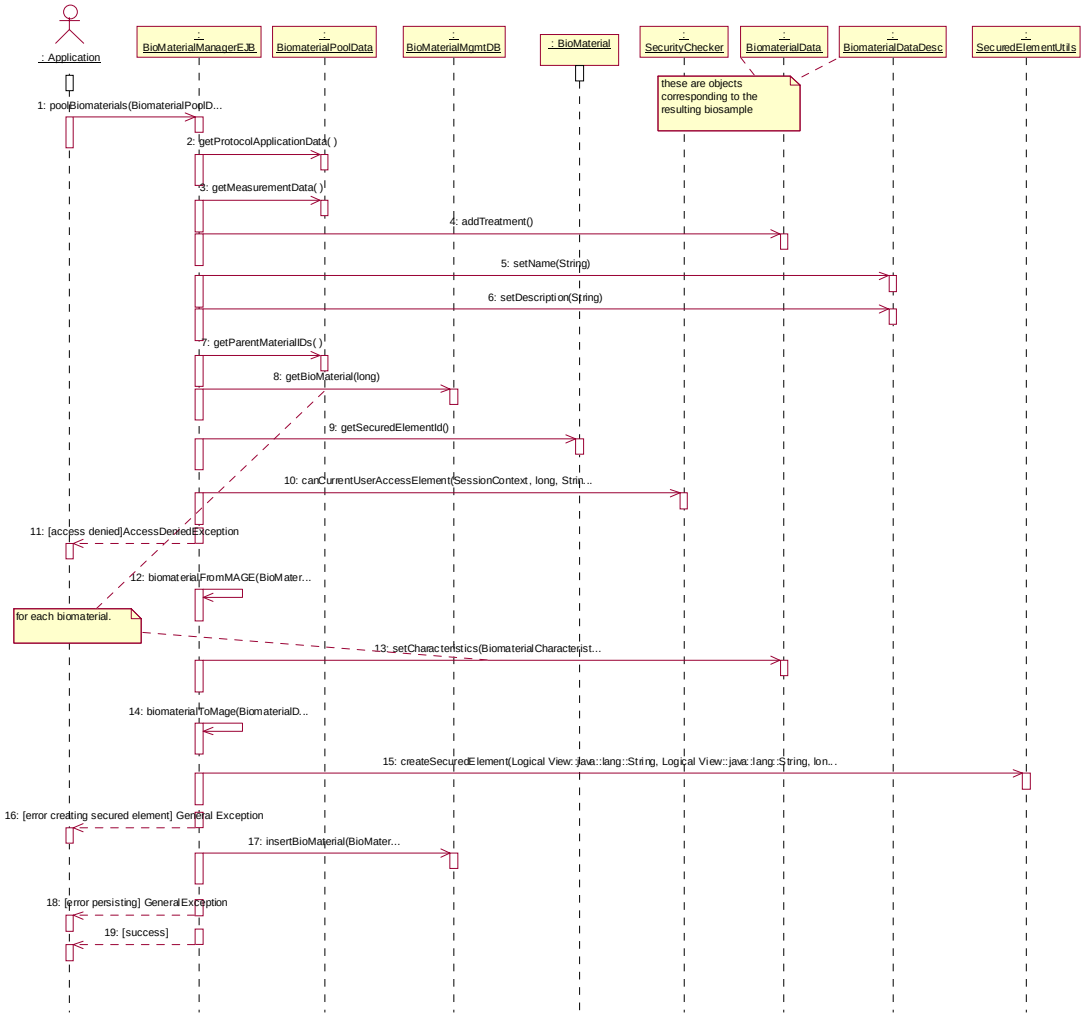
73. BiomaterialManager.modifyBisampleInfo



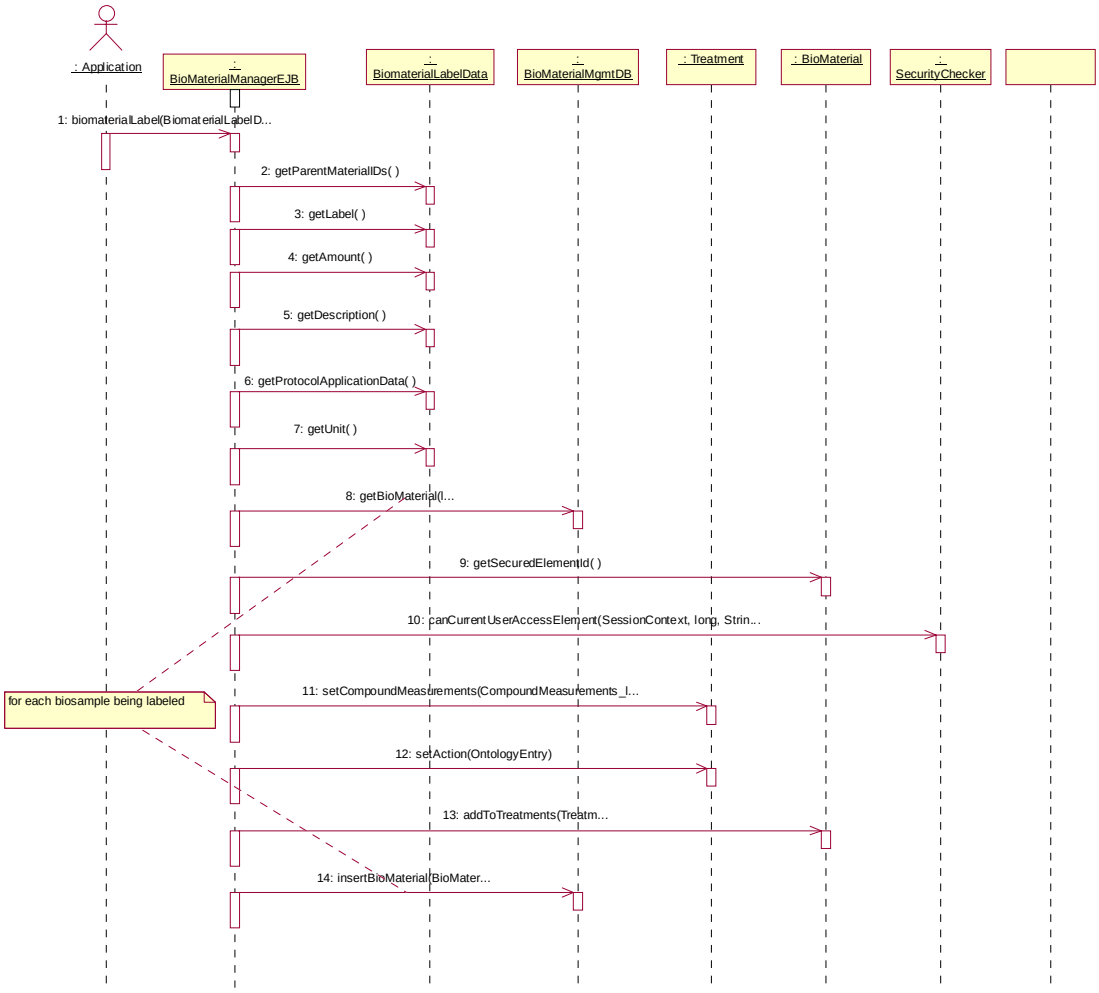
74. BiomaterialManager.splitBiomaterial



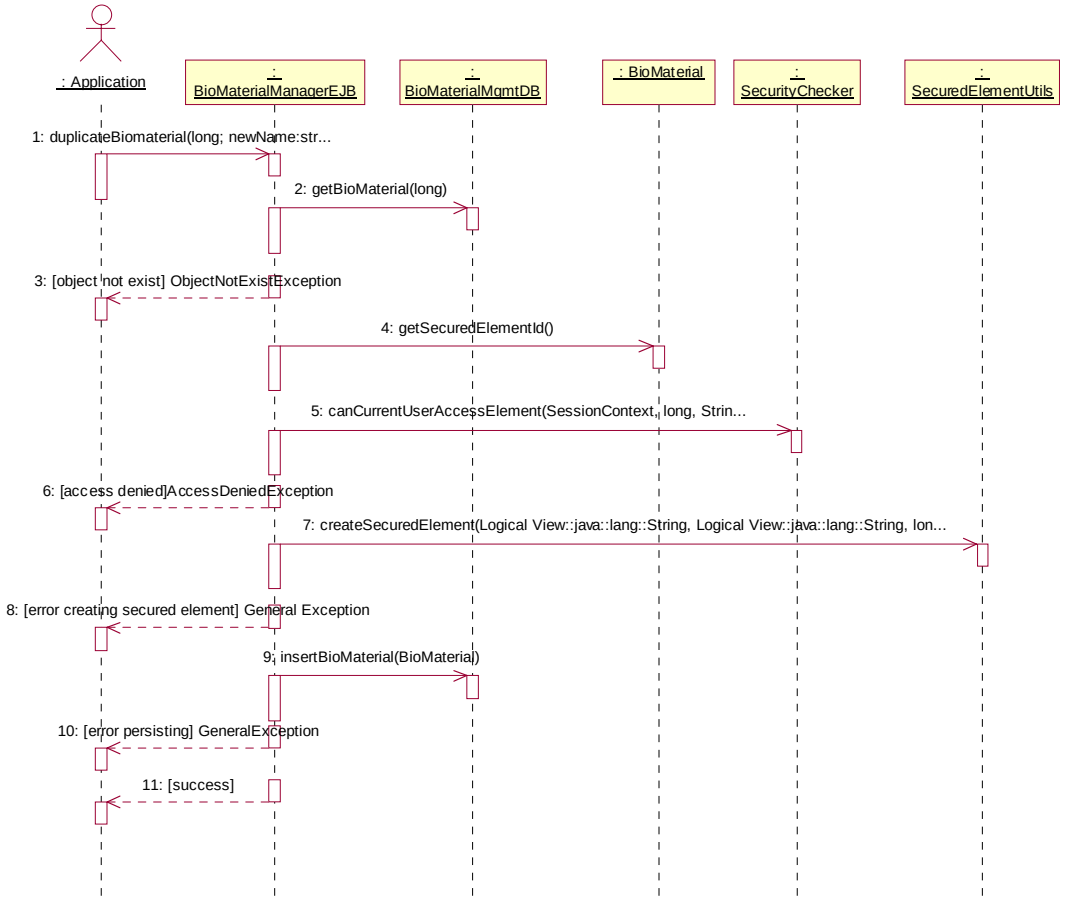
75. BiomaterialManager.poolBiomaterials



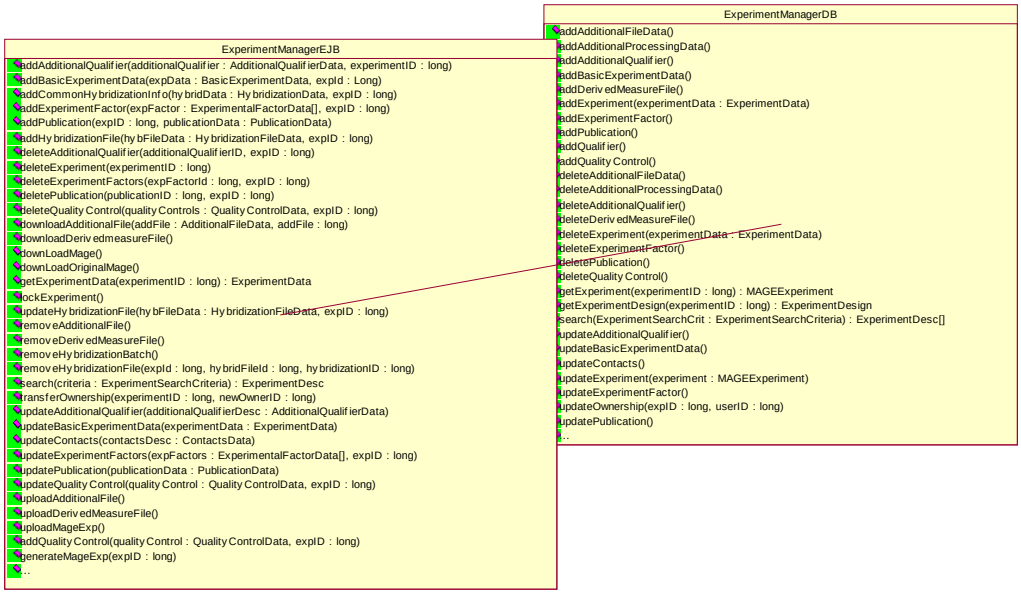
76. BiomaterialManager.biomaterialLabel

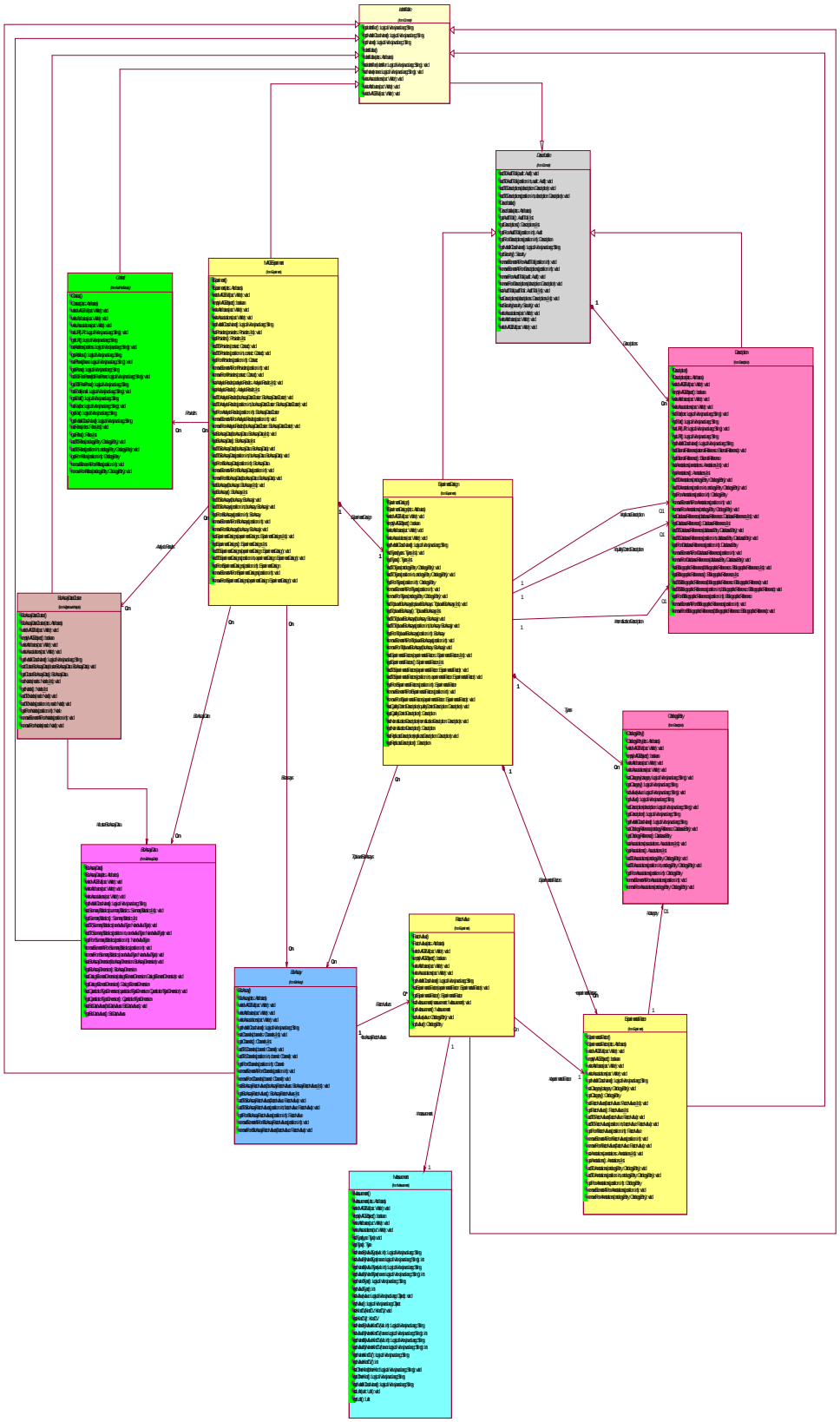


77. BiomaterialManager.duplicateBiomaterial



78. Logical View::gov::nih::nci::caarray::services::experiment


ExperimentManagerEJB
addAdditionalQualifier(additionalQualifier : AdditionalQualifierData, experimentID : long)
addBasicExperimentData(expData : BasicExperimentData, expId : Long)
addCommonHybridizationInfo(hybridData : HybridizationData, expID : long)
addExperimentFactor(expFactor : ExperimentalFactorData[], expID : long)
addPublication(expID : long, publicationData : PublicationData)
addHybridizationFile(hybFileData : HybridizationFileData, expID : long)
deleteAdditionalQualifier(additionalQualifierID, expID : long)
deleteExperiment(experimentID : long)
deleteExperimentFactors(expFactorId : long, expID : long)
deletePublication(publicationID : long, expID : long)
deleteQualityControl(qualityControls : QualityControlData, expID : long)
downloadAdditionalFile(addFile : AdditionalFileData, addFile : long)
downloadDerivedmeasureFile()
downLoadMage()
downLoadOriginalMage()
getExperimentData(experimentID : long) : ExperimentData
lockExperiment()
updateHybridizationFile(hybFileData : HybridizationFileData, expID : long)
removeAdditionalFile()
removeDerivedMeasureFile()
removeHybridizationBatch()
removeHybridizationFile(expId : long, hybridFileId : long, hybridizationID : long)
search(criteria : ExperimentSearchCriteria) : ExperimentDesc
transferOwnership(experimentID : long, newOwnerID : long)
updateAdditionalQualifier(additionalQualifierDesc : AdditionalQualifierData)
updateBasicExperimentData(experimentData : ExperimentData)
updateContacts(contactsDesc : ContactsData)
updateExperimentFactors(expFactors : ExperimentalFactorData[], expID : long)
updatePublication(publicationData : PublicationData)
updateQualityControl(qualityControl : QualityControlData, expID : long)
uploadAdditionalFile()
uploadDerivedMeasureFile()
uploadMageExp()
addQualityControl(qualityControl : QualityControlData, expID : long)
generateMageExp(expID : long)
...
ExperimentManagerDB
addAdditionalFileData()
addAdditionalProcessingData()
addAdditionalQualifier()
addBasicExperimentData()
addDerivedMeasureFile()
addExperiment(experimentData : ExperimentData)
addExperimentFactor()
addPublication()
addQualifier()
addQualityControl()
deleteAdditionalFileData()
deleteAdditionalProcessingData()
deleteAdditionalQualifier()
deleteDerivedMeasureFile()
deleteExperiment(experimentData : ExperimentData)
deleteExperimentFactor()
deletePublication()
deleteQualityControl()
getExperiment(experimentID : long) : MAGEExperiment
getExperimentDesign(experimentID : long) : ExperimentDesign
search(ExperimentSearchCrit : ExperimentSearchCriteria) : ExperimentDesc[]
updateAdditionalQualifier()
updateBasicExperimentData()
updateContacts()
updateExperiment(experiment : MAGEExperiment)
updateExperimentFactor()
updateOwnership(expID : long, userID : long)
updatePublication()
...



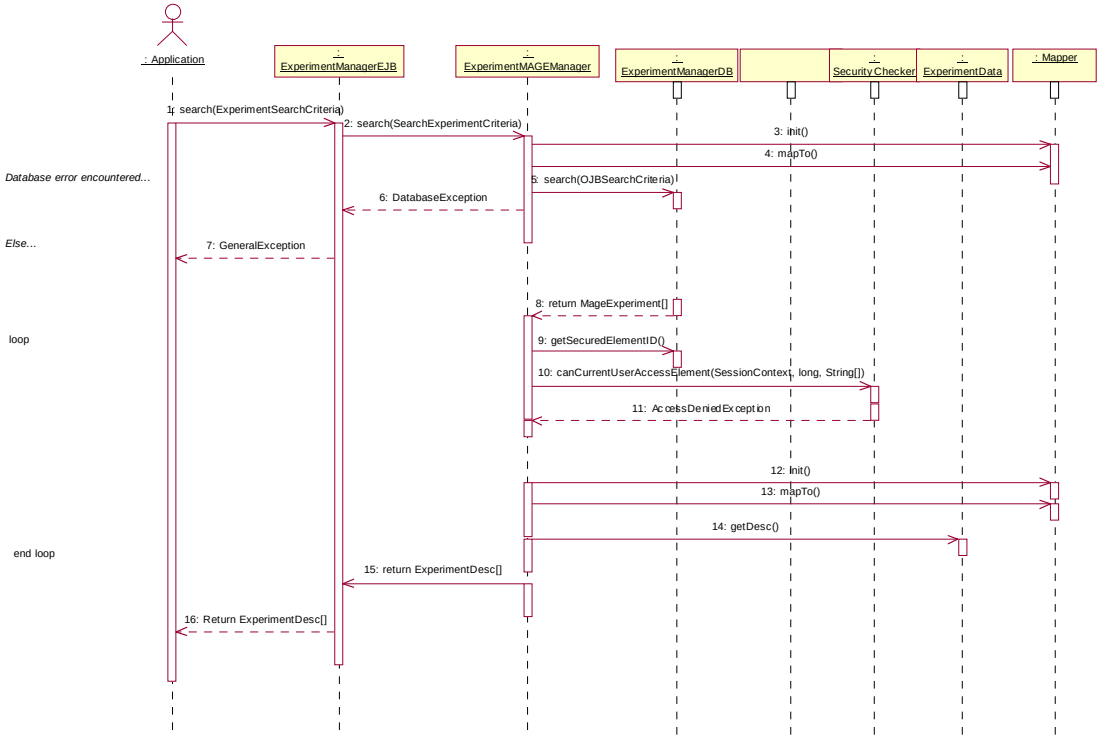
78.1 ExperimentManagerEJBThis is the EJB called by clients to accomplish tasks related to Experiment management. The actual uploaded data gets transferred directly to the ftp site.

78.2 ExperimentMAGEManager

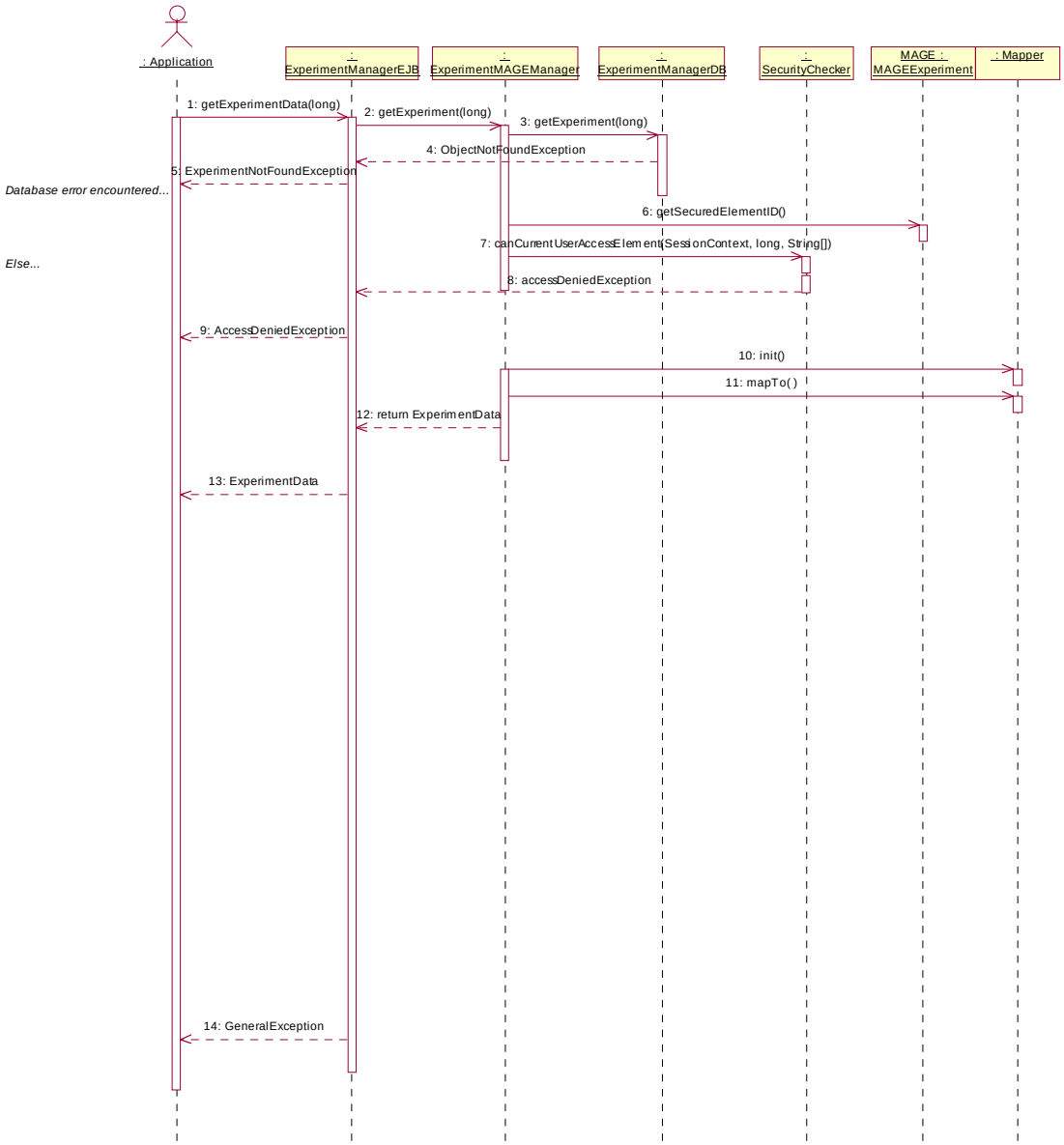
78.3 ExperimentManagerDBThis class is the DAO associated with the ExperimentManagerEJB. It encapsulates the database access logic and gets called by the EJB.

78.4

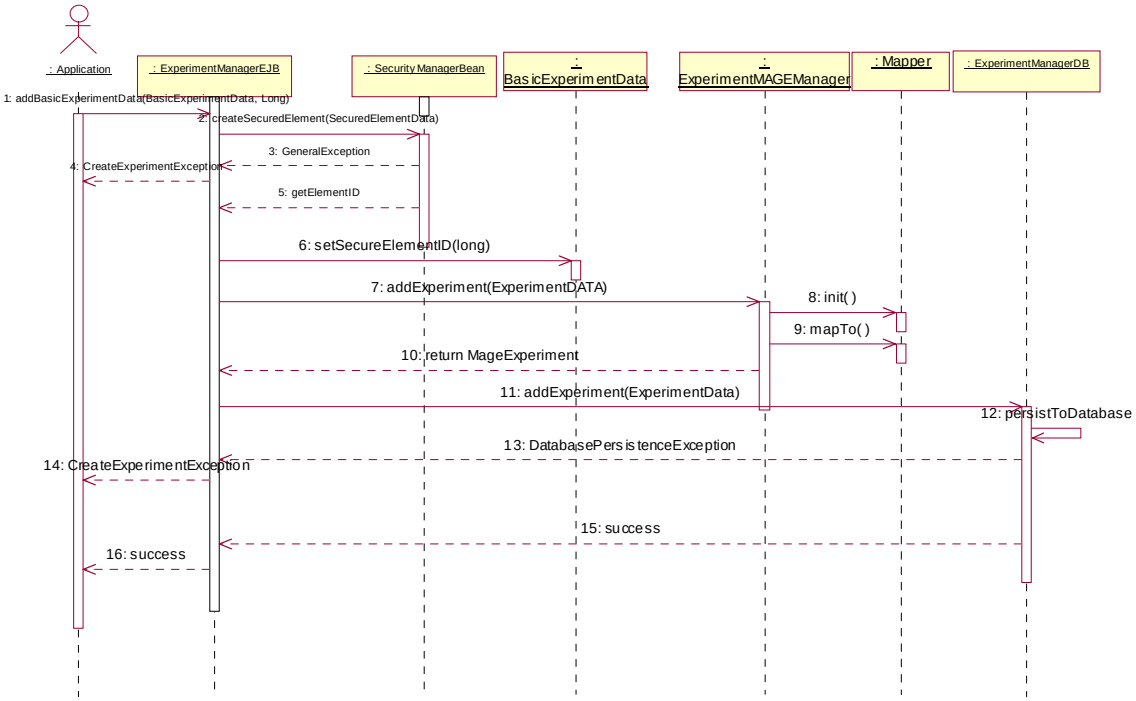
79. ExperimentManager.search



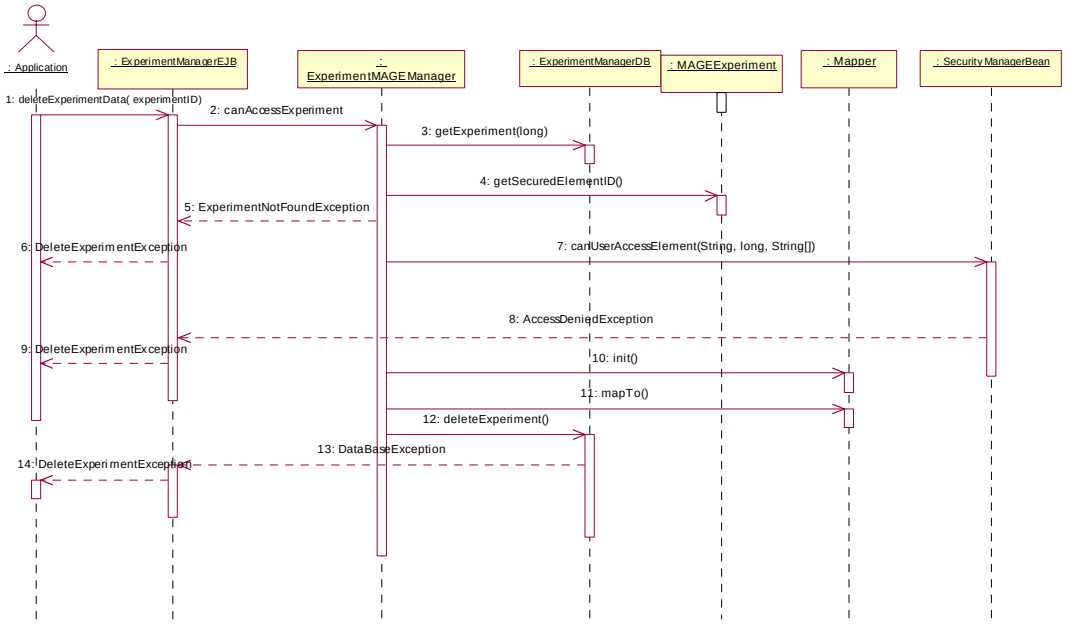
80. ExperimentManager.getExperimentData



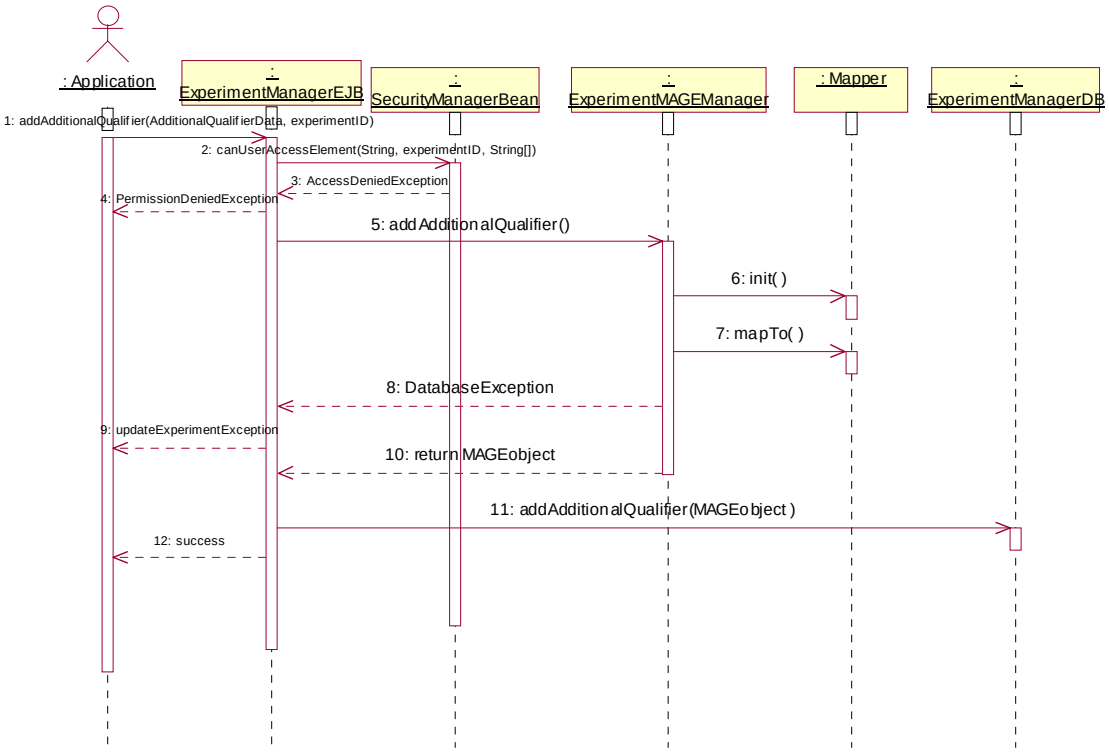
81. ExperimentManager.addExperiment



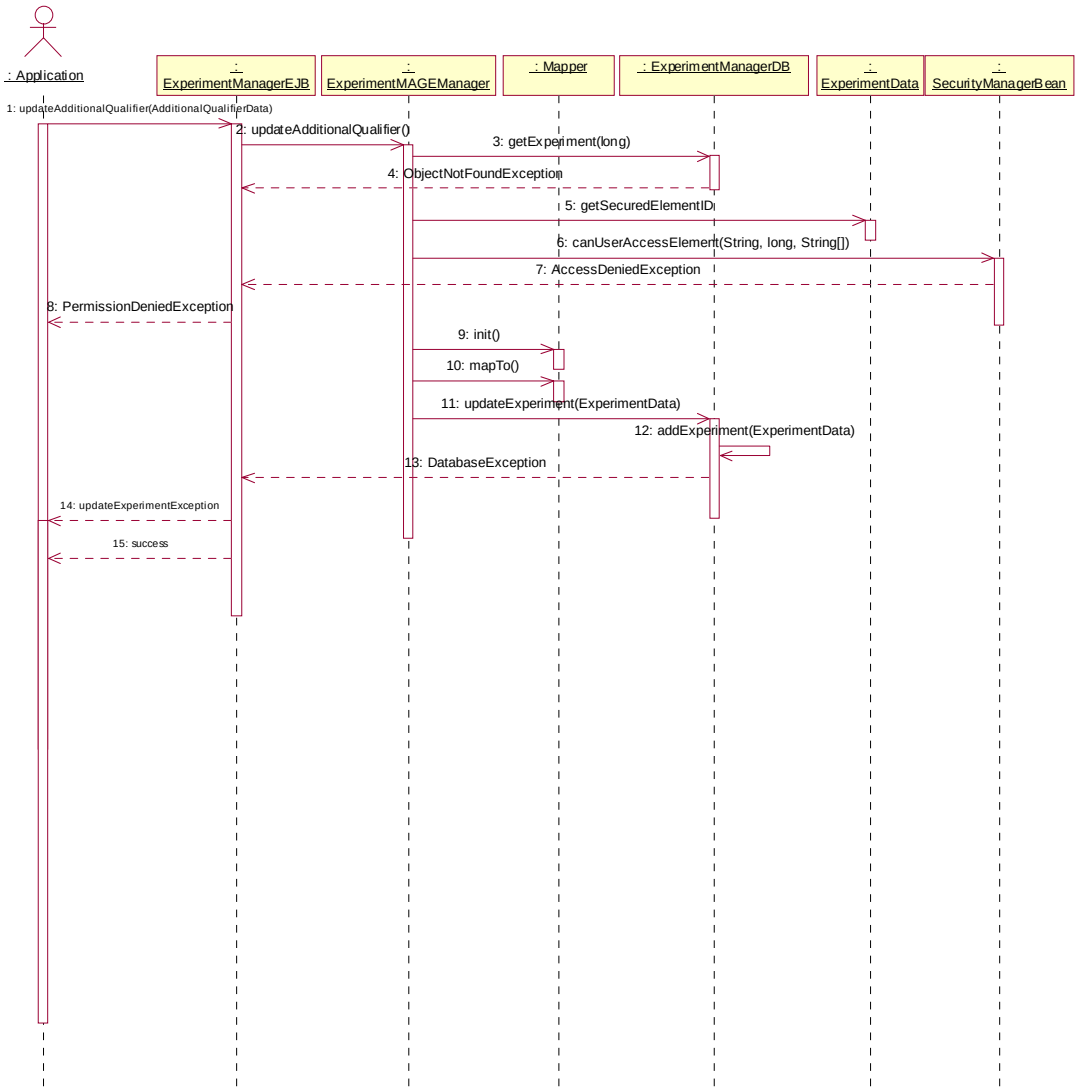
82. ExperimentManager.deleteExperiment



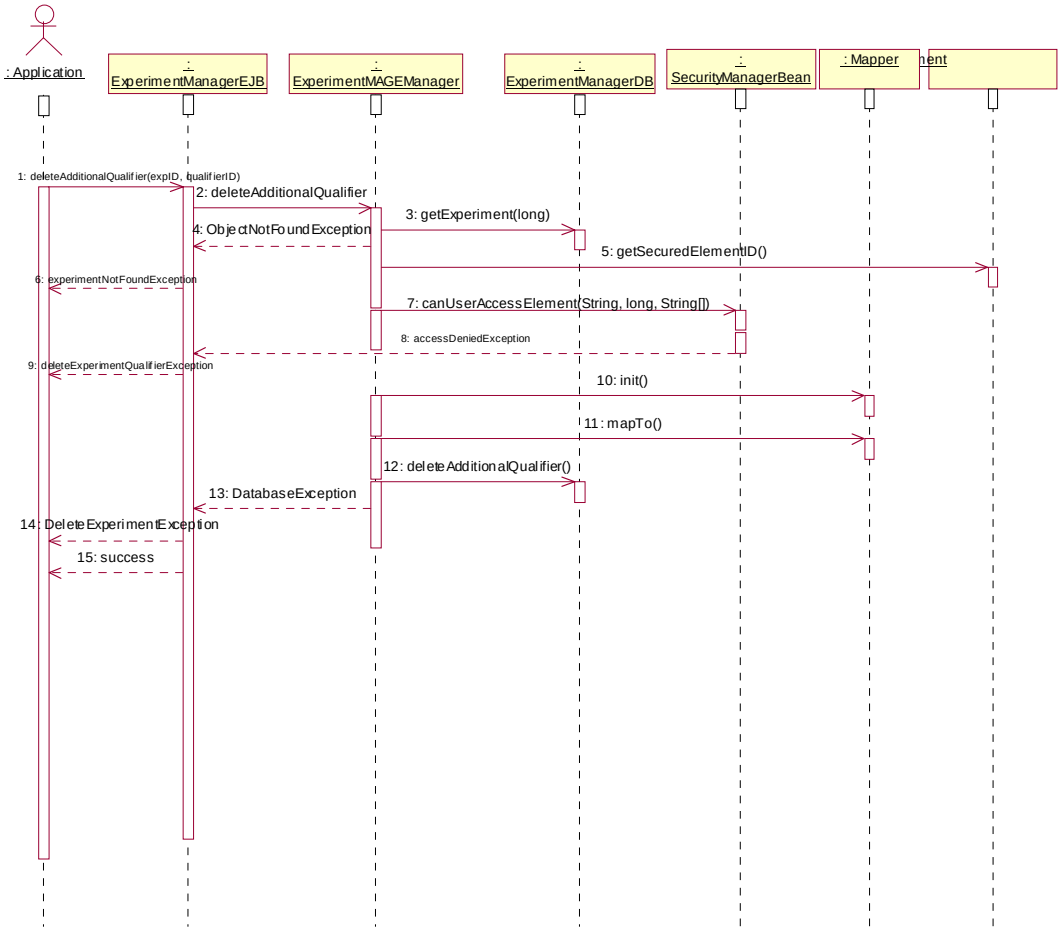
83. ExperimentManager.addAdditionalQualifier



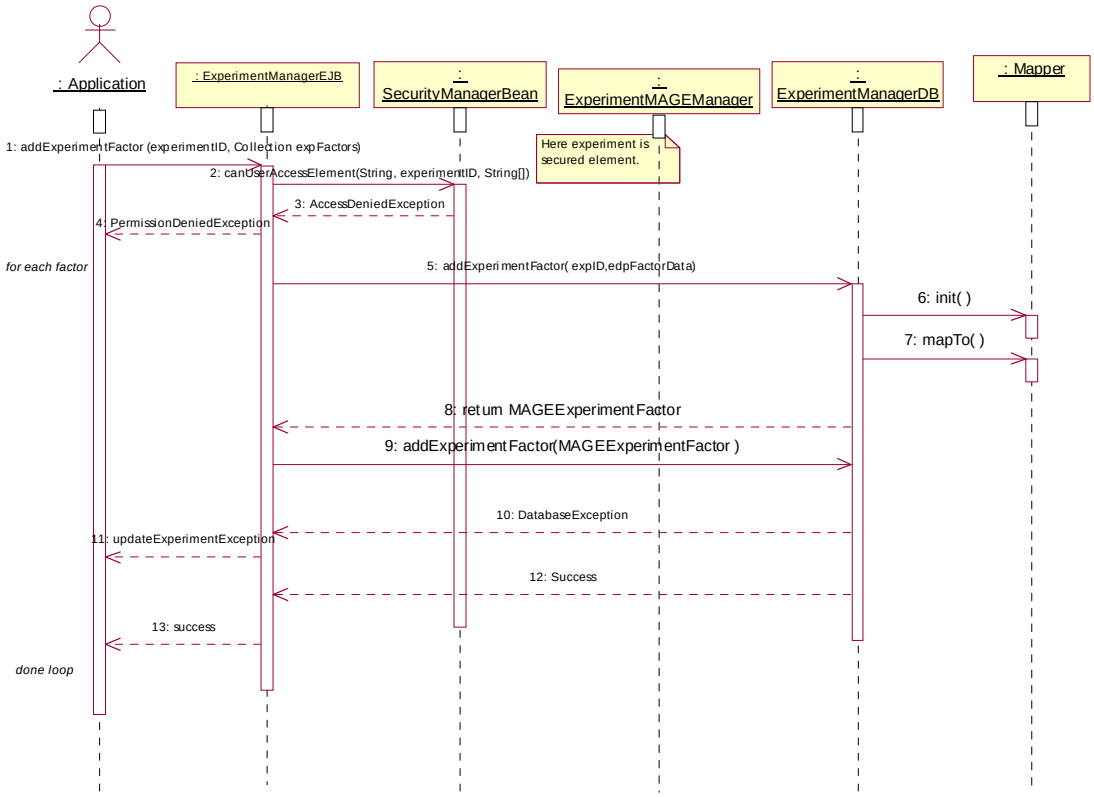
84. ExperimentManager.updateAdditionalQualifier



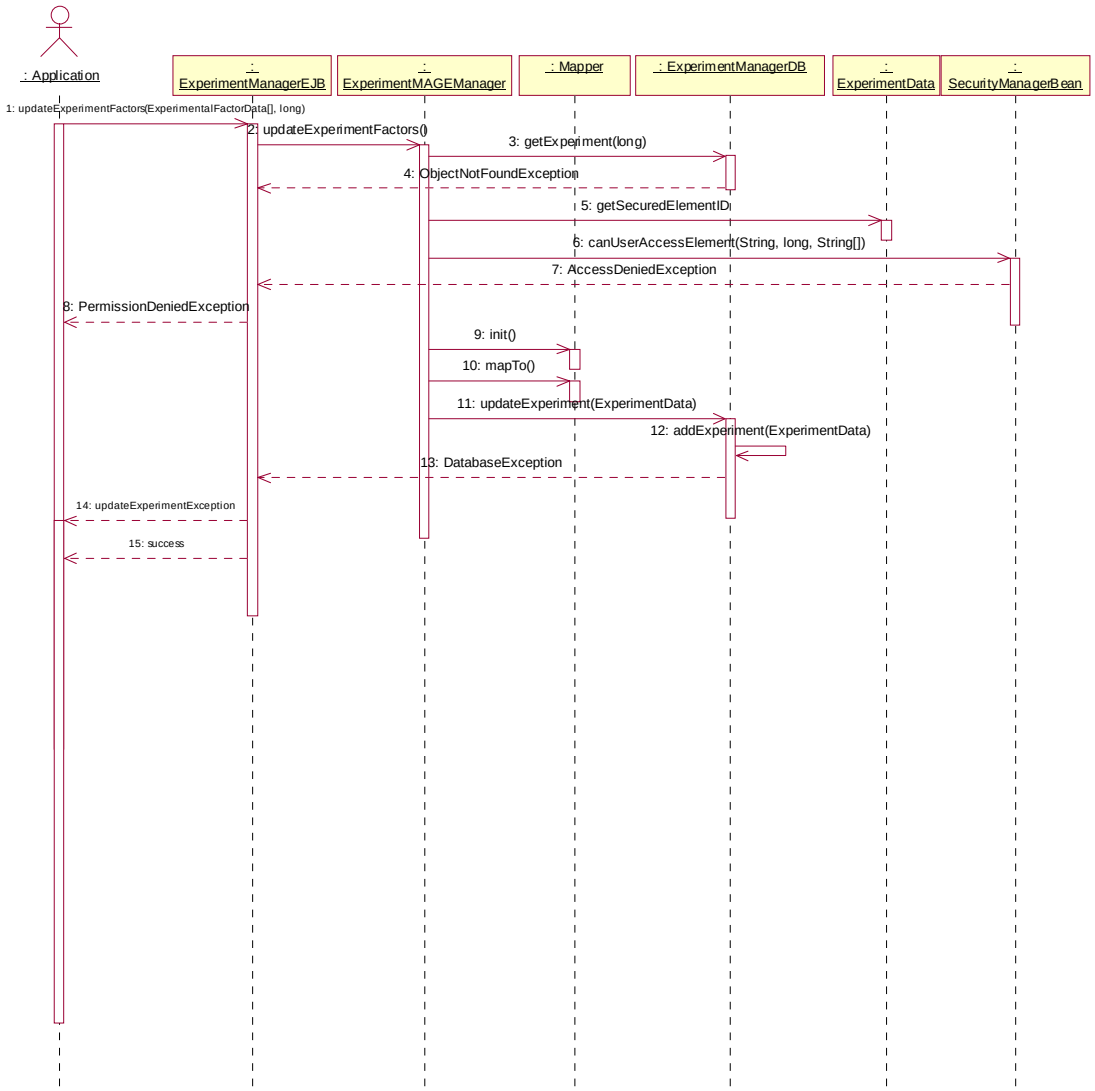
85. ExperimentManager.deleteAdditionalQualifier



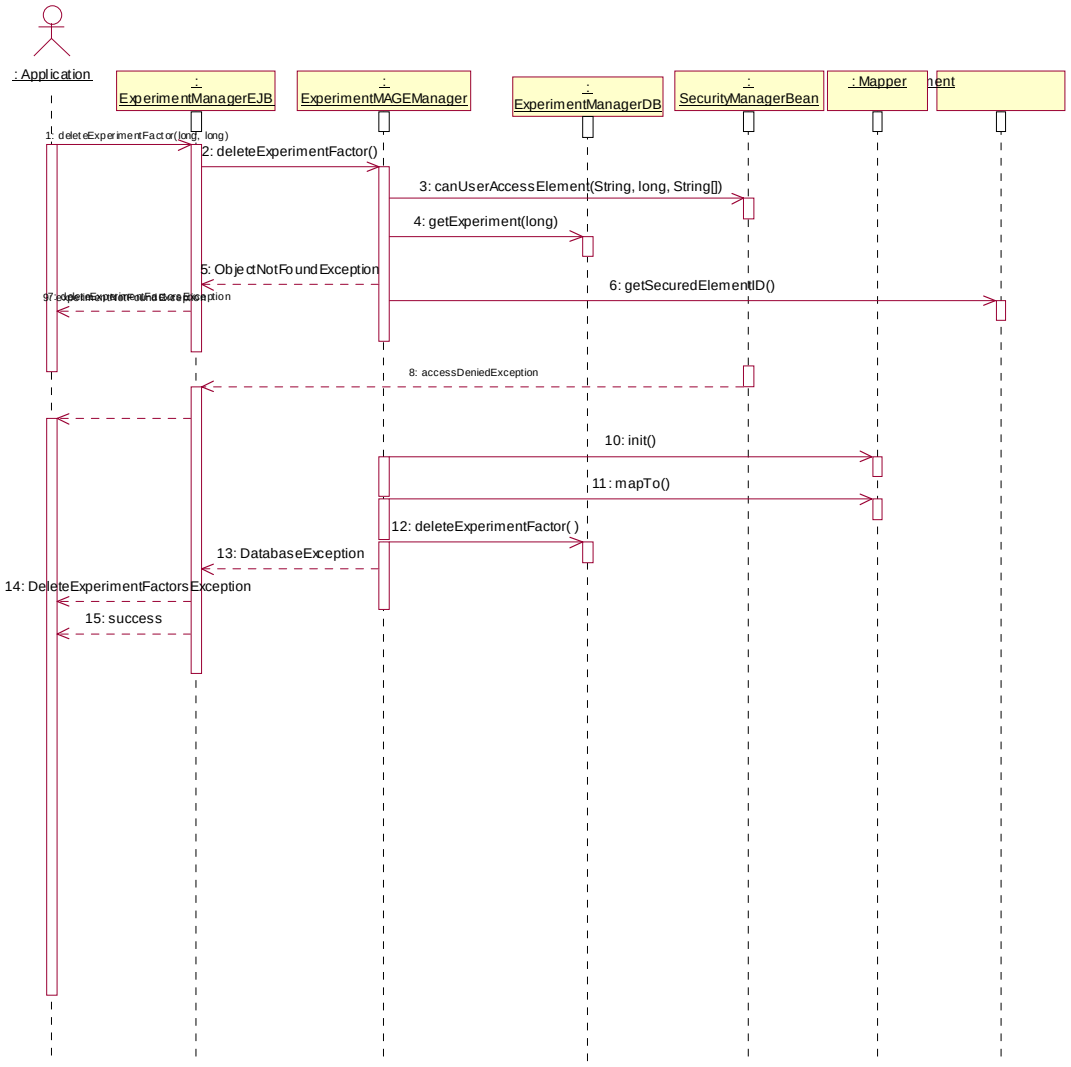
86. ExperimentManager.addExperimentFactors



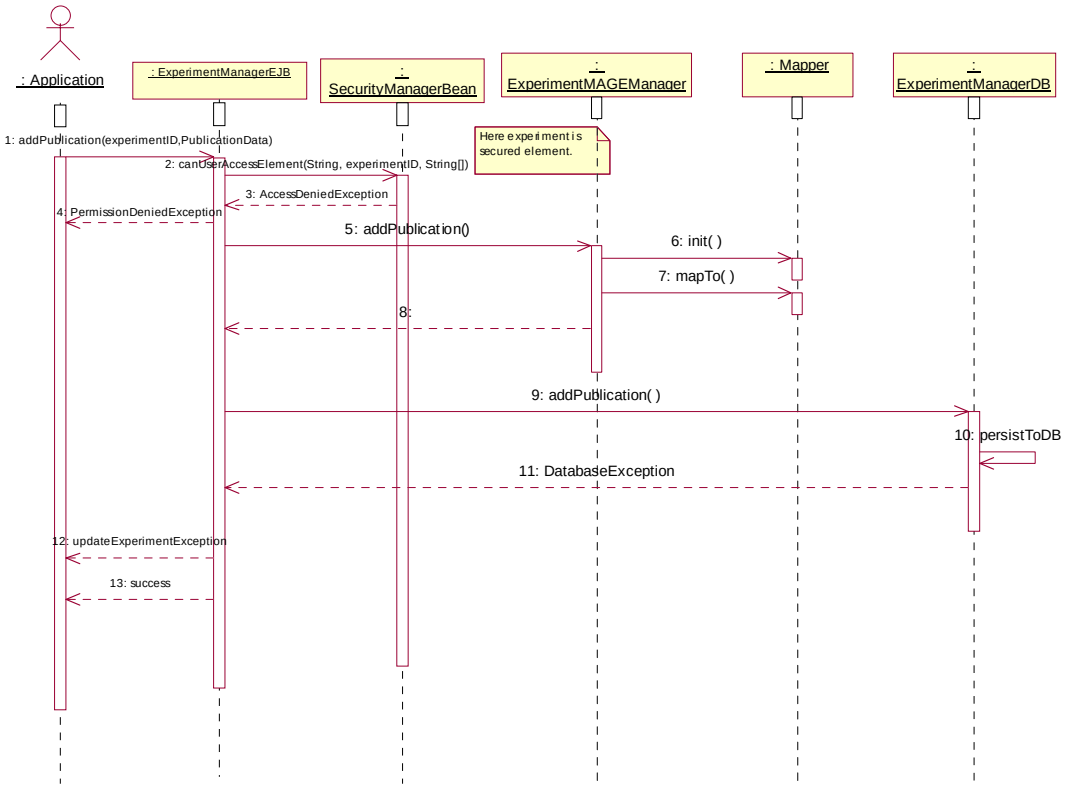
87. ExperimentManager.updateExperimentFactors



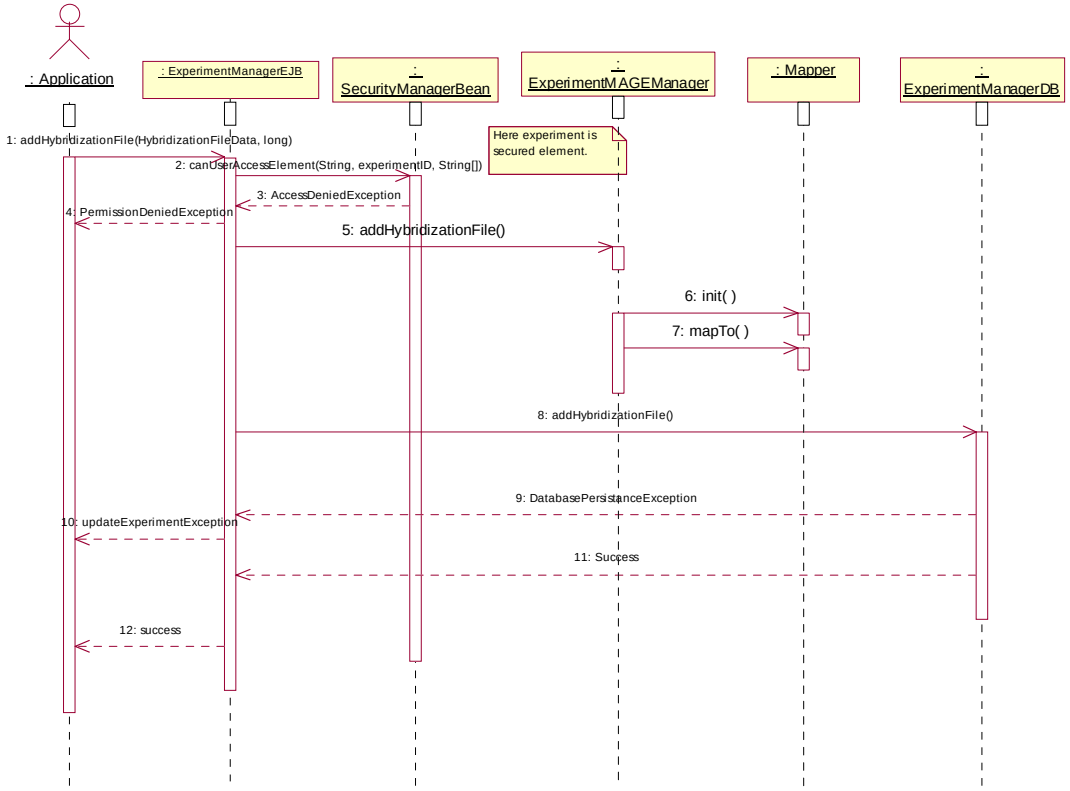
88. ExperimentManager.deleteExperimentFactors



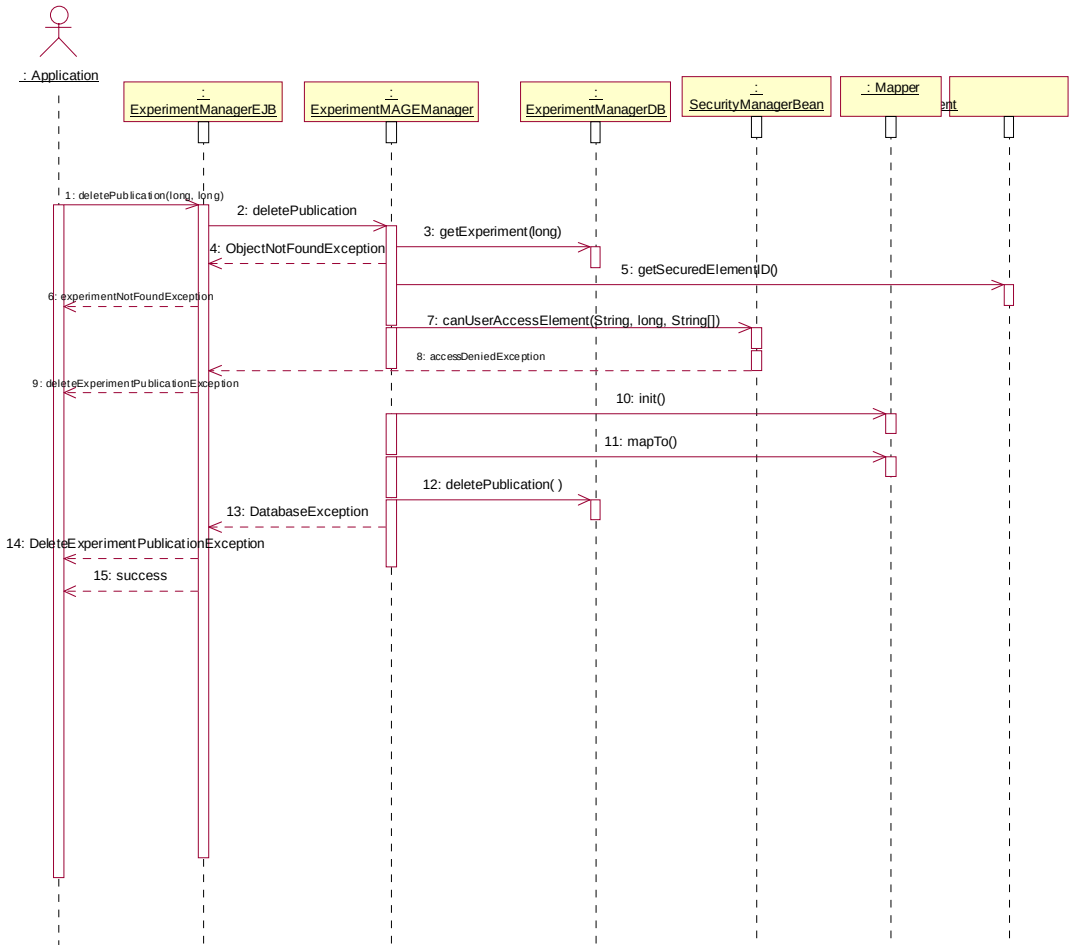
89. ExperimentManager.addPublication



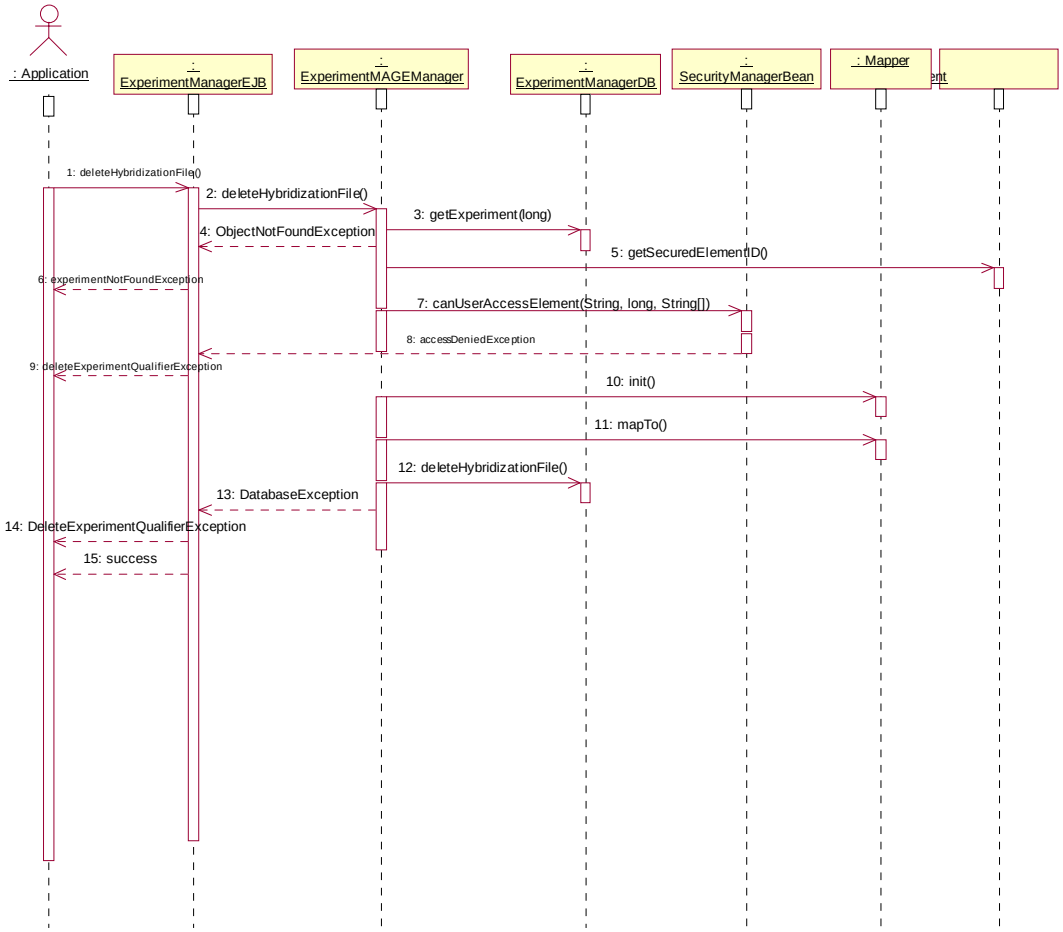
90. ExperimentManager.addHybridizationFile



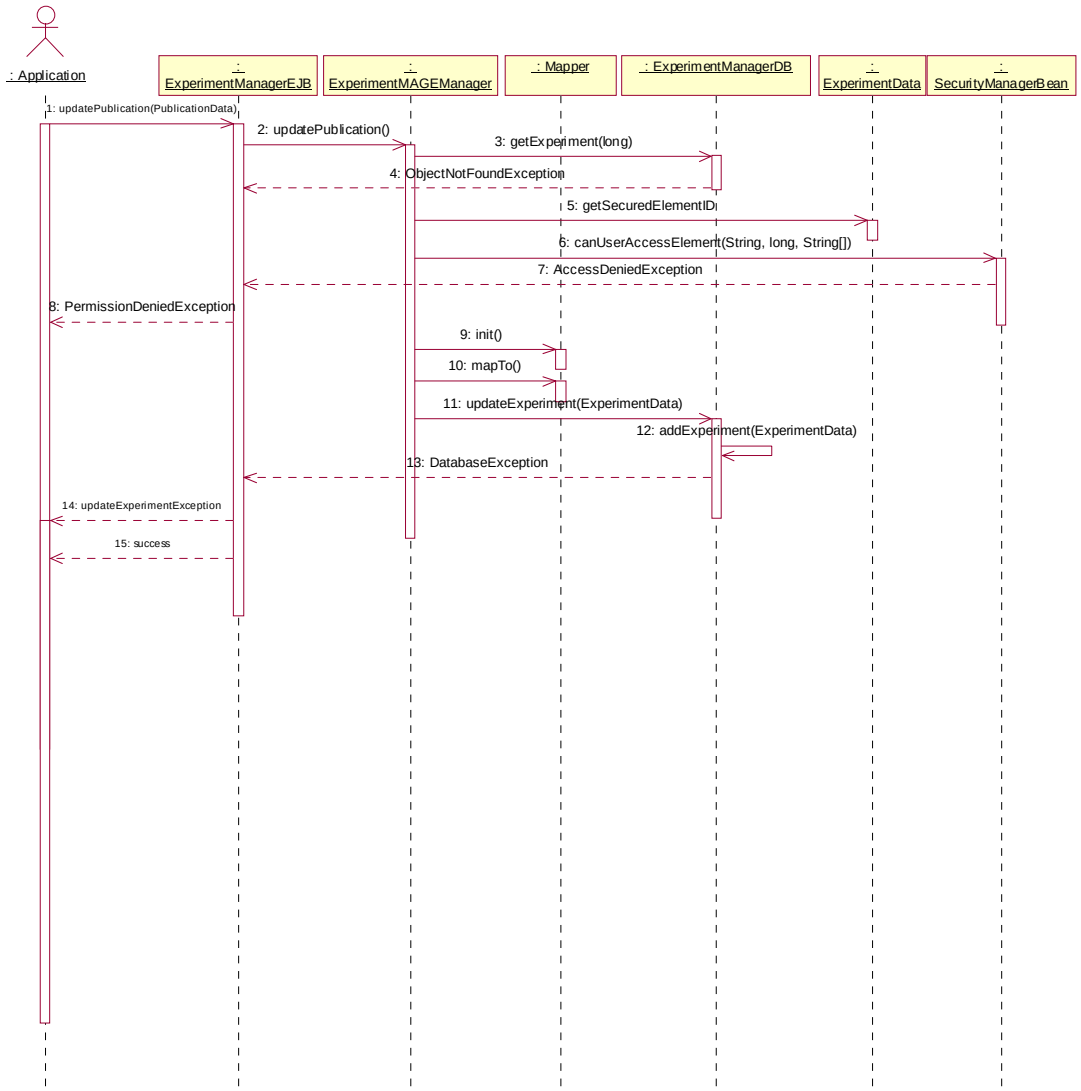
91. ExperimentManager.deletePublication



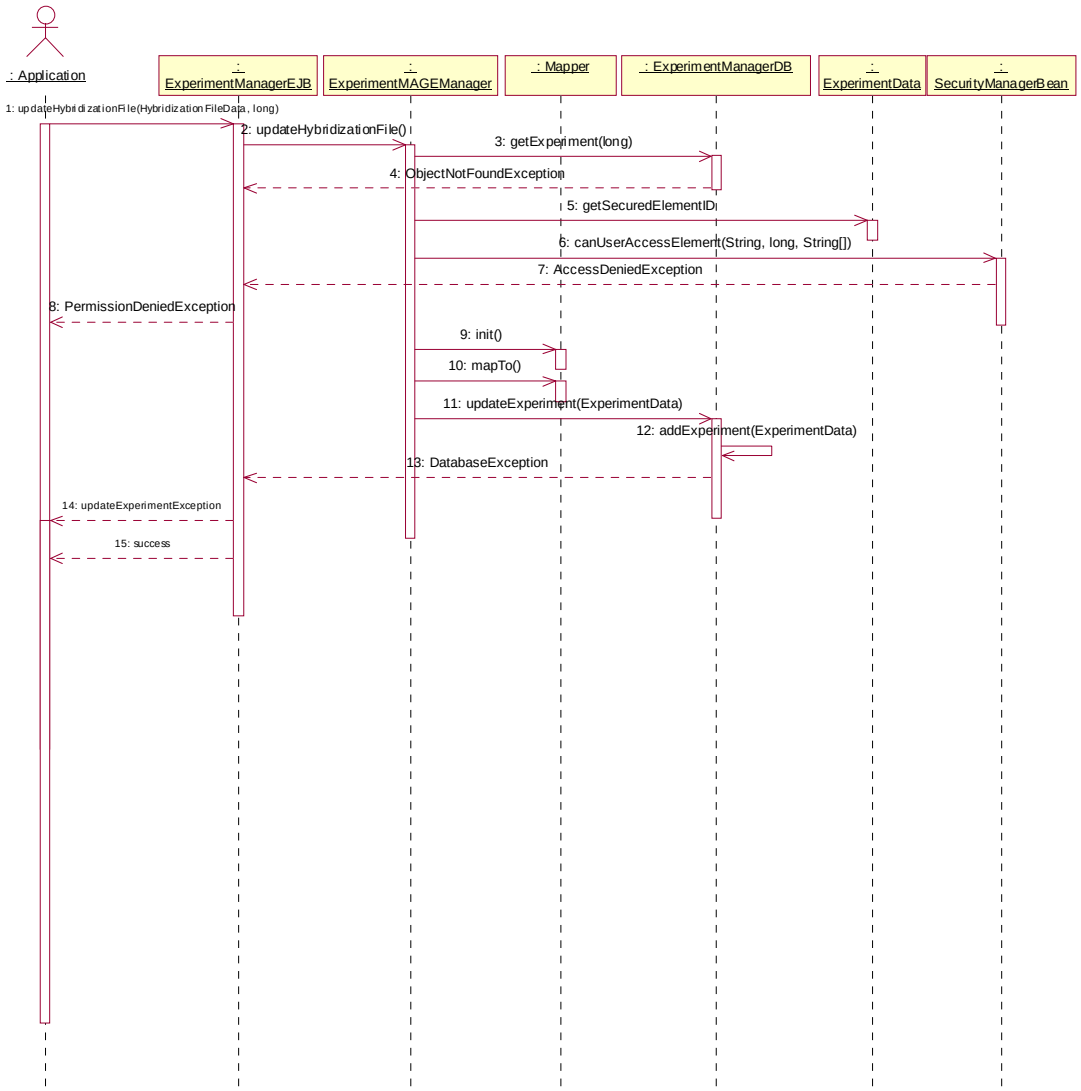
92. ExperimentManager.deleteHybridizationFile



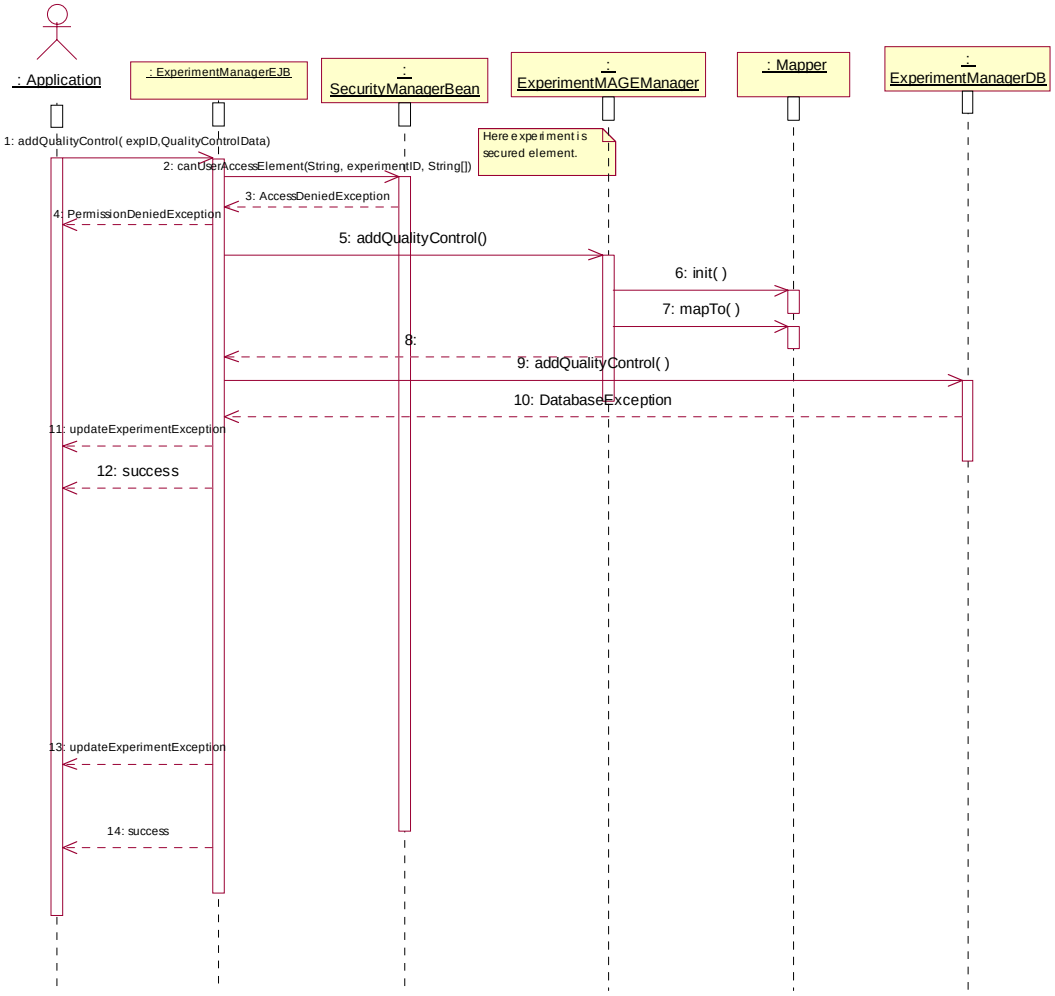
93. ExperimentManager.updatePublication



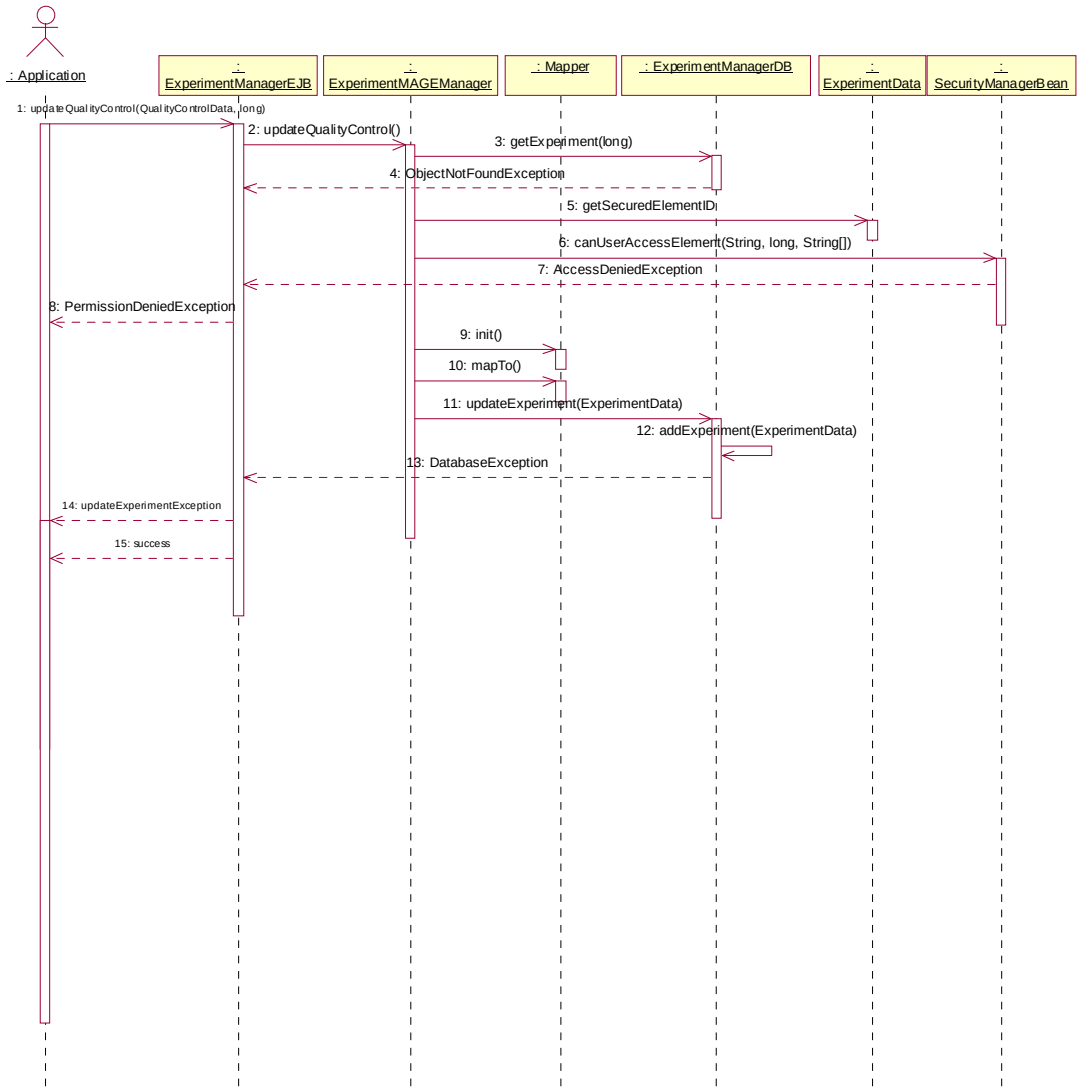
94. ExperimentManager.updateHybridizationFile



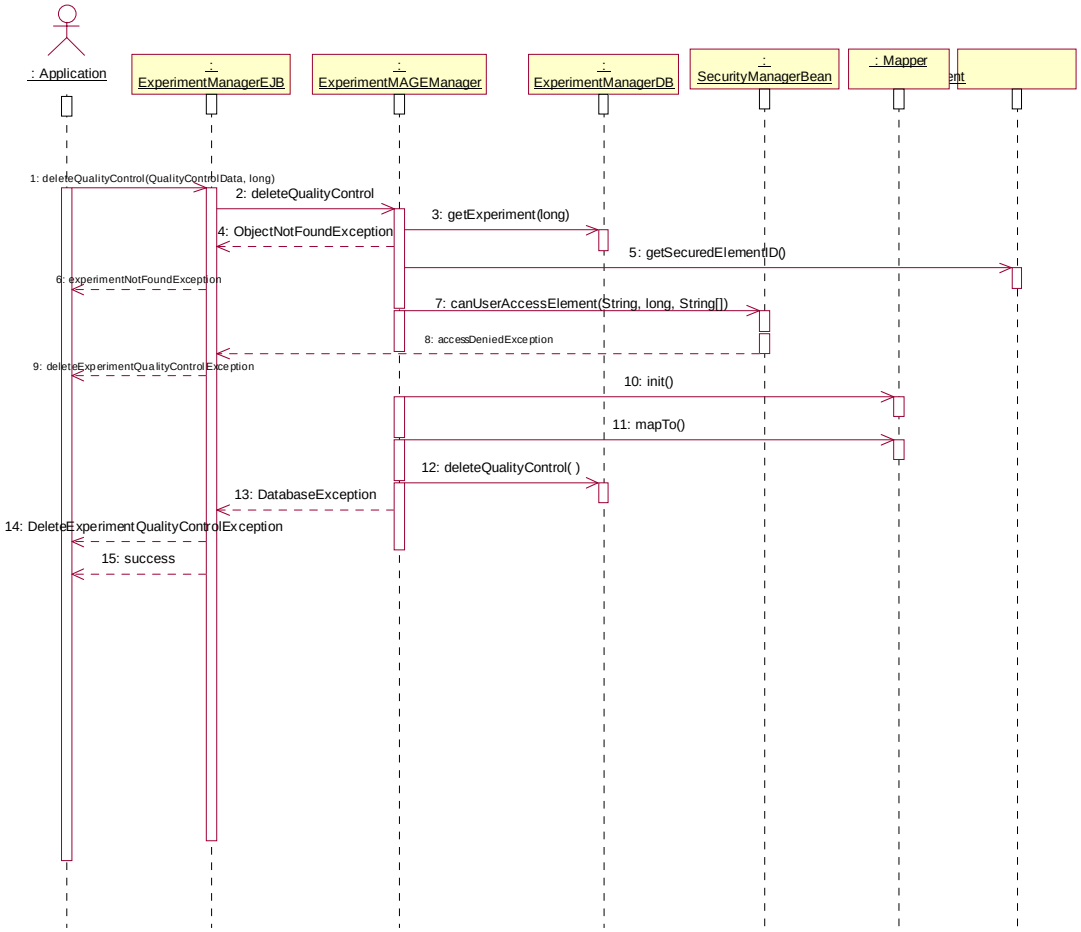
95. ExperimentManager.addQualityControl



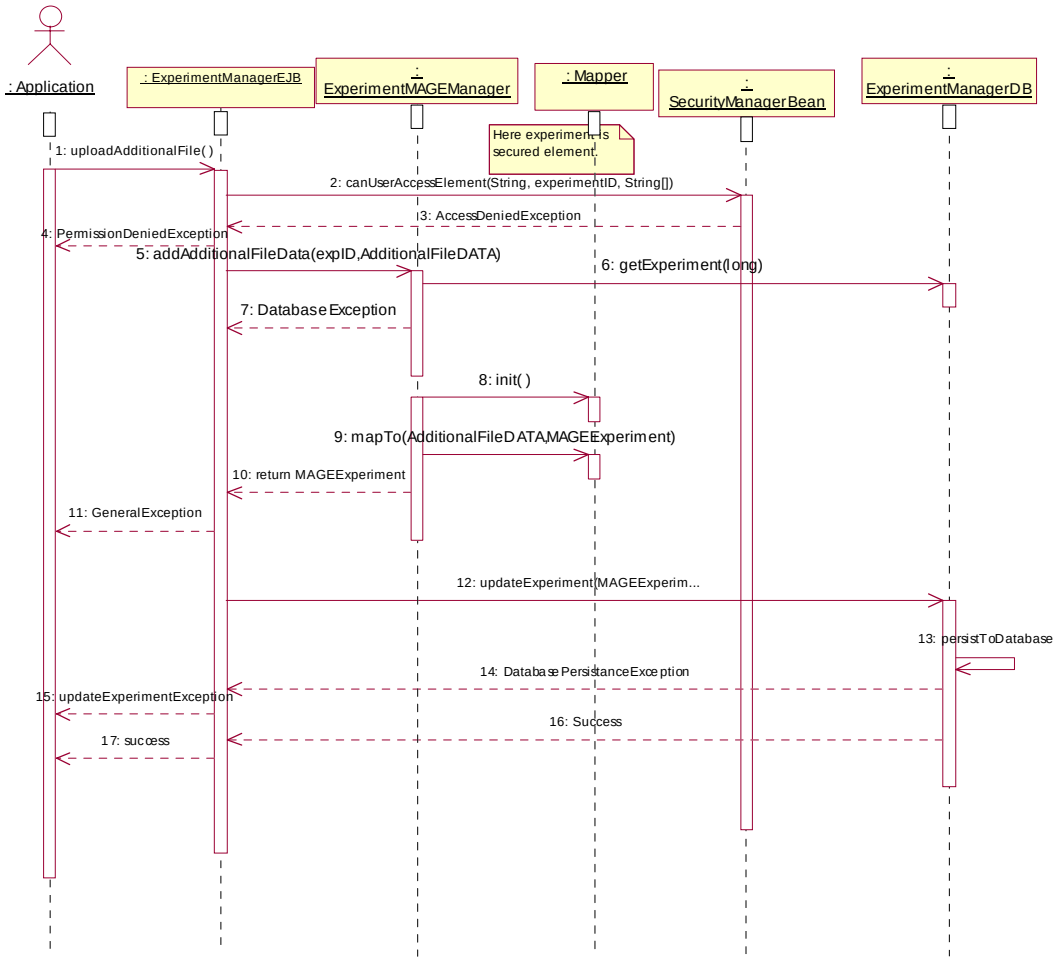
96. ExperimentManager.updateQualityControl



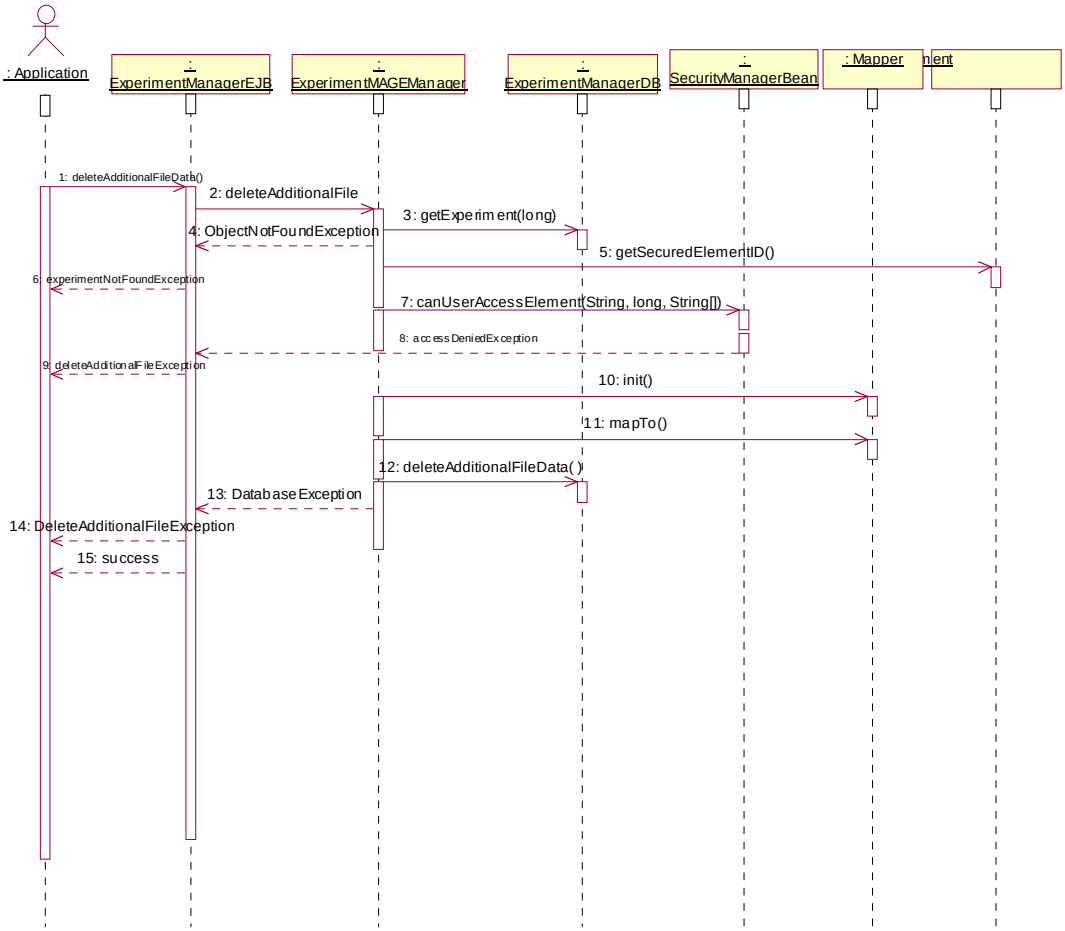
97. ExperimentManager.deleteQualityControl



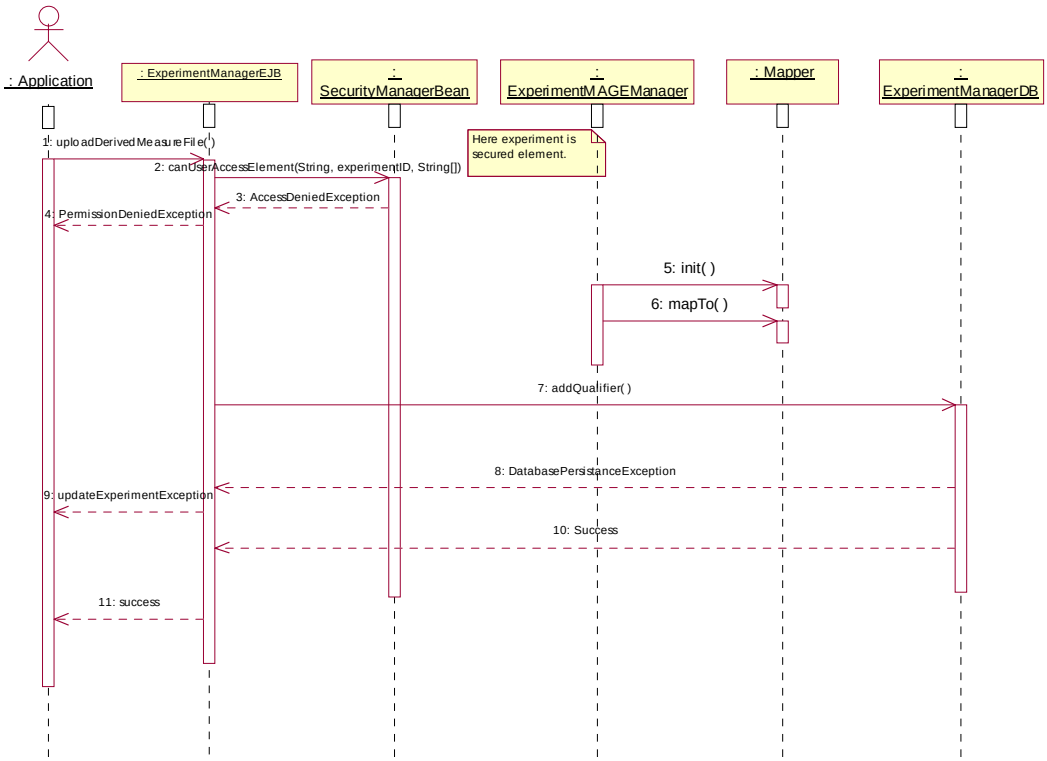
98. ExperimentManager.addAdditionalUploadedFile



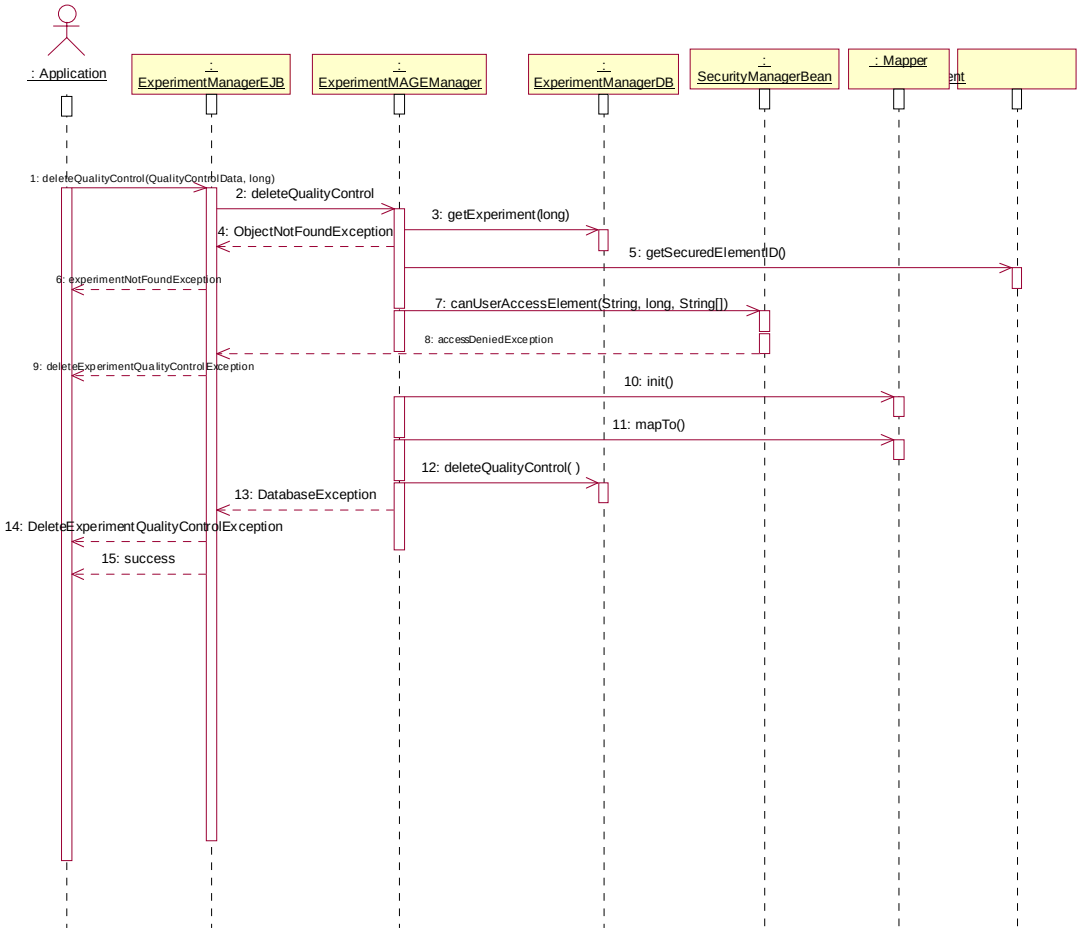
99. ExperimentManager.deleteAdditionalUploadedFile



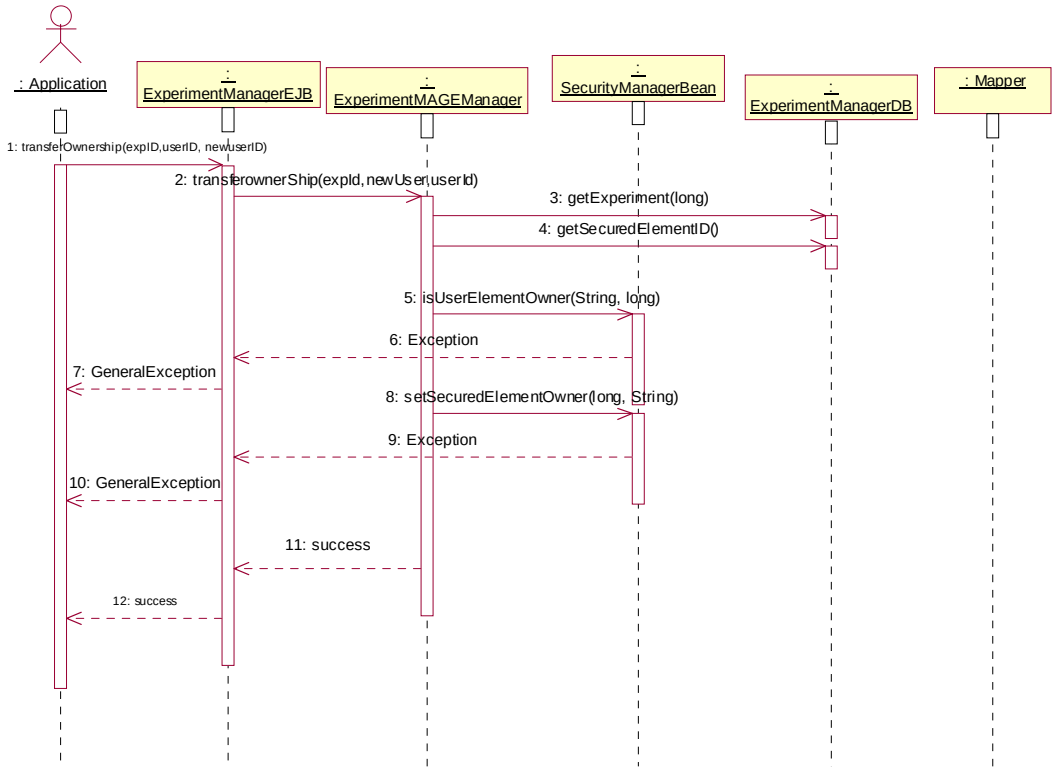
100. ExperimentManager.addDerivedMeasureFile



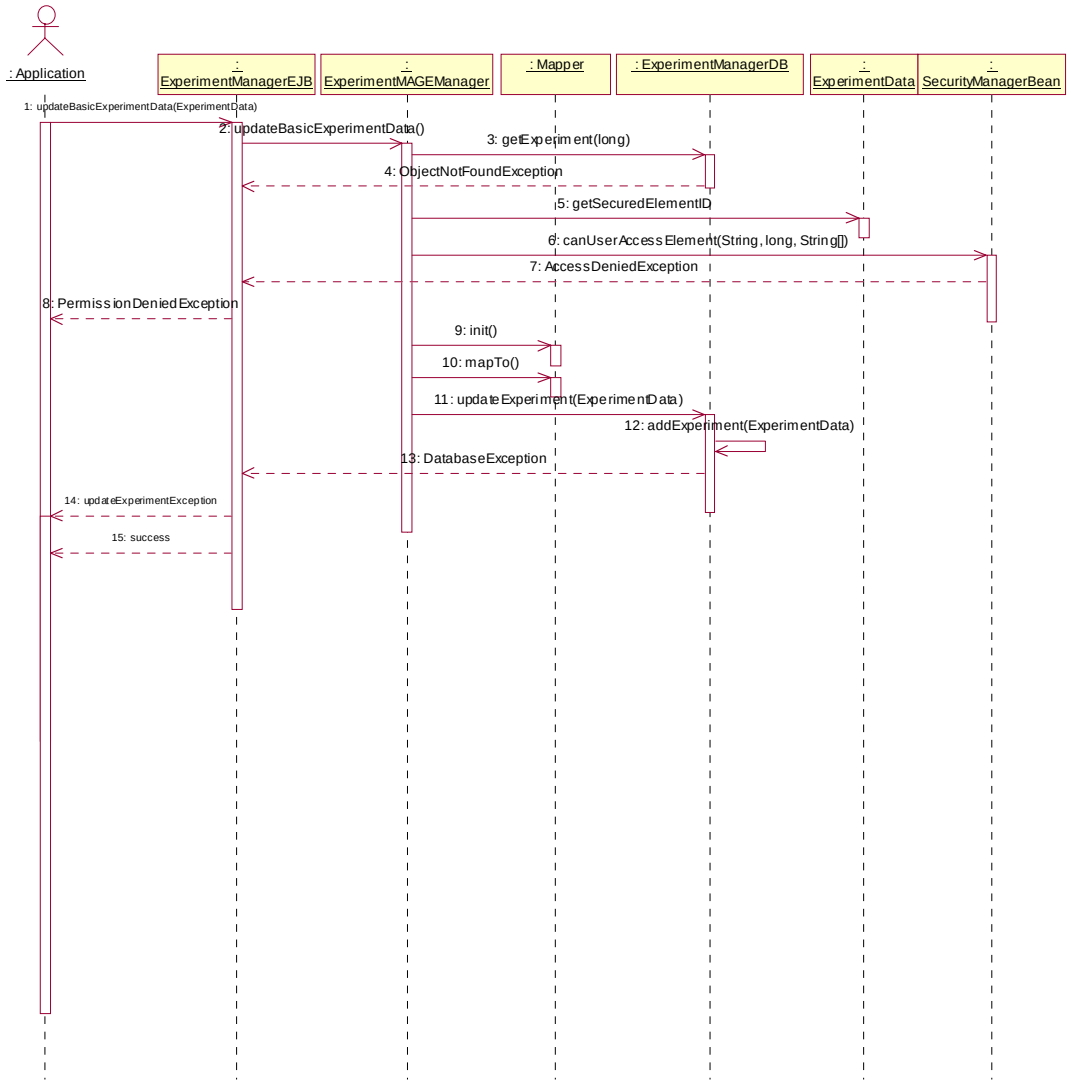
101. ExperimentManager.deleteDerivedMeasureFile



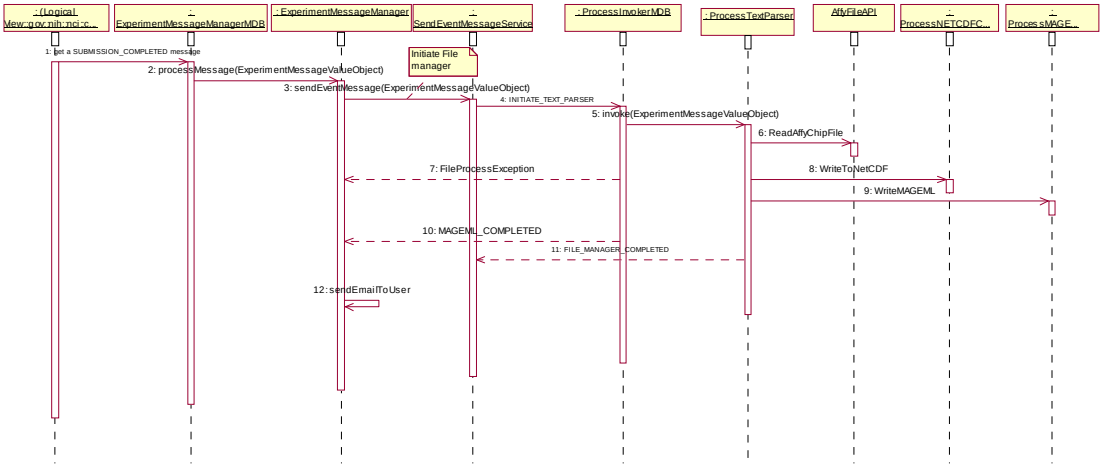
102. ExperimentManager.TransferExpOwnership



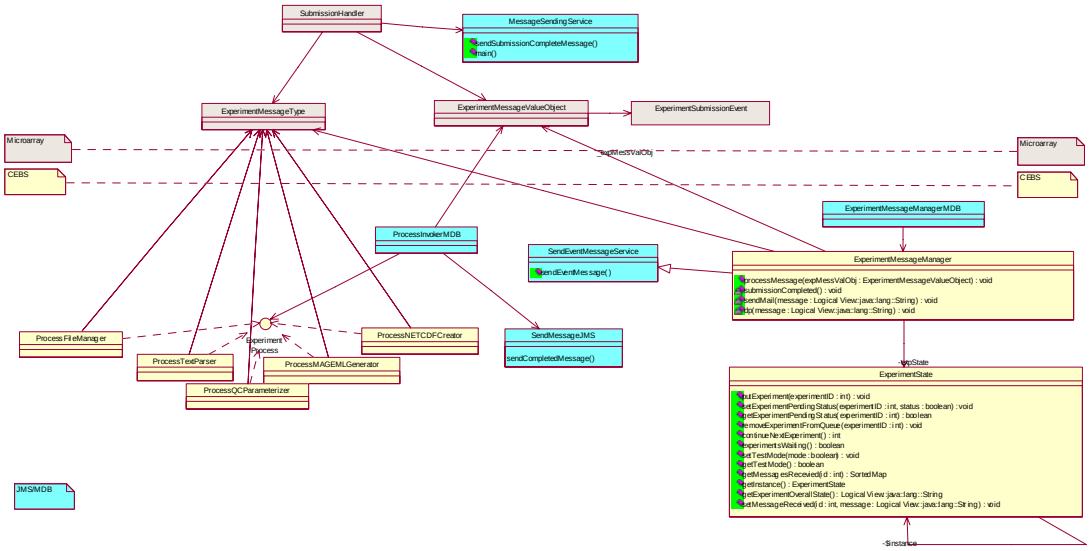
103. ExperimentManager.updateBasicExperimentData



104. ExperimentManager.processExperimentFile



105. Logical View::gov::nih::nci::caarray::services::FileParser



105.1 SendEventMessageService

105.2 ProcessFileManager@author Paul Boyer

105.3 ProcessMAGEMLGenerator@author Paul Boyer

105.4 ProcessNETCDFCreator@author Paul Boyer

105.5 ProcessQCParameterizer@author Paul Boyer

105.6 ProcessTextParser@author Paul Boyer

105.7 ExperimentStateimport gov.nih.nci.dc.bean.databeans.ExperimentSubmissionEvent;

105.8 import javax.naming.NamingException;

105.9 import java.util.Collections;

105.10 import javax.naming.InitialContext;

105.11 import javax.naming.Context;

105.12 @author Paul Boyer

105.13 ExperimentMessageManagerMDB

105.14 ExperimentMessageType

105.15 MessageSendingService@author Paul Boyer

105.16 Receives an event from SubmissionHandler that a Submission is Completed.

105.17 Then it sends a JMS message on Topic "topic/testTopic"

105.18 with ExperimentMessageType set to SUBMISSION_COMPLETED

105.19 ExperimentMessageManager@author Paul Boyer

105.20 ExperimentSubmissionEventpublic class ExperimentSubmissionEvent implements Serializable

105.21 Encapsulates the data required by objects persisting non-form input data pertaining to

105.22 a Microarray experiment (data files, XML files, etc...).

105.23 @author John Yost SAIC

105.24 @version 1.0

105.25 SubmissionHandlerimport java.io.*;

105.26 SubmissionHandler.java

105.27 Created on August 16, 2001, 7:09 AM

105.28 @author John Yost SAIC

105.29 @version 1.0

105.30 The SubmissionHandler class coordinates the persistence of form data via the FormDataBean object. Once the

105.31 form data has been persisted an ExperimentSubmissionEvent is fired and all registered listeners are notified.

105.32 @author John Yost SAIC

105.33 @version 1.0

105.34 Experiment Process@author Paul Boyer

105.35 ExperimentMessageValueObject

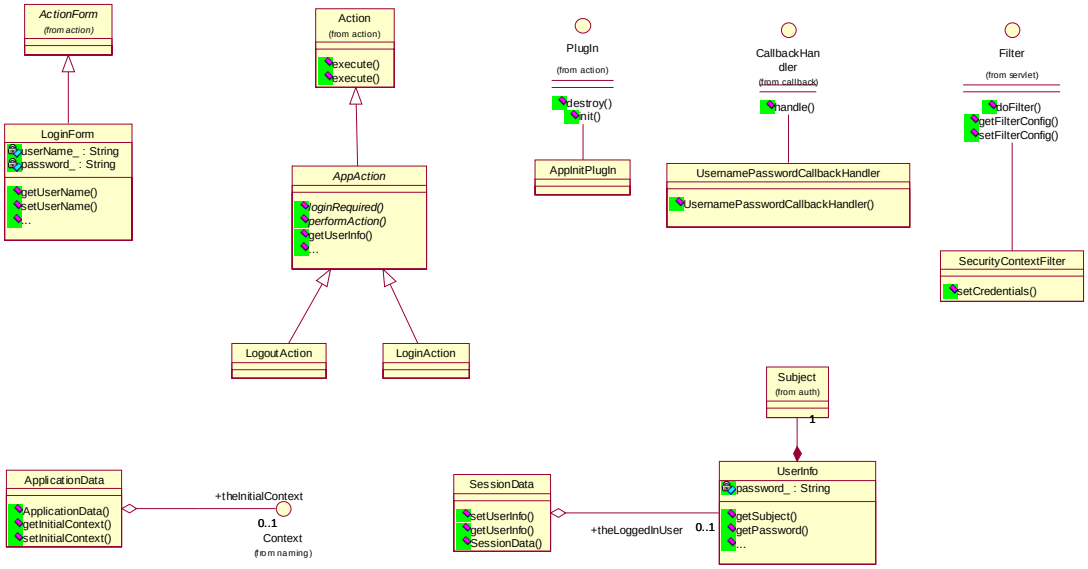
105.36 SendMessageJMS

105.37 ProcessInvokerMDB

105.38

106. Logical View::gov::nih::nci::caarray::ui

107. Logical View::gov::nih::nci::caarray::ui::core



This diagram shows core user interface classes. These classes provide a framework that is utilized by all other user interface components.

107.1 ApplicationDataThis class provides a container for all application scope data needed by user interface components.

107.2 SessionDataThis class provides a container for all session scope data needed by user interface components.

107.3 LoginActionThis action will perform the login operation.

107.4 LogoutActionThis class handles the user logouts.

107.5 UserInfoThis object is used to store information about the currently logged in user.

107.6 AppActionAbstract base class that all Action objects utilized in the caArray user interface will extend. This class performs core services such as verifying that a user is logged on when necessary. It also provides convenience methods for derived classes to utilize.

107.7 AppInitPlugInA PlugIn that is used to perform servlet initialization tasks.

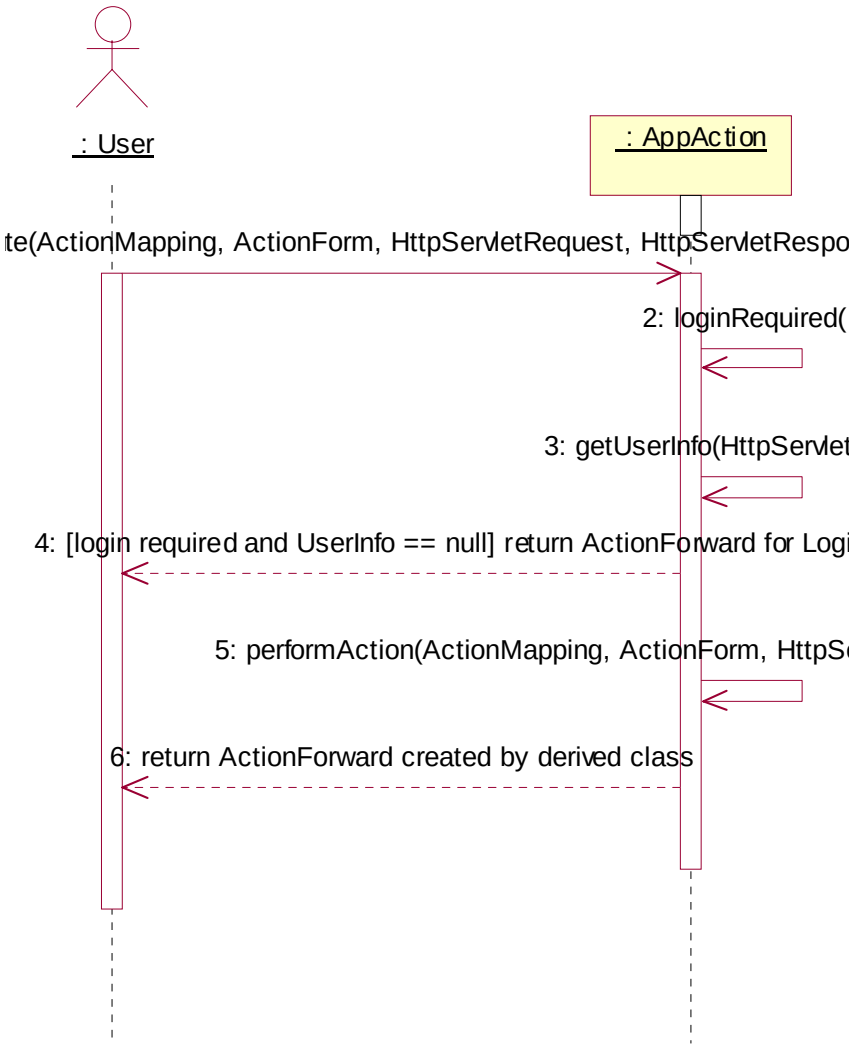
107.8 LoginFormThis class is used to store data from the login form. It is used in conjunction with the LoginAction class.

107.9 UsernamePasswordCallbackHandlerThis CallbackHandler is used to temporarily store the username and password needed for a JAAS login operation.

107.10 SecurityContextFilterThis class is a Filter that is used to set container specific security associations for each request.

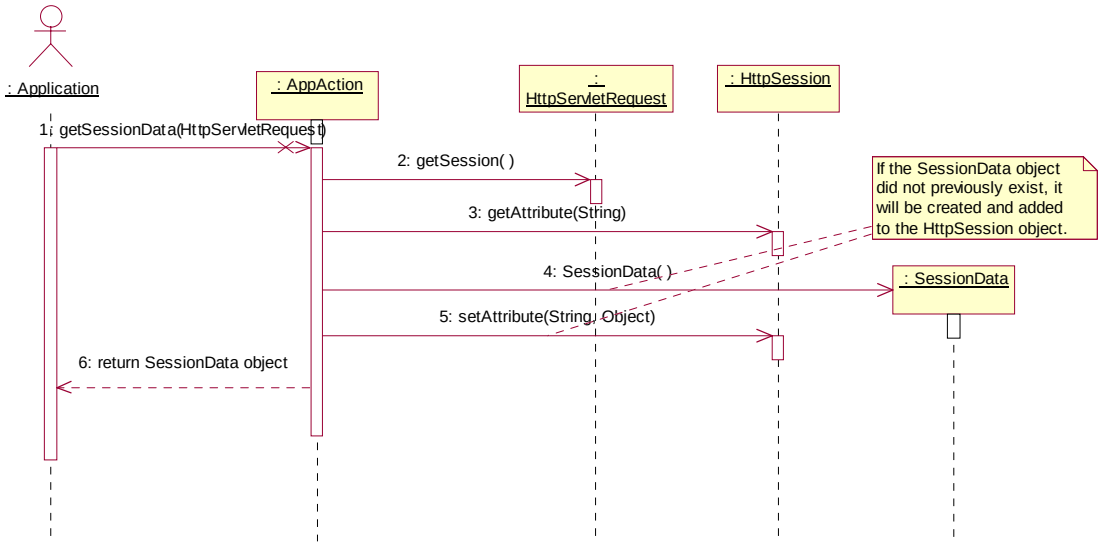
107.11

108. AppAction.execute



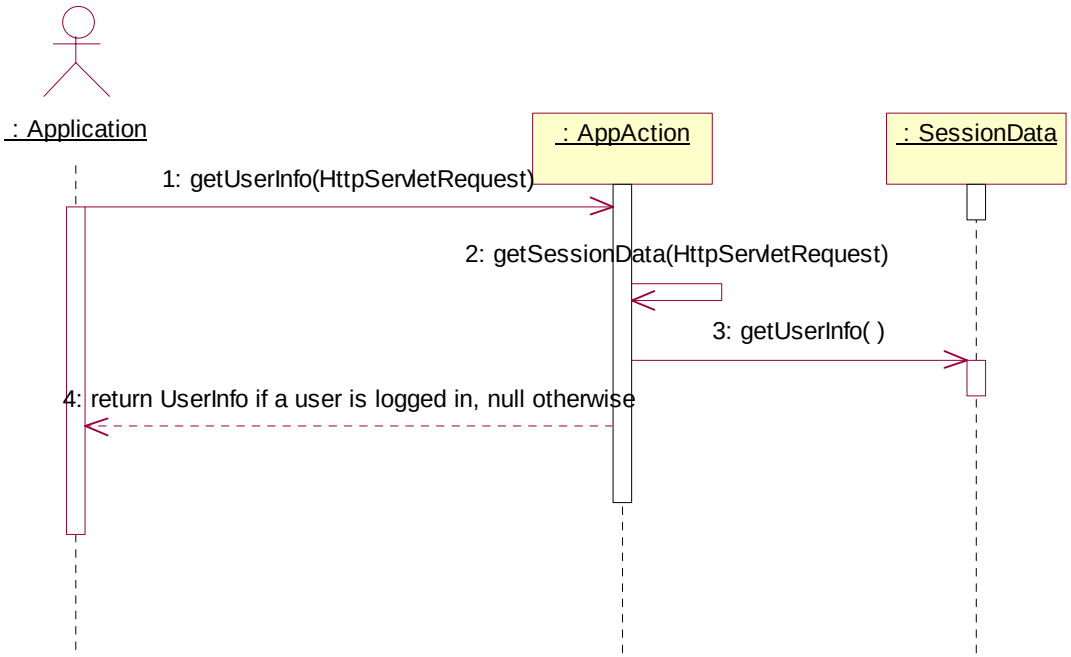
This diagram shows the processing peformed for every request. First a check is done to see if a login is required for this request. This method is abstract and must be implemented by the AppAction derived class to return true if login is required, or false if not. Next the information about the currently logged in user is obtained from the session. If a login is required and no user is currently logged in, an ActionForward object is returned that will cause the LoginAction to be performed. Otherwise the performAction() method of the derived AppAction will be called.

109. AppAction.getSessionData



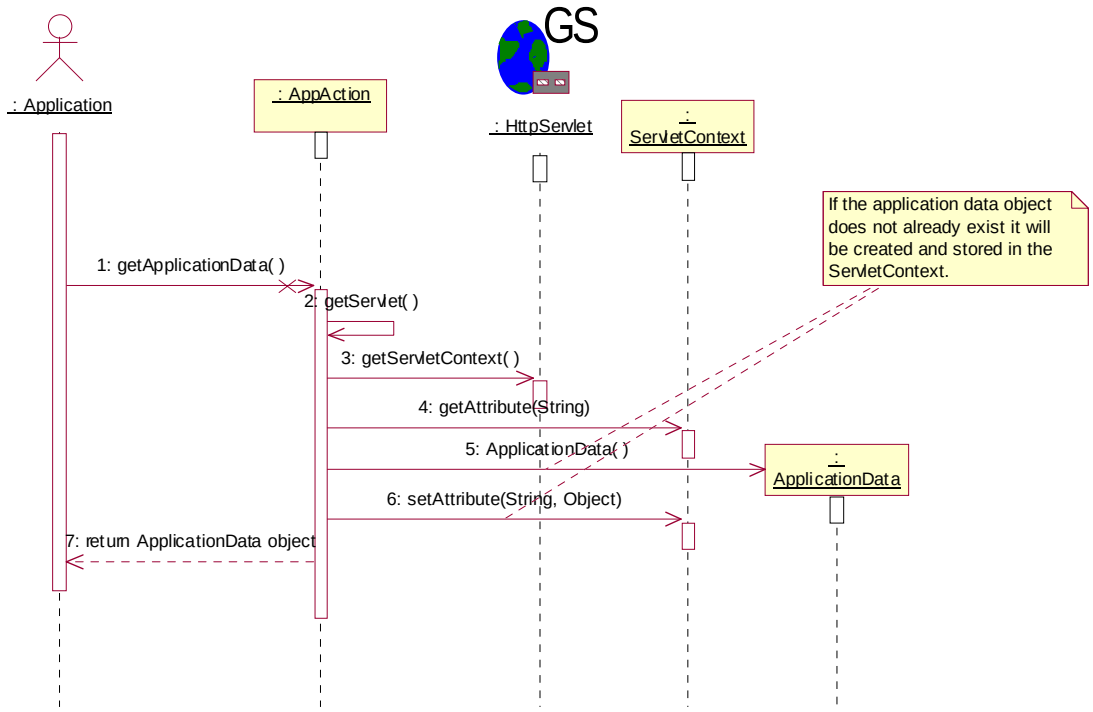
This method is provided to make it convenient for every AppAction derived handler to obtain the session scope data for the request being handled. The HttpSession object will be obtained from the specified request object. The "sessionData" attribute is then ontained from the HttpSession object. If the SessionData object does not already exist in the HttpSession object, one is created and the attribute is set.

110. AppAction.getUserInfo



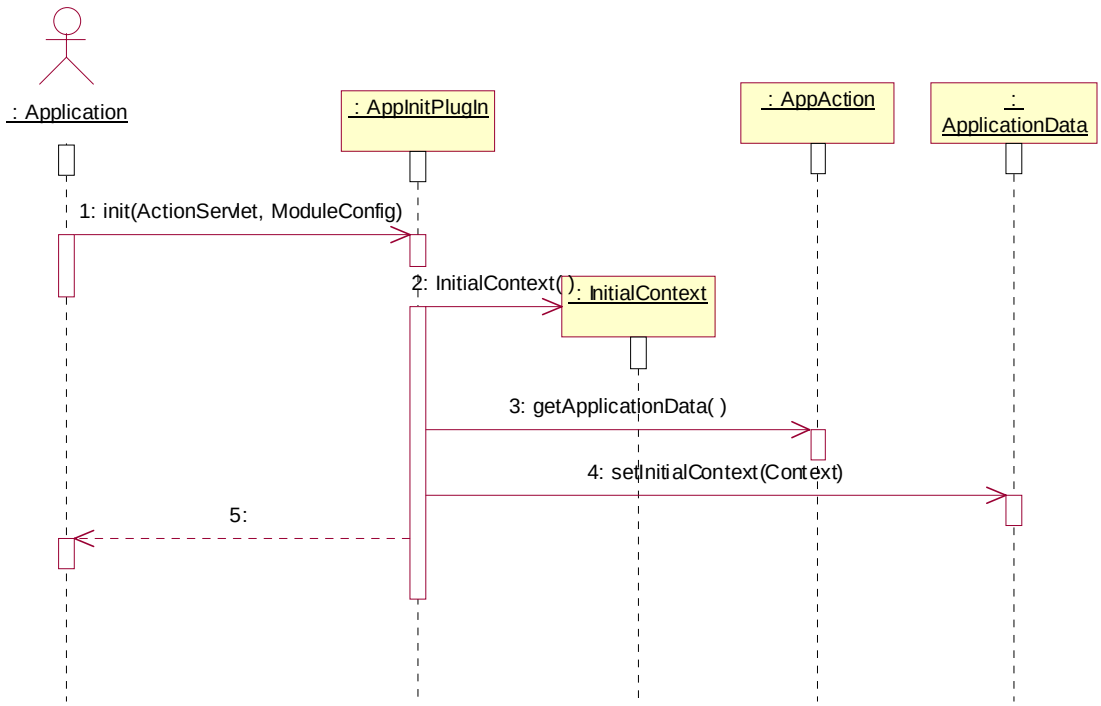
This diagram shows how the AppAction object obtains the UserInfo for the currently logged in user. It obtains the SessionData to get access to all session scope data stored by the application, then it calls getUserInfo() to get the information about the currently logged in user. This object is added to the SessionData by the LoginAction and is removed from the SessionData by the LogoutAction. If there is no user logged in this method will return null.

111. AppAction.getApplicationData



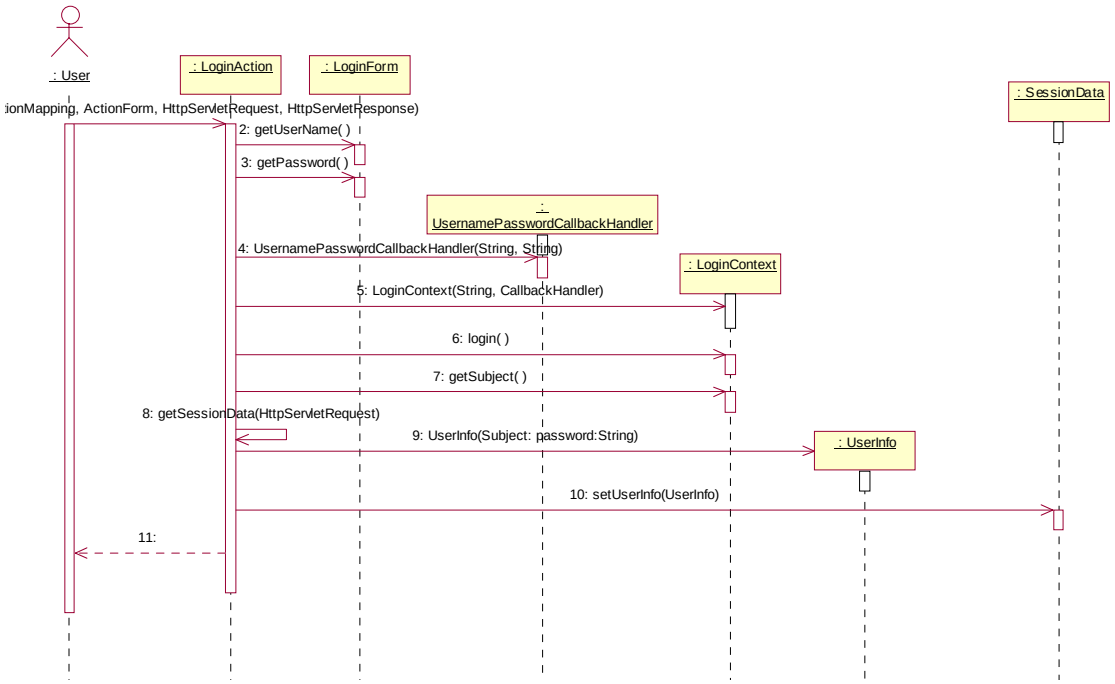
This method is provided to make it convenient for every AppAction derived handler to obtain the application scope data. The ApplicationData class is obtained from the servlet context. If one did not already exist, a new one is created.

112. AppInitPlugin.init



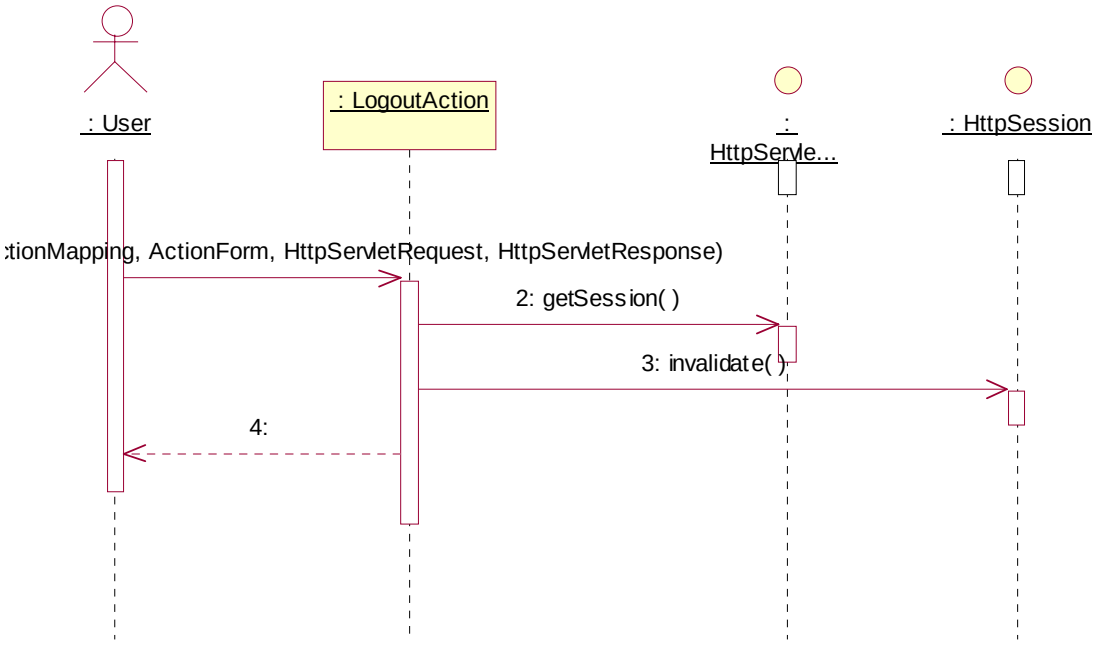
This diagram shows how the InitialContext used to locate EJBs is set into the ApplicationData object at servlet initialization.

113. LoginAction.performAction



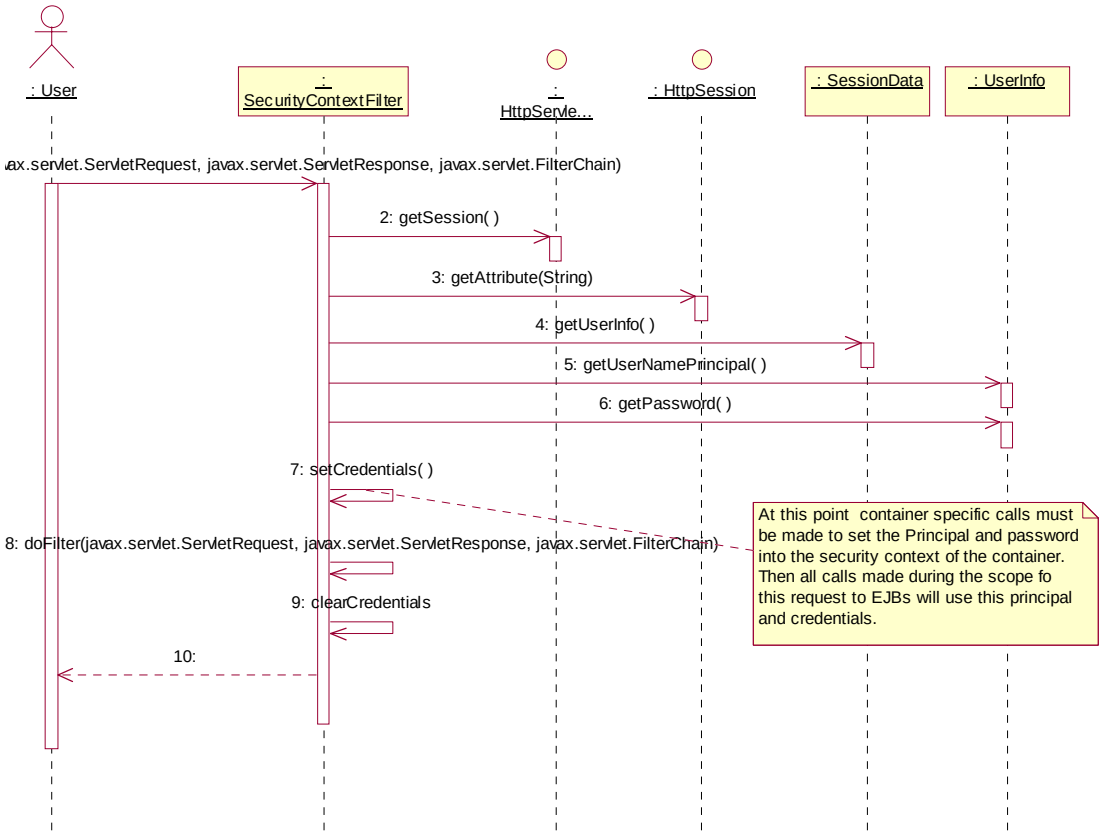
This diagram shows the processing performed when a user attempts to login. Ther username and password entered are obtained from the login form. Then a callback handler is created. This object is needed in order to perform the JAAS login on the LoginContext object. If the login succeeds, the authenticated Subject is obtained from the LoginContext and is set into the SessionData object for this user.

114. LogoutAction.performAction



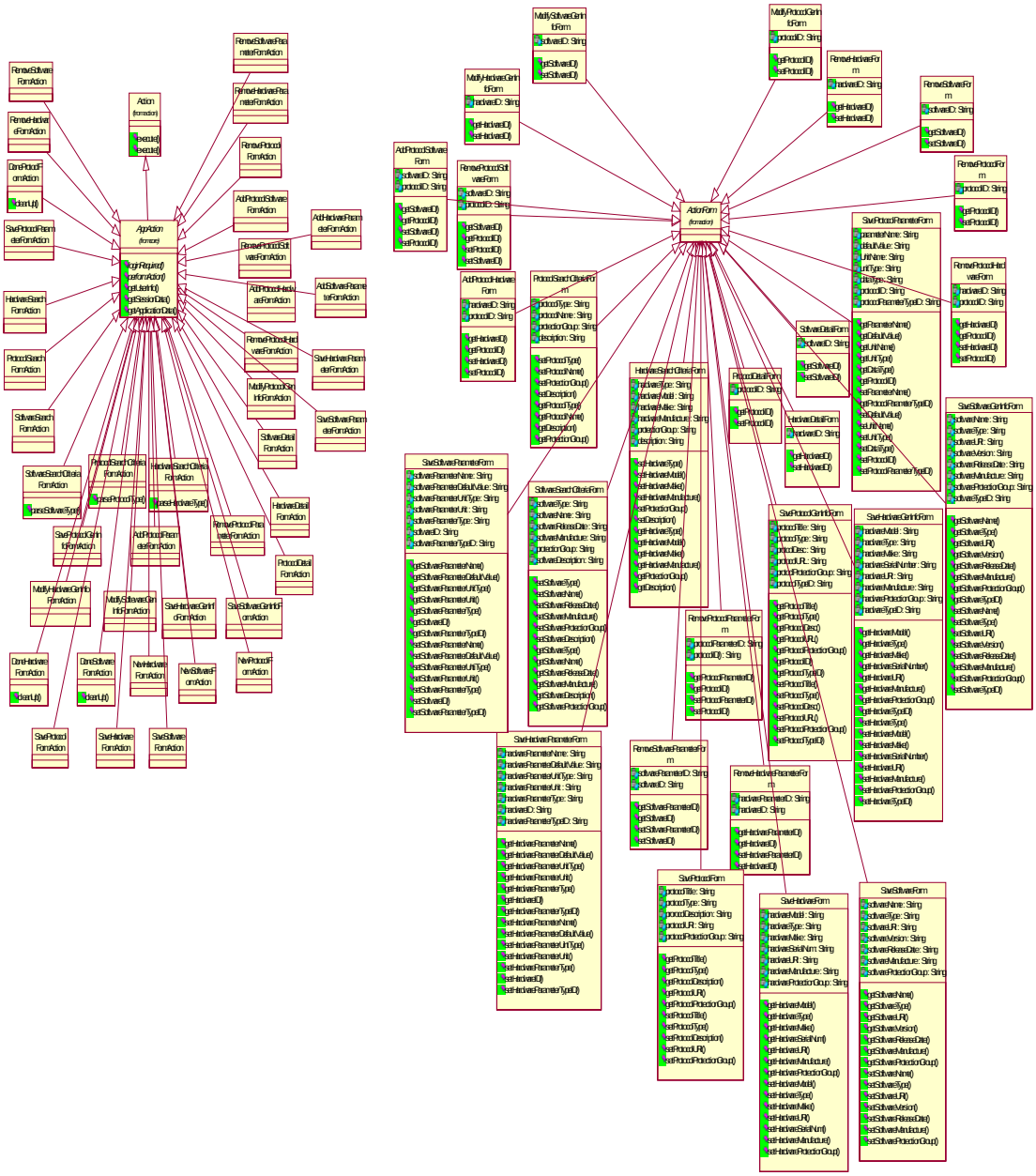
This diagram shows the processing peformed when a user logs out of the application. The Session object for this user is obtained from the incoming HTTPServletReqeuest and is invalidated. This results in the application no longer retaining the session data for this user, including the UserInfo object. Thus the application will no longer think there is a user logged in for this session.

115. SecurityContextFilter.doFilter



This diagram shows how a servlet filter is used to set the security principal and credentials needed for EJB calls. The setting lasts for the duration of the request and then is cleared. This code is EJB container specific.

116. Logical View::gov::nih::nci::caarray::ui::protocol



116.1 ProtocolSearchCriteriaFormActionThis class is to gather the search protocol information and create a protocol search criteria. After calling ProtocolManager.search, the result which contains ProtocolDesc[] will be returned

116.2 ProtocolSearchFormActionThis class is to get protocl Type items and visibility items in order to create Protocol Search Page

116.3 ProtocolSearchCriteriaForm

116.4 HardwareSearchFormActionThis class is to get hardwareType, protection group and hardware Manufacture in order to create Hardware Search Page

116.5 HardwareSearchCriteriaFormActionThis class is to create Hardware search criteria and call hardwareManager.search to get matched hardware types back, and use this result to create hardware search JSP page.

116.6 HardwareSearchCriteriaForm

116.7 SoftwareSearchFormAction

116.8 SoftwareSearchCriteriaFormAction

116.9 SoftwareSearchCriteriaForm

116.10 ProtocolDetailFormAction

116.11 ProtocolDetailForm

116.12 HardwareDetailFormAction

116.13 HardwareDetailForm

116.14 SoftwareDetailFormAction

116.15 SoftwareDetailForm

116.16 ModifyProtocolGenInfoFormAction

116.17 SaveProtocolGenInfoFormAction

116.18 SaveProtocolGenInfoForm

116.19 SaveProtocolParameterForm

116.20 AddProtocolParameterFormAction

116.21 SaveProtocolParameterFormAction

116.22 RemoveProtocolParameterFormAction

116.23 RemoveProtocolParameterForm

116.24 RemoveProtocolHardwareFormAction

116.25 RemoveProtocolHardwareForm

116.26 AddProtocolHardwareFormAction

116.27 AddProtocolHardwareForm

116.28 RemoveProtocolSoftwareFormAction

116.29 RemoveProtocolSoftwareForm

116.30 AddProtocolSoftwareFormAction

116.31 AddProtocolSoftwareForm

116.32 RemoveProtocolFormAction

116.33 RemoveProtocolForm

116.34 DoneProtoclFormAction

116.35 ModifyHardwareGenInfoFormAction

116.36 ModifySoftewareGenInfoFormAction

116.37 SaveHardwareGenInfoFormAction

116.38 SaveSoftwareGenInfoFormAction

116.39 SaveHardwareGenInfoForm

116.40 SaveSoftwareGenInfoForm

116.41 RemoveHardwareParameterFormAction

116.42 RemoveSoftwareParameterFormAction

116.43 RemoveHardwareParameterForm

116.44 RemoveSoftwareParameterForm

116.45 AddHardwareParameterFormAction

116.46 AddSoftwareParameterFormAction

116.47 SaveHardwareParameterFormAction

116.48 SaveSoftwareParameterFormAction

116.49 SaveHardwareParameterForm

116.50 SaveSoftwareParameterForm

116.51 RemoveHardwareFormAction

116.52 RemoveSoftwareFormAction

116.53 RemoveHardwareForm

116.54 RemoveSoftwareForm

116.55 DoneHardwareFormAction

116.56 DoneSoftwareFormAction

116.57 NewHardwareFormAction

116.58 NewSoftwareFormAction

116.59 NewProtocolFormAction

116.60 SaveProtocolFormAction

116.61 SaveProtocolForm

116.62 SaveHardwareFormAction

116.63 SaveSoftwareFormAction

116.64 SaveHardwareForm

116.65 SaveSoftwareForm

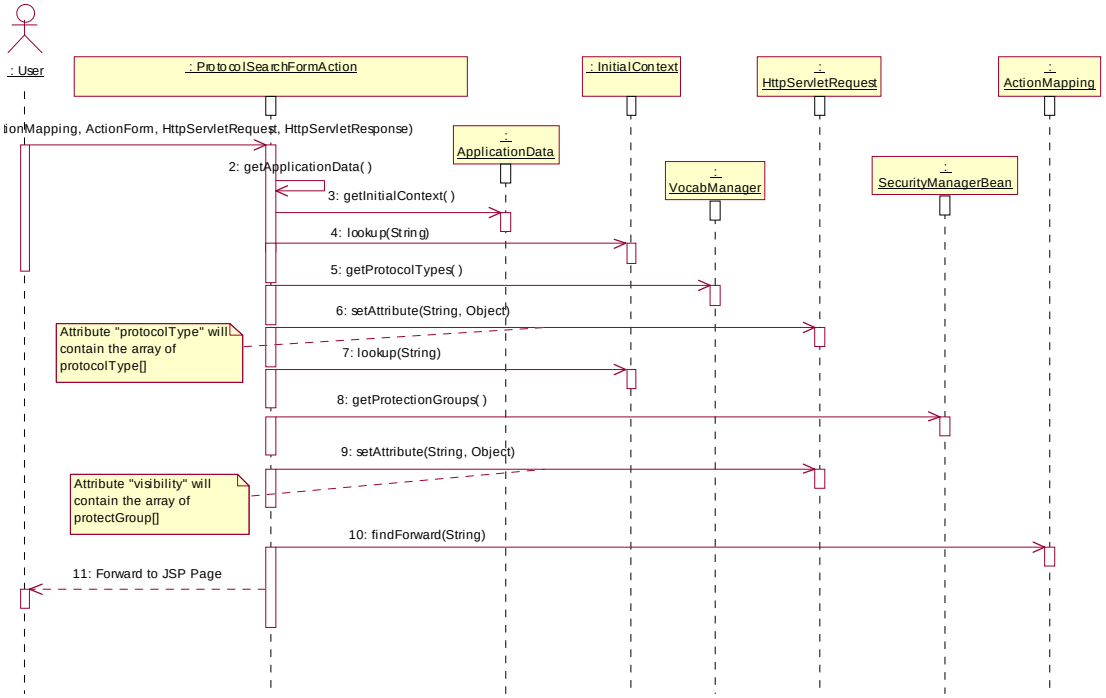
116.66 ModifyHardwareGenInfoForm

116.67 ModifyProtocolGenInfoForm

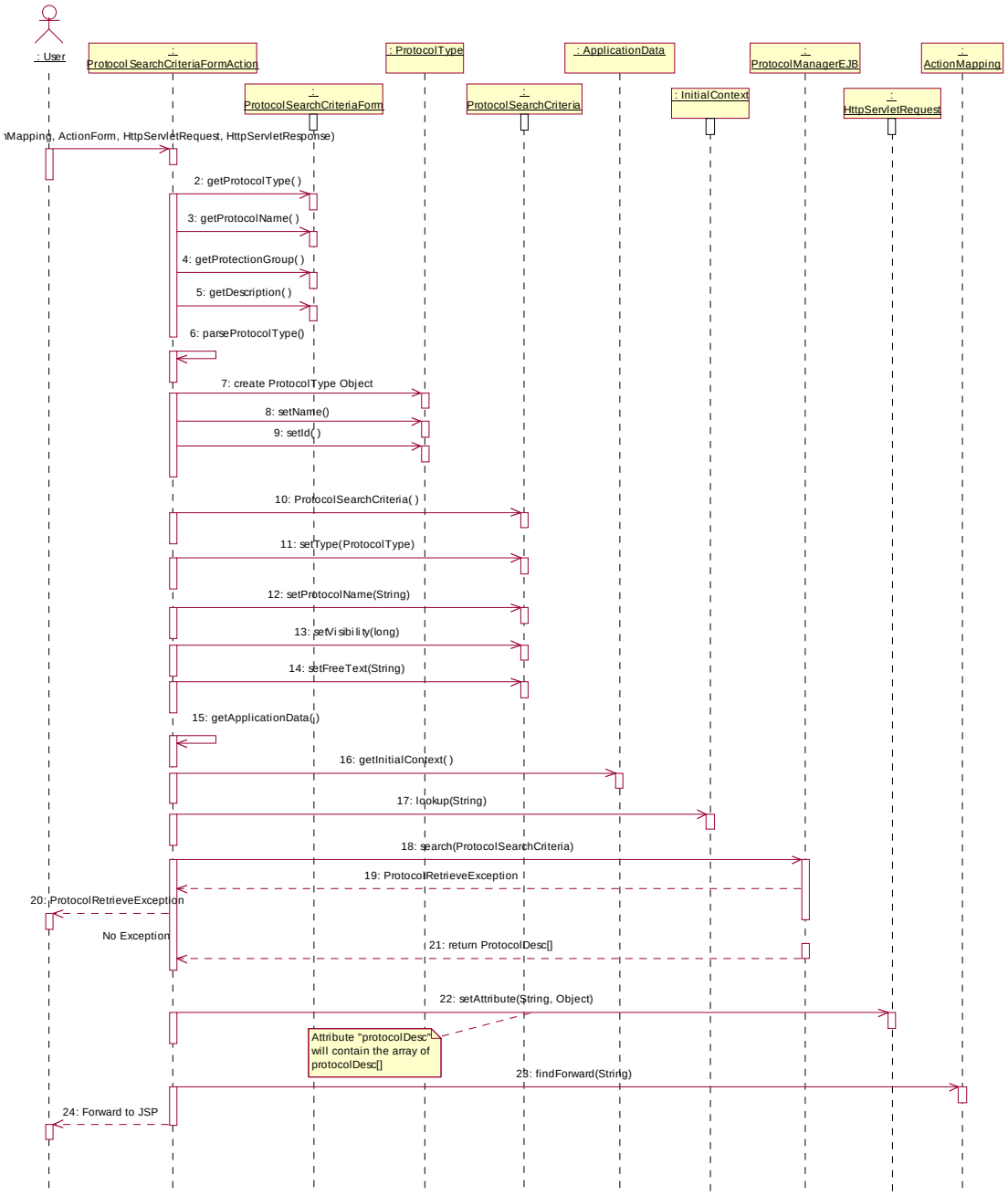
116.68 ModifySoftwareGenInfoForm

116.69

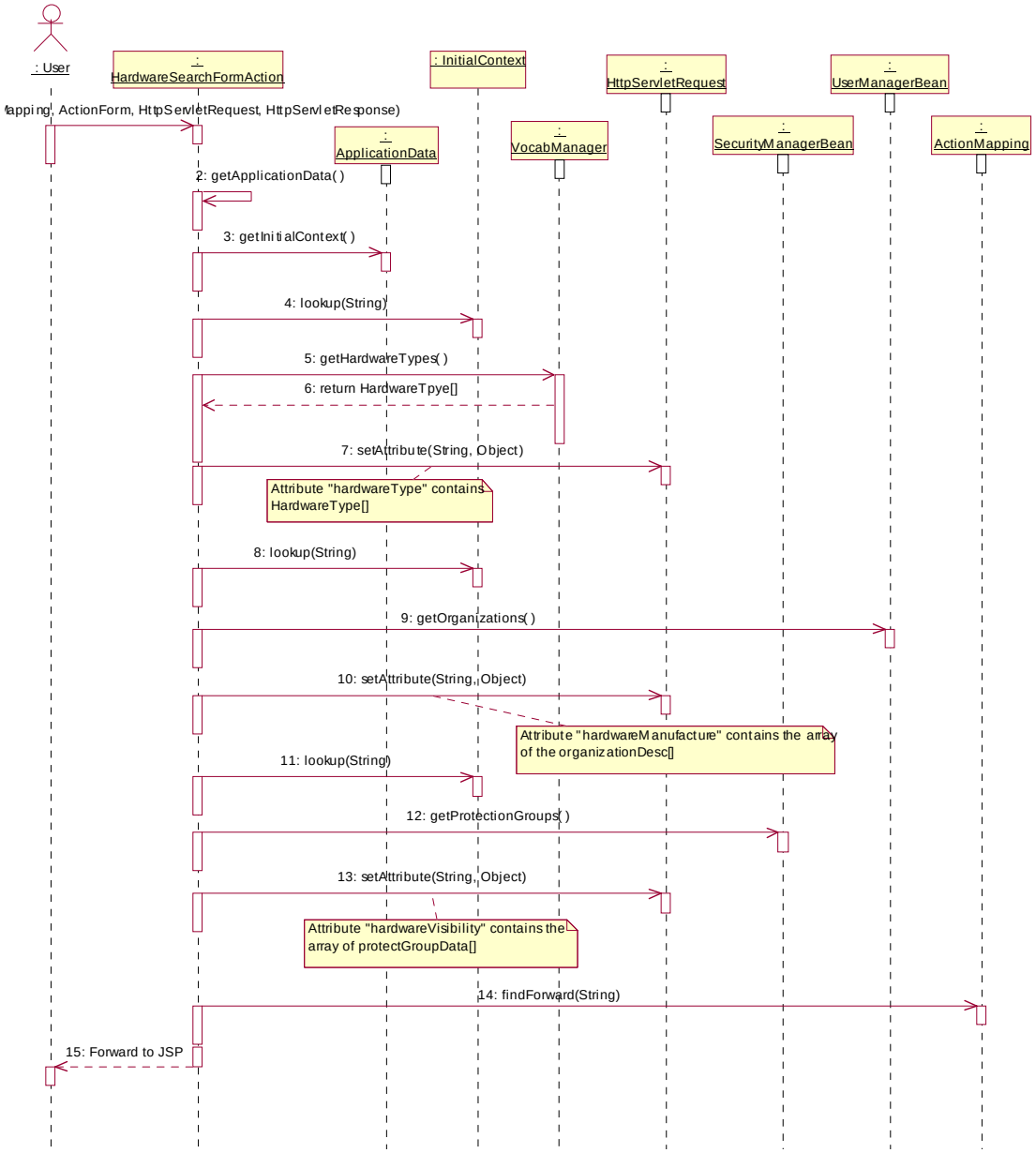
117. BuildProtocolSearchFormAction.performAction



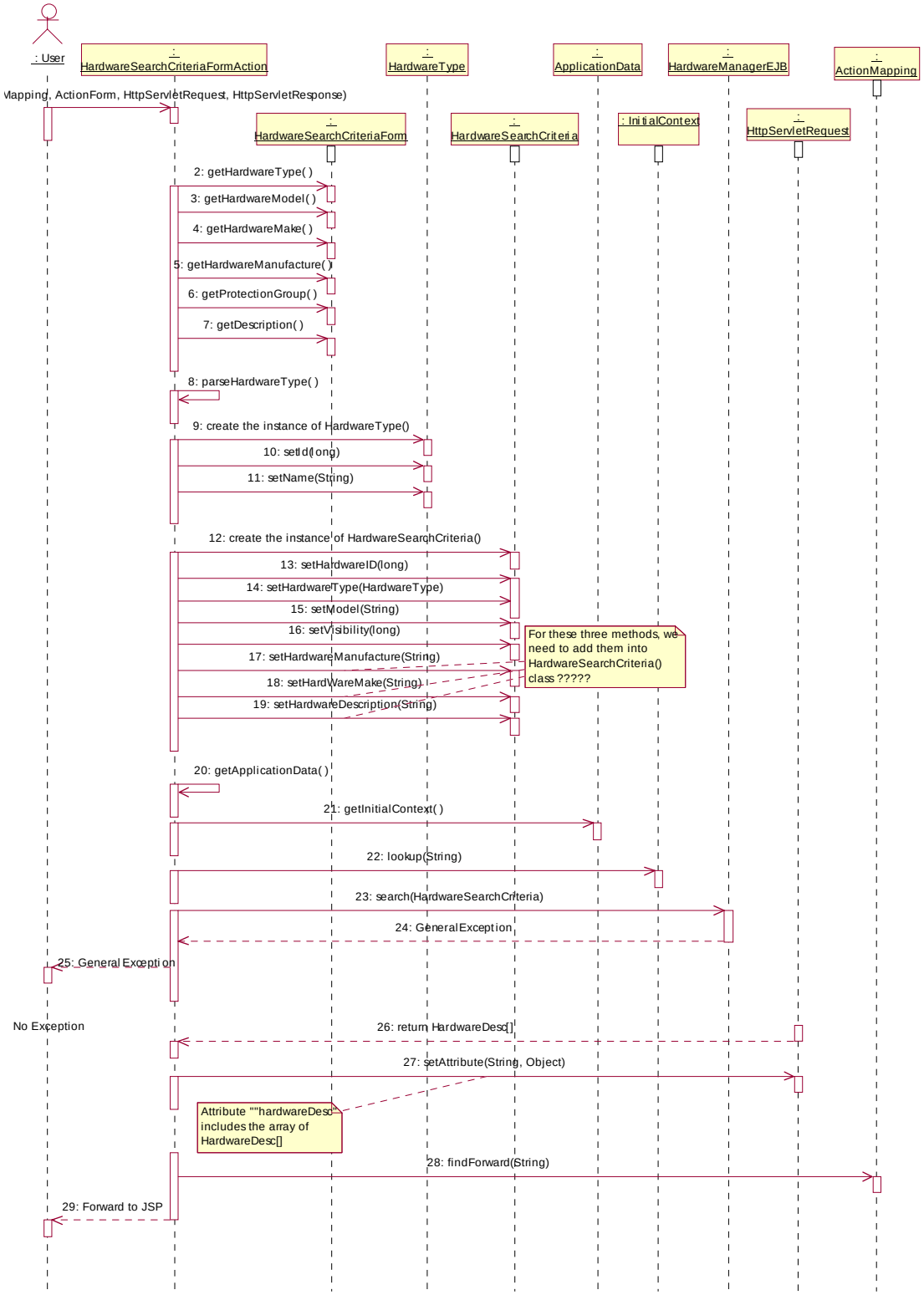
118. BuildProtocolSearchCriteriaFormAction.performAction



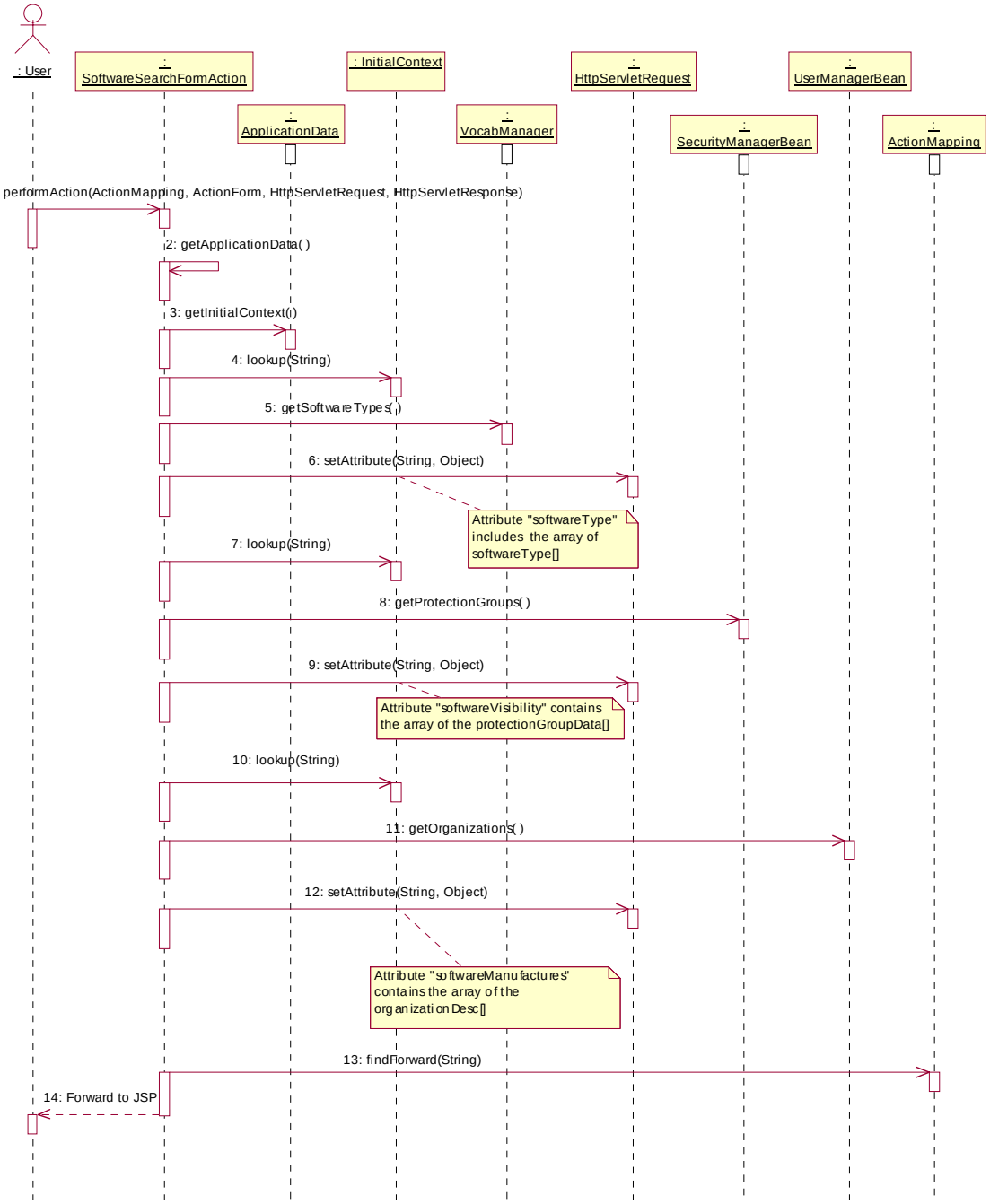
119. BuildHardwareSearchFormAction.performAction



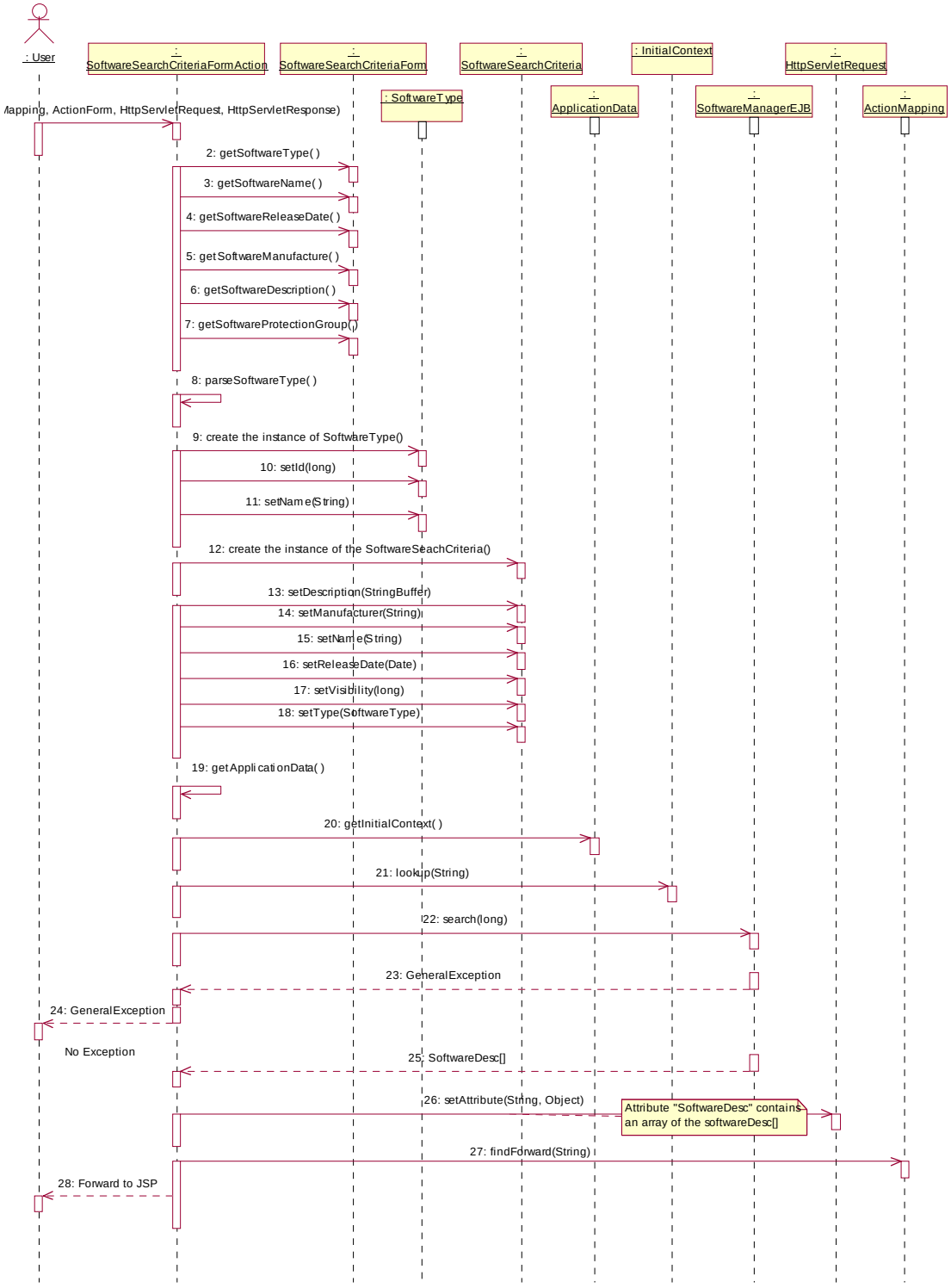
120. BuildHardwareSearchCriteriaFormAction.performAction


: User
: HardwareSearchCriteriaFormAction
: HardwareType
: ApplicationData
: HardwareManagerEJB
: ActionMapping
Mapping, ActionForm, HttpServletRequest, HttpServletResponse)
: HardwareSearchCriteriaForm
: HardwareSearchCriteria
: InitialContext
: HttpServletRequest
2: getHardwareType( )
3: getHardwareModel( )
4: getHardwareMake( )
5: getHardwareManufacture( )
6: getProtectionGroup( )
7: getDescription( )
8: parseHardwareType( )
9: create the instance of HardwareType()
10: setId(long)
11: setName(String)
12: create the instance of HardwareSearchCriteria()
13: setHardwareID(long)
14: setHardwareType(HardwareType)
15: setModel(String)
16: setVisibility(long)
17: setHardwareManufacture(String)
18: setHardWareMake(String)
19: setHardwareDescription(String)
For these three methods, we need to add them into HardwareSearchCriteria() class ?????
20: getApplicationData( )
21: getInitialContext( )
22: lookup(String)
23: search(HardwareSearchCriteria)
24: GeneralException
25: GeneralException
No Exception
26: return HardwareDesc[]
27: setAttribute(String, Object)
Attribute ""hardwareDesc" includes the array of HardwareDesc[]
28: findForward(String)
29: Forward to JSP

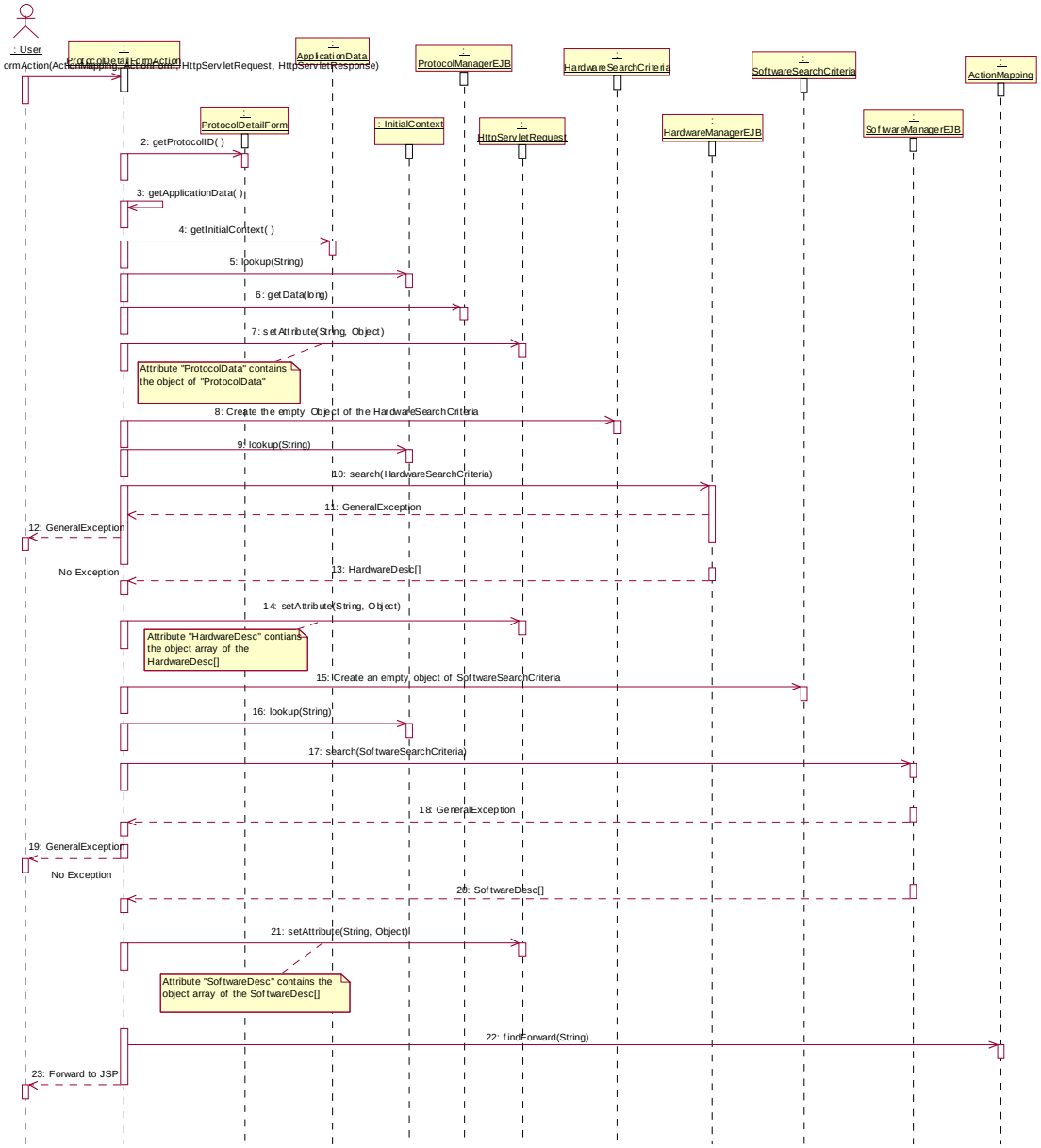
121. BuildSoftwareSearchFormAction.performAction



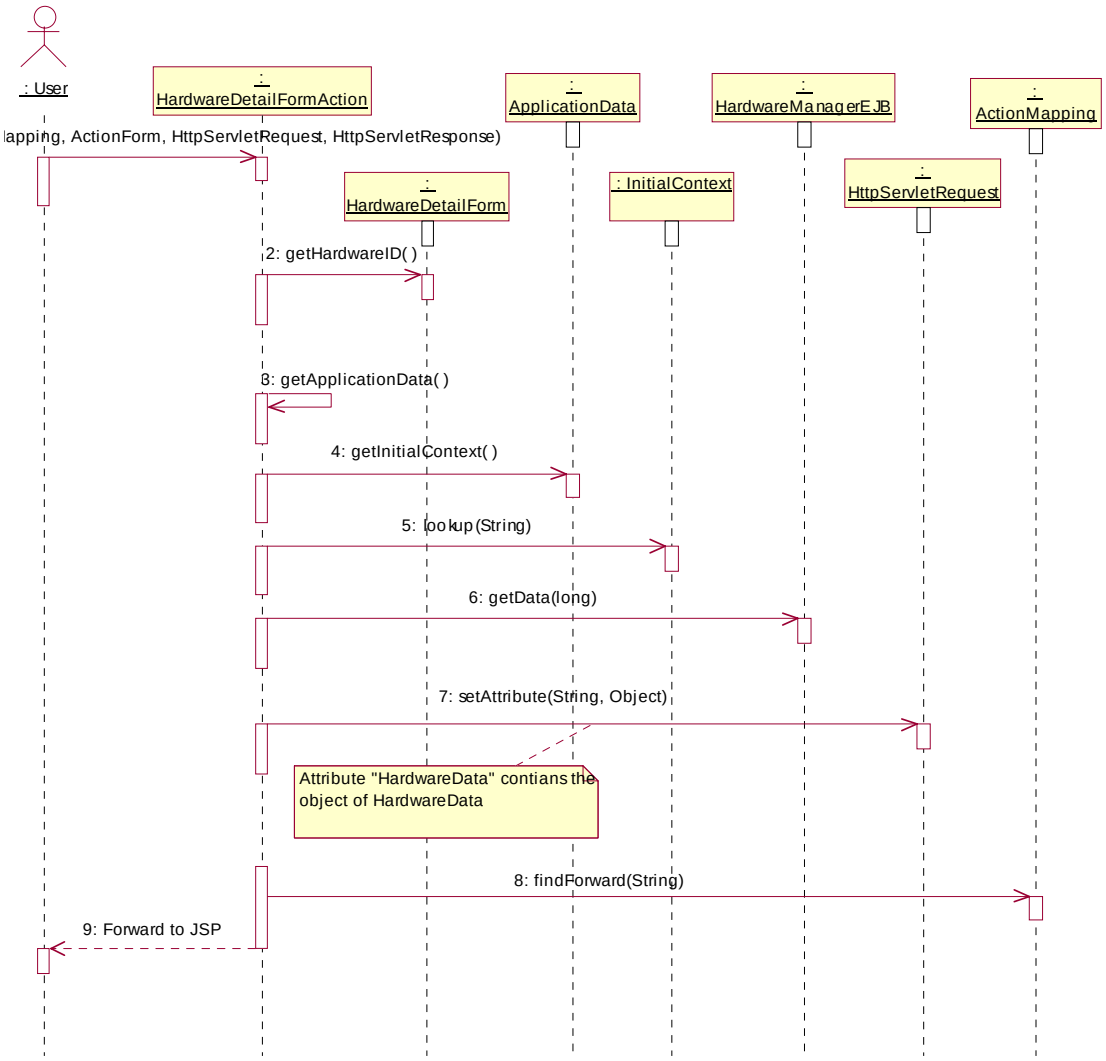
122. BuildSoftwareSearchCriteriaFormAction.performAction


: User
: SoftwareSearchCriteriaFormAction
: SoftwareSearchCriteriaForm
: SoftwareSearchCriteria
: InitialContext
: HttpServletRequest
: SoftwareType
: ApplicationData
: SoftwareManagerEJB
: ActionMapping
Mapping, ActionForm, HttpServletRequest, HttpServletResponse)
2: getSoftwareType( )
3: getSoftwareName( )
4: getSoftwareReleaseDate( )
5: getSoftwareManufacture( )
6: getSoftwareDescription( )
7: getSoftwareProtectionGroup( )
8: parseSoftwareType( )
9: create the instance of SoftwareType()
10: setId(long)
11: setName(String)
12: create the instance of the SoftwareSeachCriteria()
13: setDescription(StringBuffer)
14: setManufacturer(String)
15: setName(String)
16: setReleaseDate(Date)
17: setVisibility(long)
18: setType(SoftwareType)
19: getApplicationData( )
20: getInitialContext( )
21: lookup(String)
22: search(long)
23: GeneralException
24: GeneralException
No Exception
25: SoftwareDesc[]
26: setAttribute(String, Object)
Attribute "SoftwareDesc" contains an array of the softwareDesc[]
27: findForward(String)
28: Forward to JSP

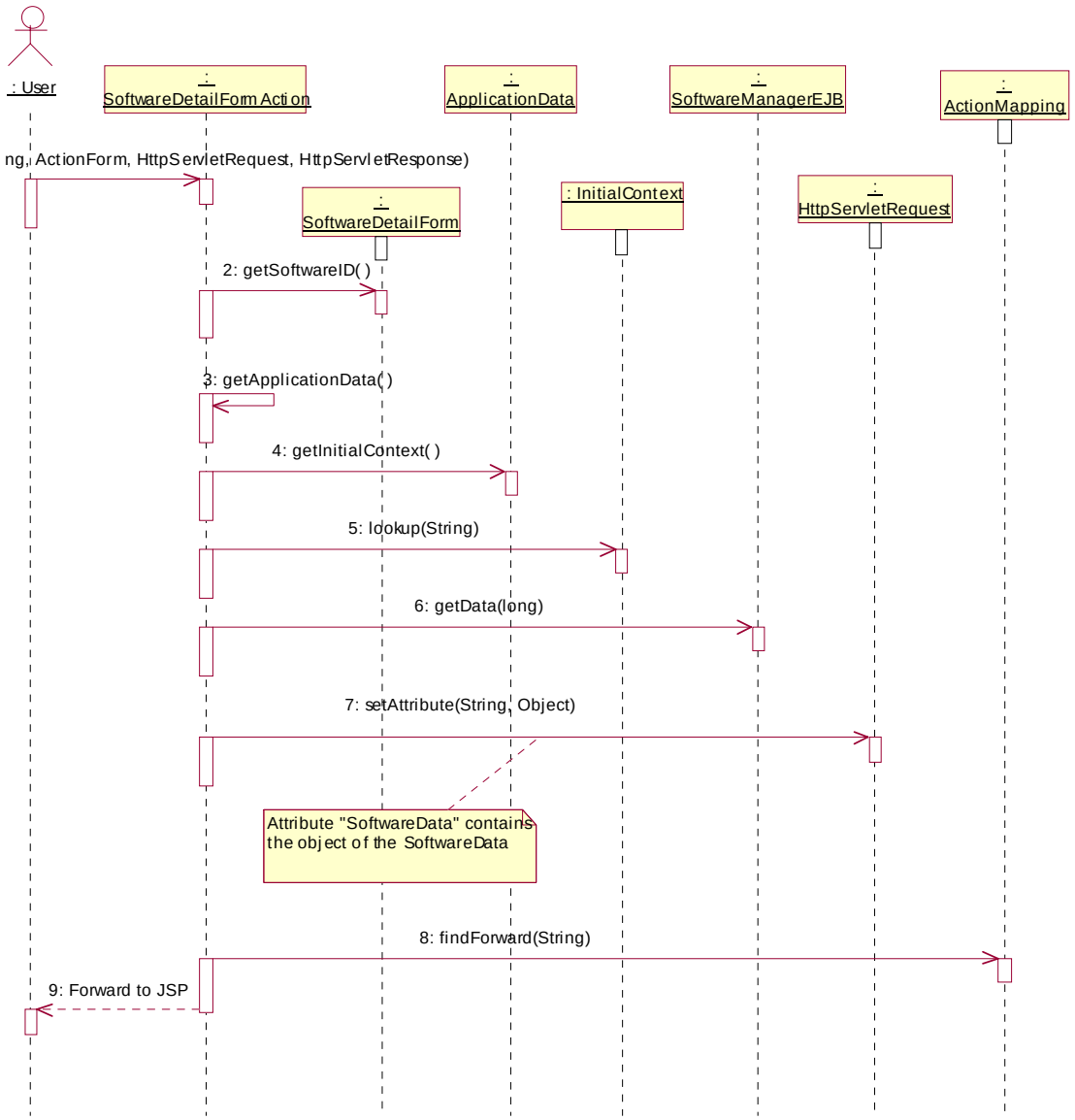
123. BuildProtocolDetailFormAction.performAction



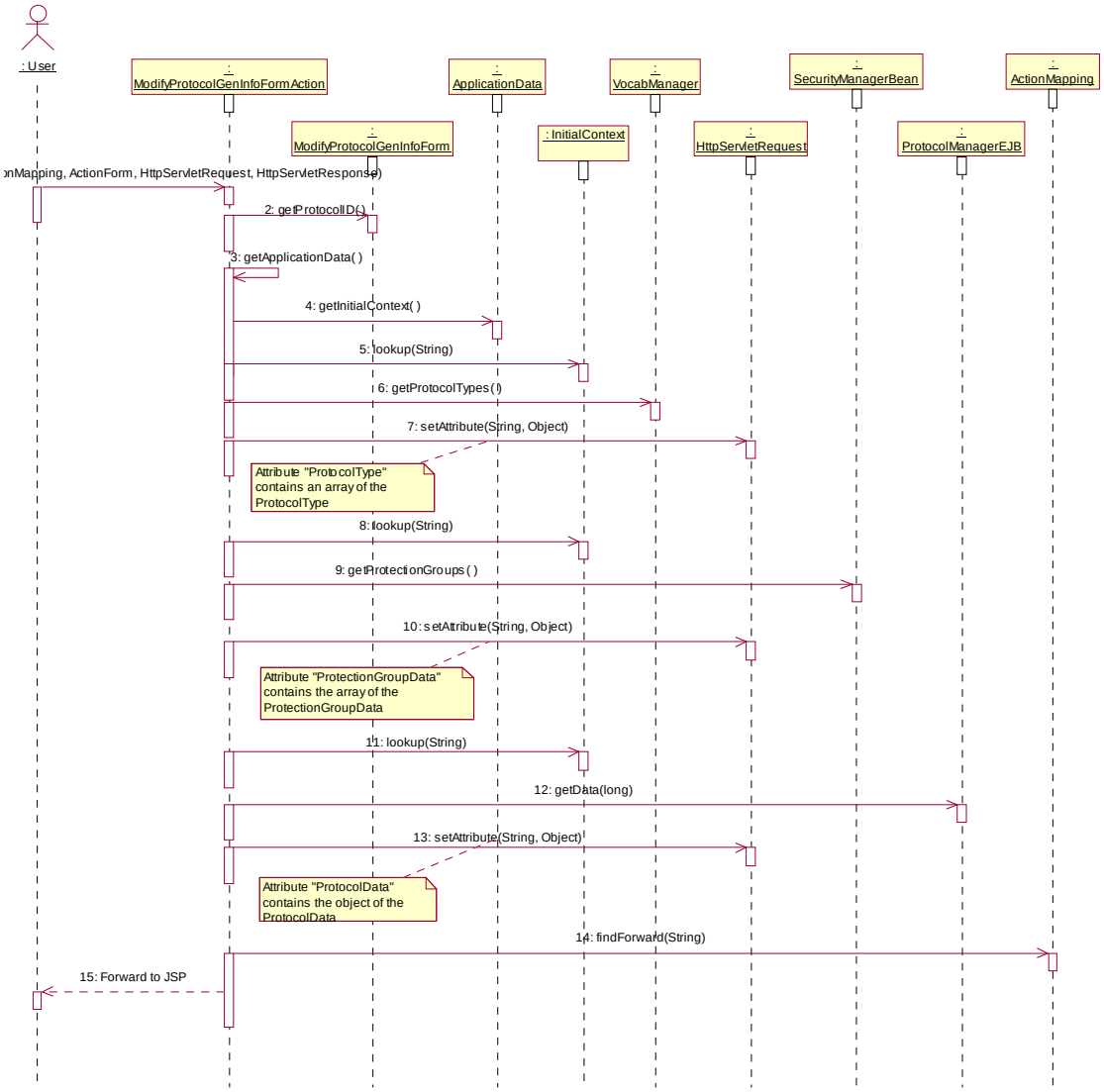
124. BuildHardwareDetailFormAction.performAction



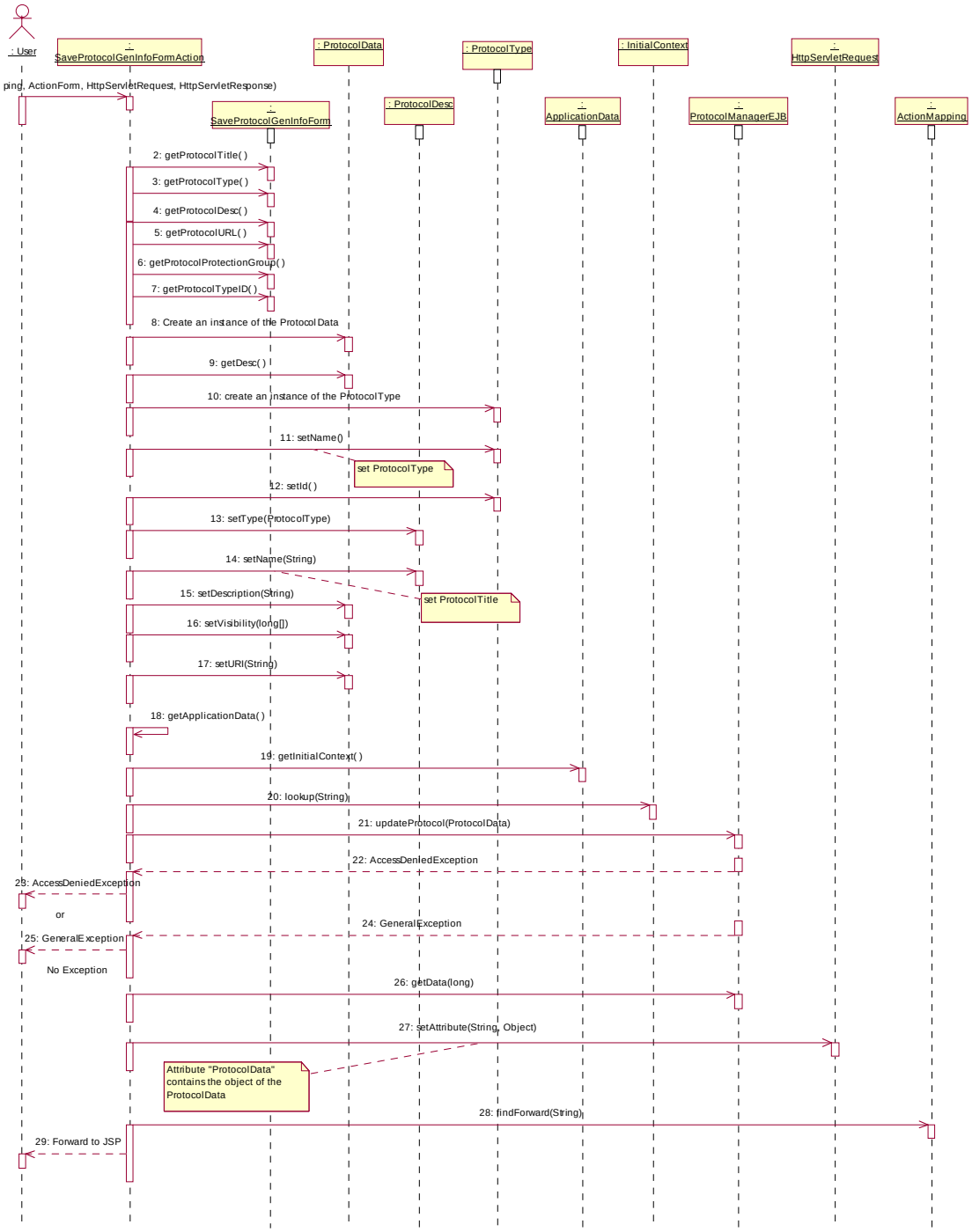
125. BuildSoftwareDetailFormAction.performAction



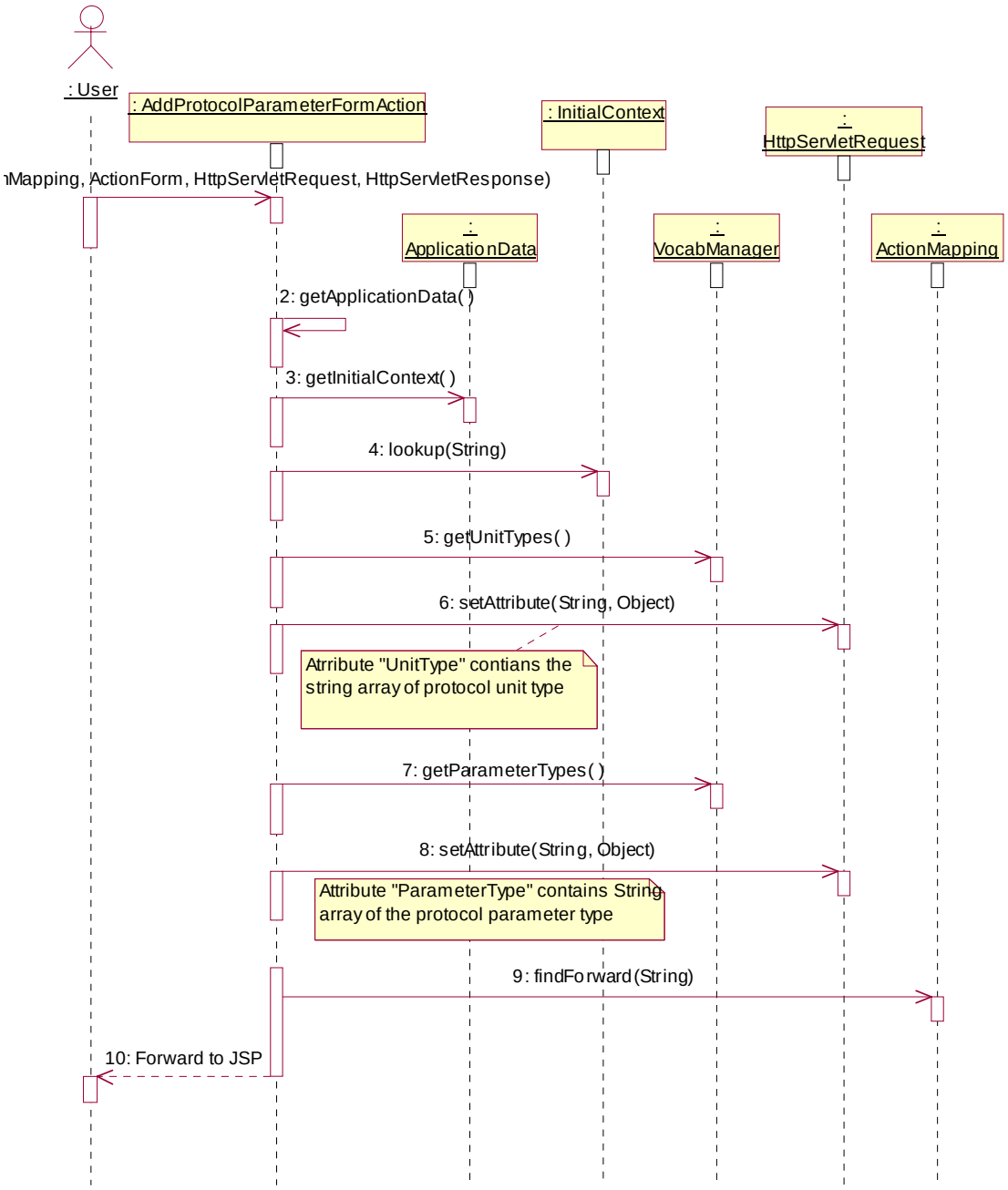
126. BuildModifyProtocolGenInfoFormAction.performAction



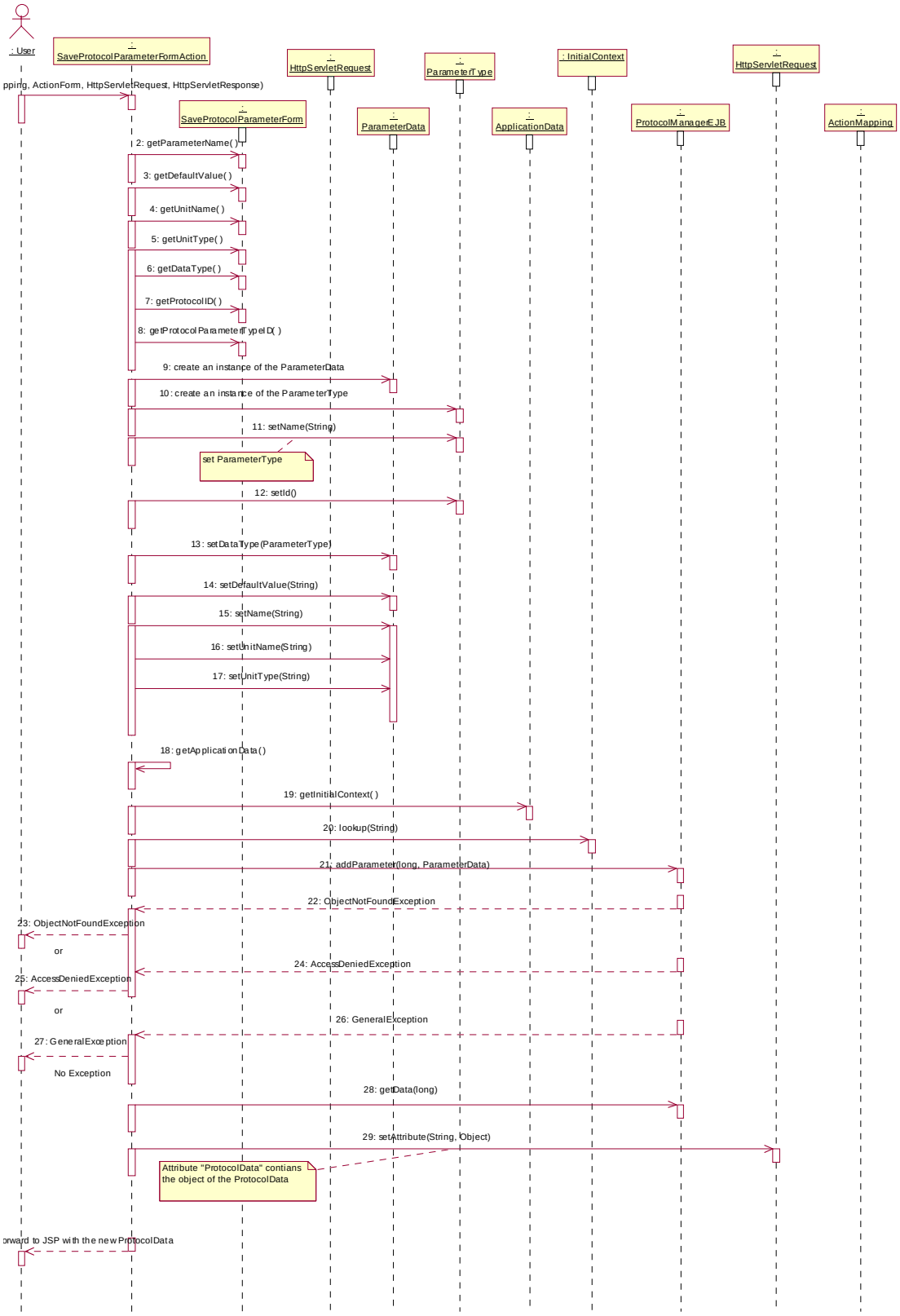
127. BuildSaveProtocolGenInfoAction.performAction



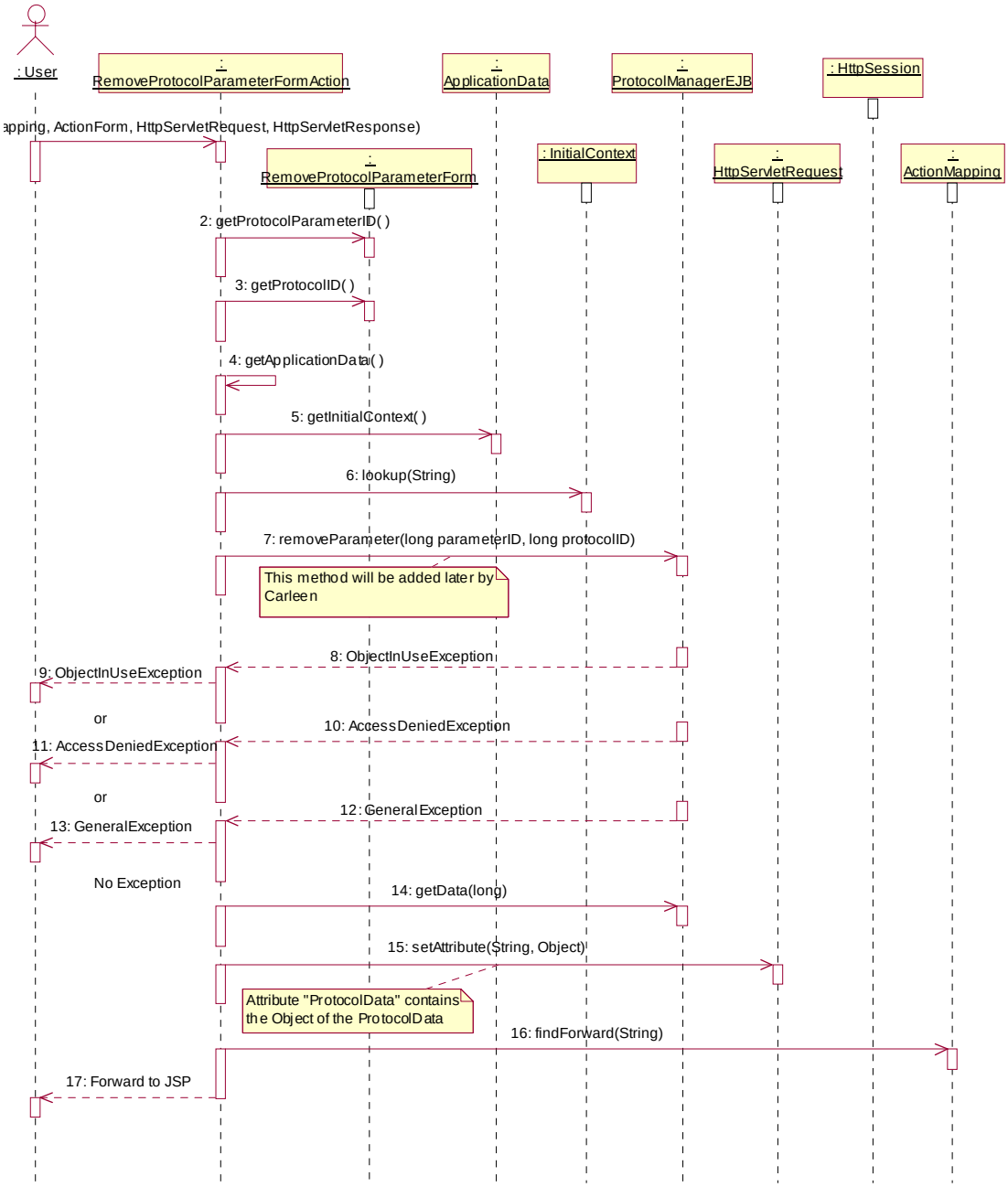
128. BuildAddProtocolParameterFormAction.performAction



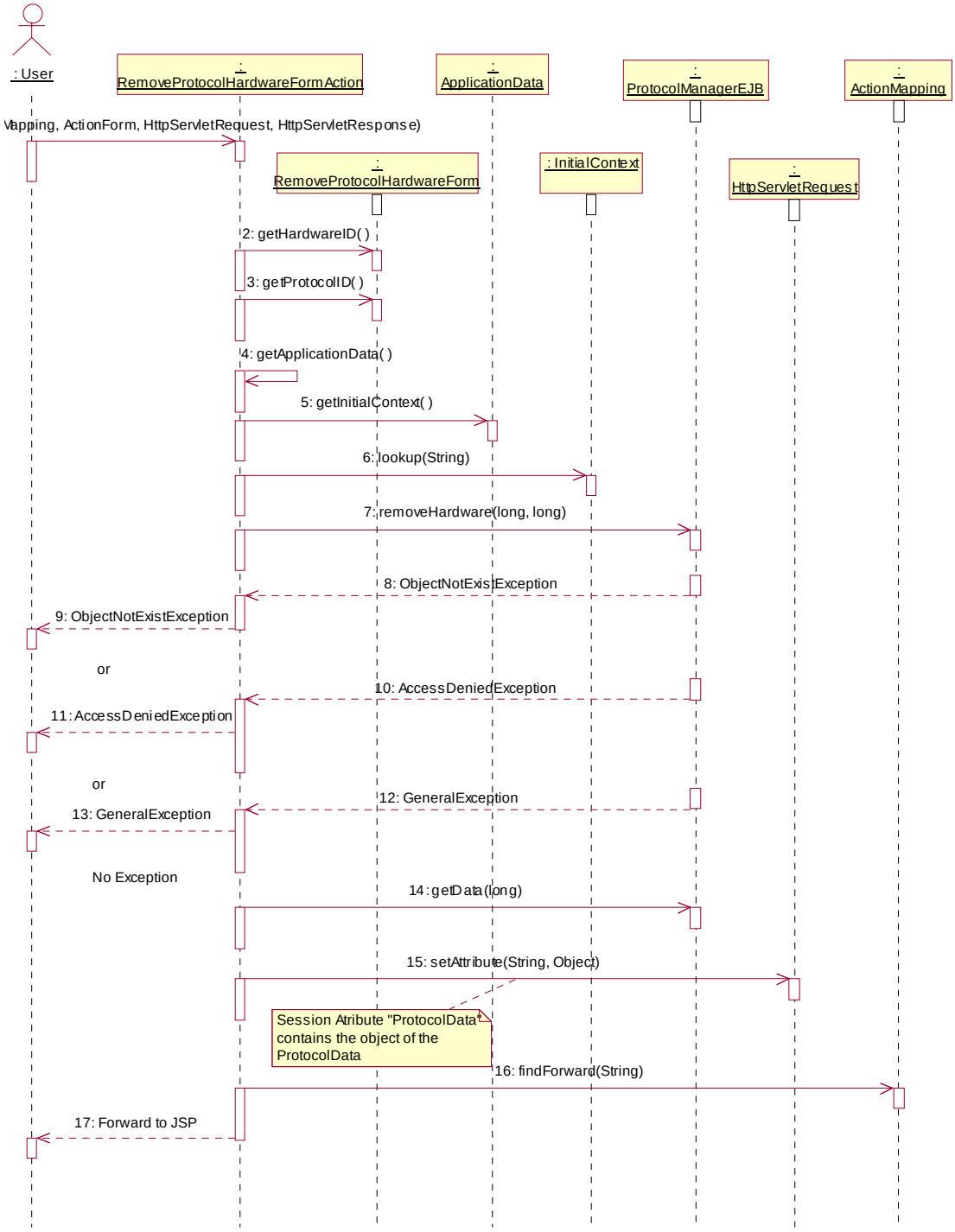
129. BuildSaveProtocolParameterFormAction.performAction


: User
: SaveProtocolParameterFormAction
: HttpServletRequest
: ParameterType
: InitialContext
: HttpServletRequest
pping, ActionForm, HttpServletRequest, HttpServletResponse)
: SaveProtocolParameterForm
: ParameterData
: ApplicationData
: ProtocolManagerEJB
: ActionMapping
2: getParameterName( )
3: getDefaultValue( )
4: getUnitName( )
5: getUnitType( )
6: getDataType( )
7: getProtocolID( )
8: getProtocolParameterTypeID( )
9: create an instance of the ParameterData
10: create an instance of the ParameterType
11: setName(String)
set ParameterType
12: setId()
13: setDataType(ParameterType)
14: setDefaultValue(String)
15: setName(String)
16: setUnitName(String)
17: setUnitType(String)
18: getApplicationData()
19: getInitialContext( )
20: lookup(String)
21: addParameter(long, ParameterData)
22: ObjectNotFoundException
23: ObjectNotFoundException
or
24: AccessDeniedException
25: AccessDeniedException
or
26: GeneralException
27: GeneralException
No Exception
28: getData(long)
29: setAttribute(String, Object)
Attribute "ProtocolData" contians the object of the ProtocolData
orward to JSP with the new ProtocolData

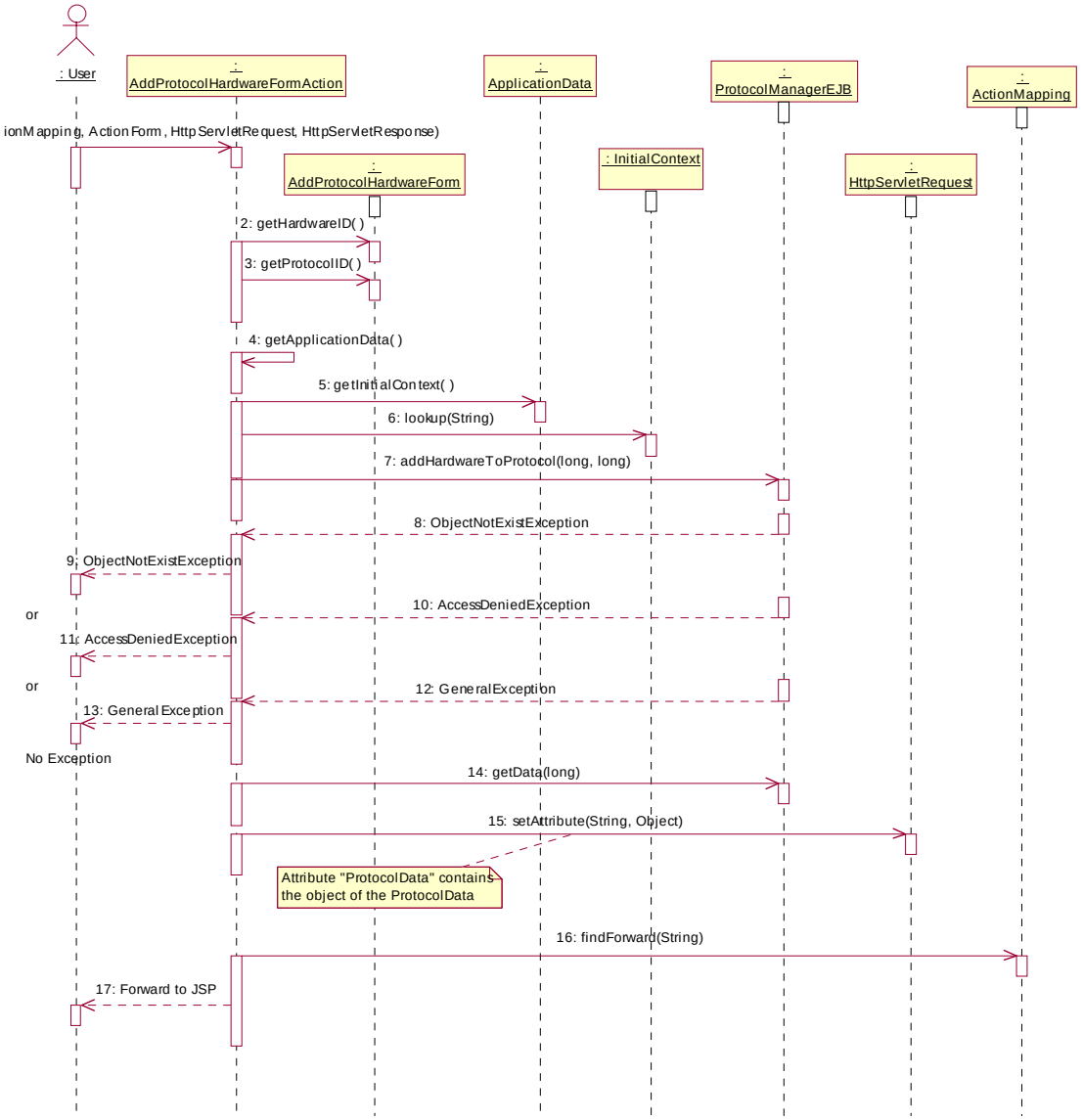
130. BuildRemoveProtocolParameterFormAction.performAction



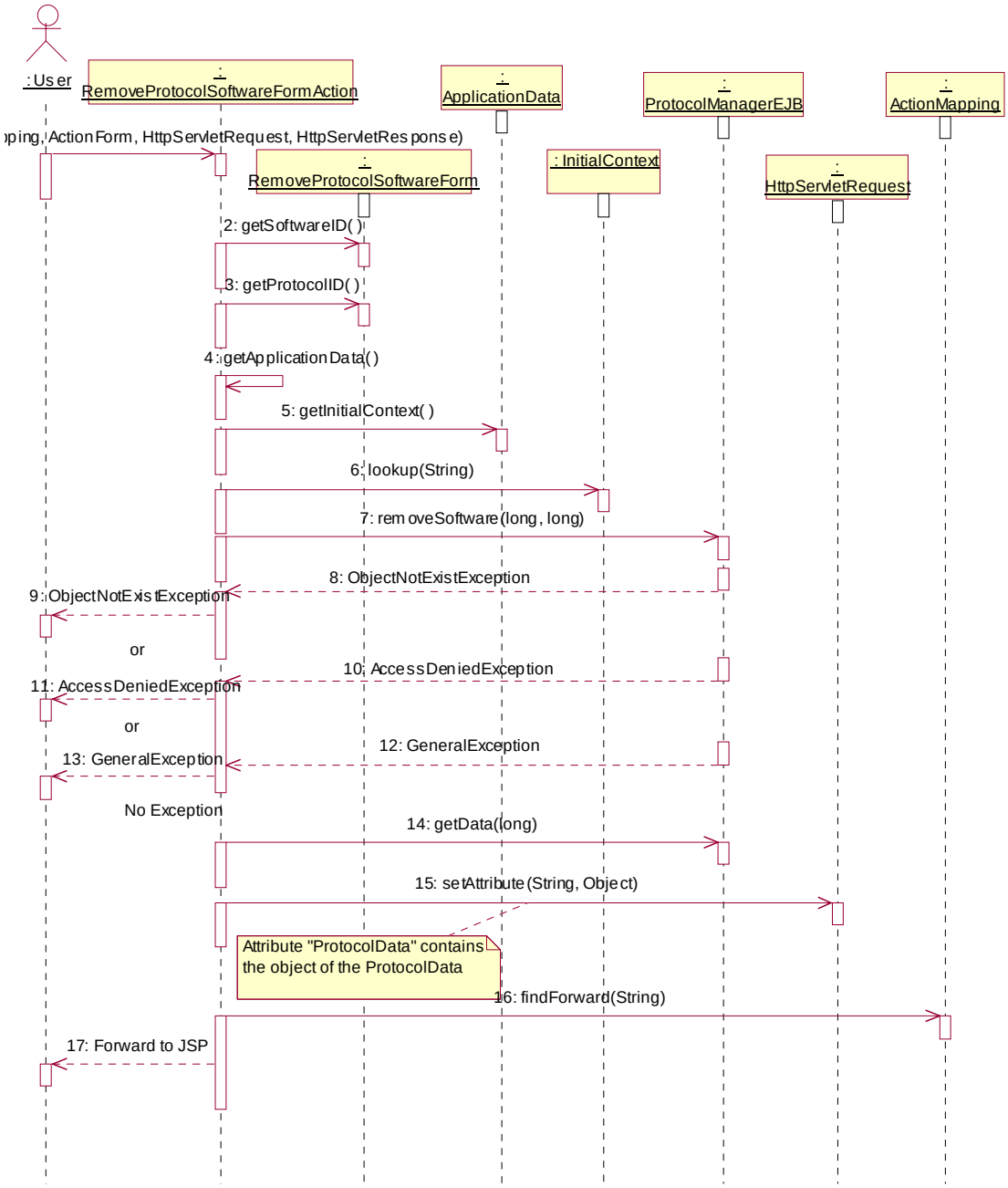
131. BuildRemoveProtocolHardwareFormAction.performAction



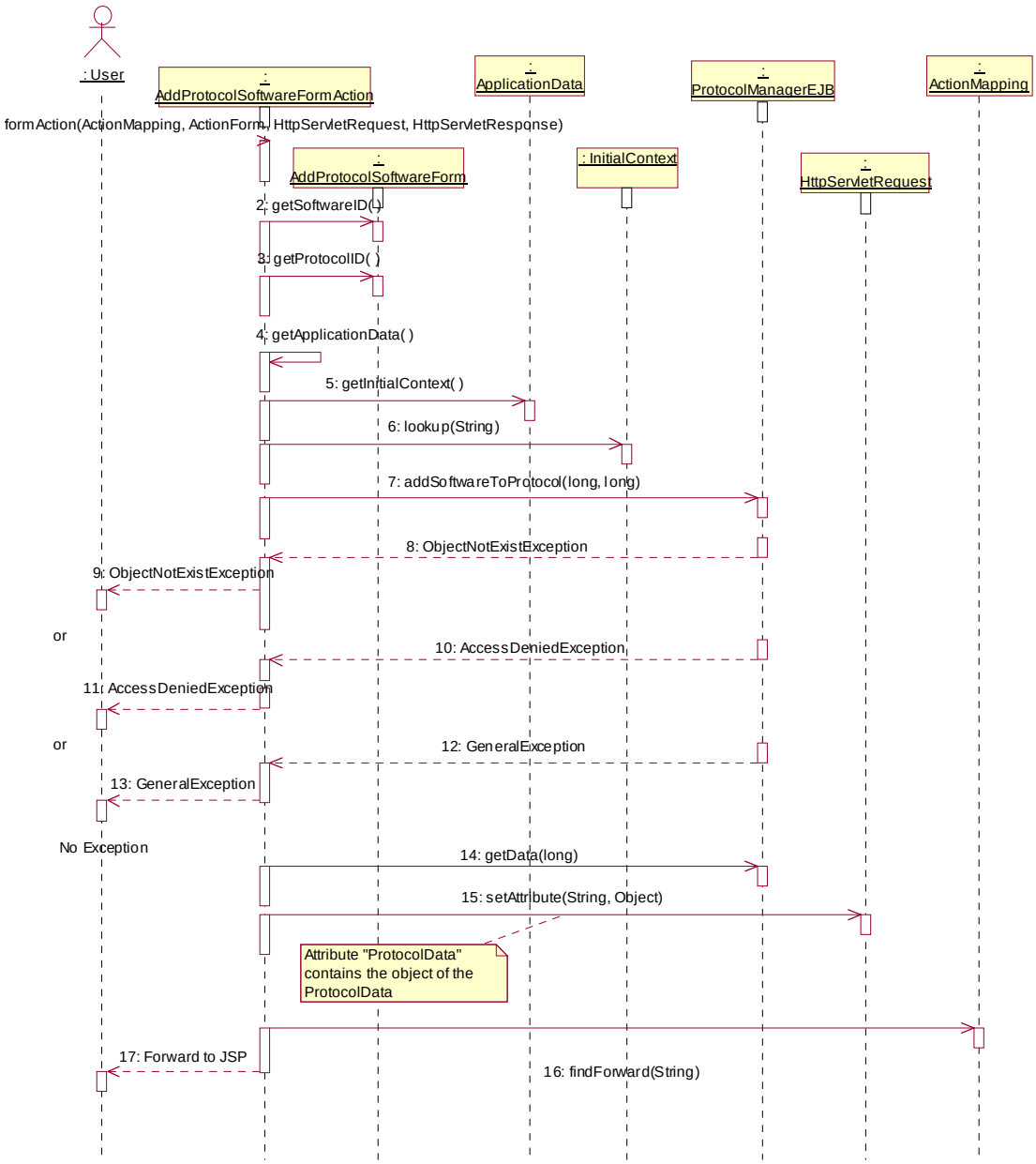
132. BuildAddProtocolHardwareFormAction.performAction



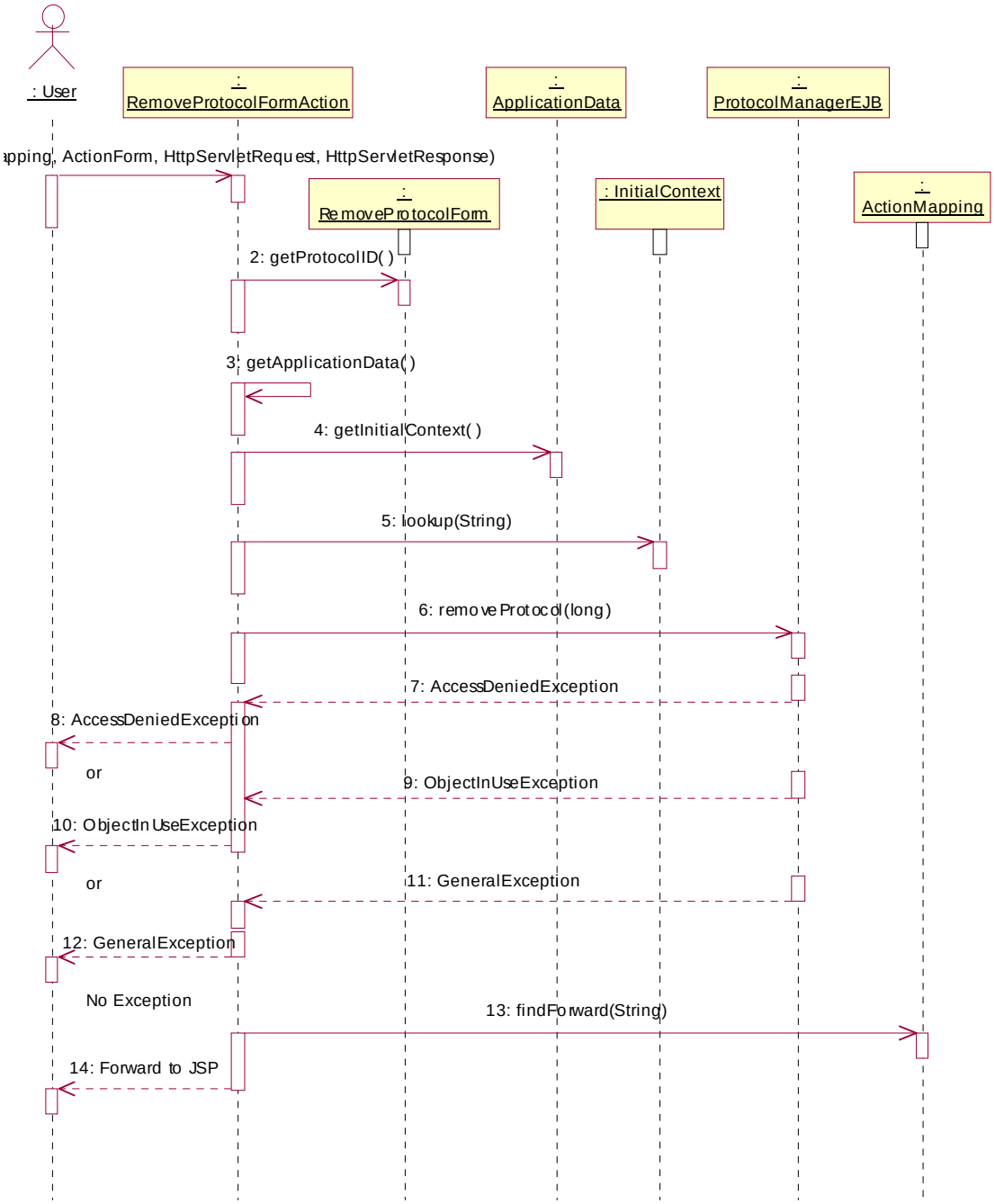
133. BuildRemoveProtocolSoftwareFormAction.PerformAction



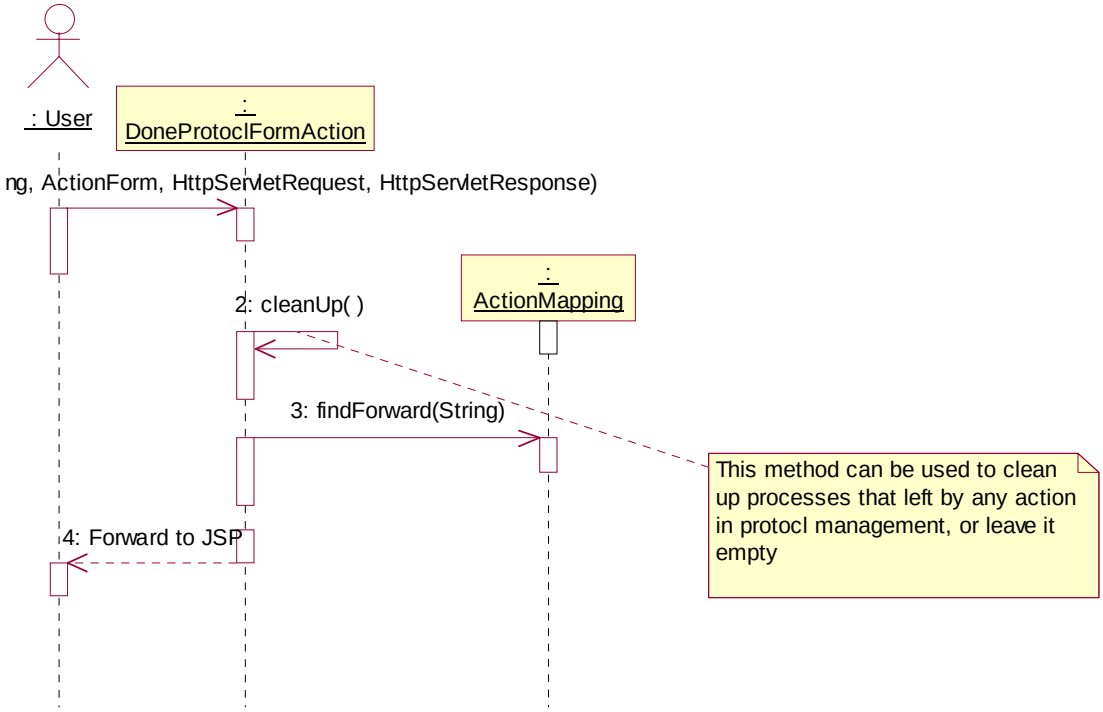
134. BuildAddProtocolSoftwareFormAction.performAction



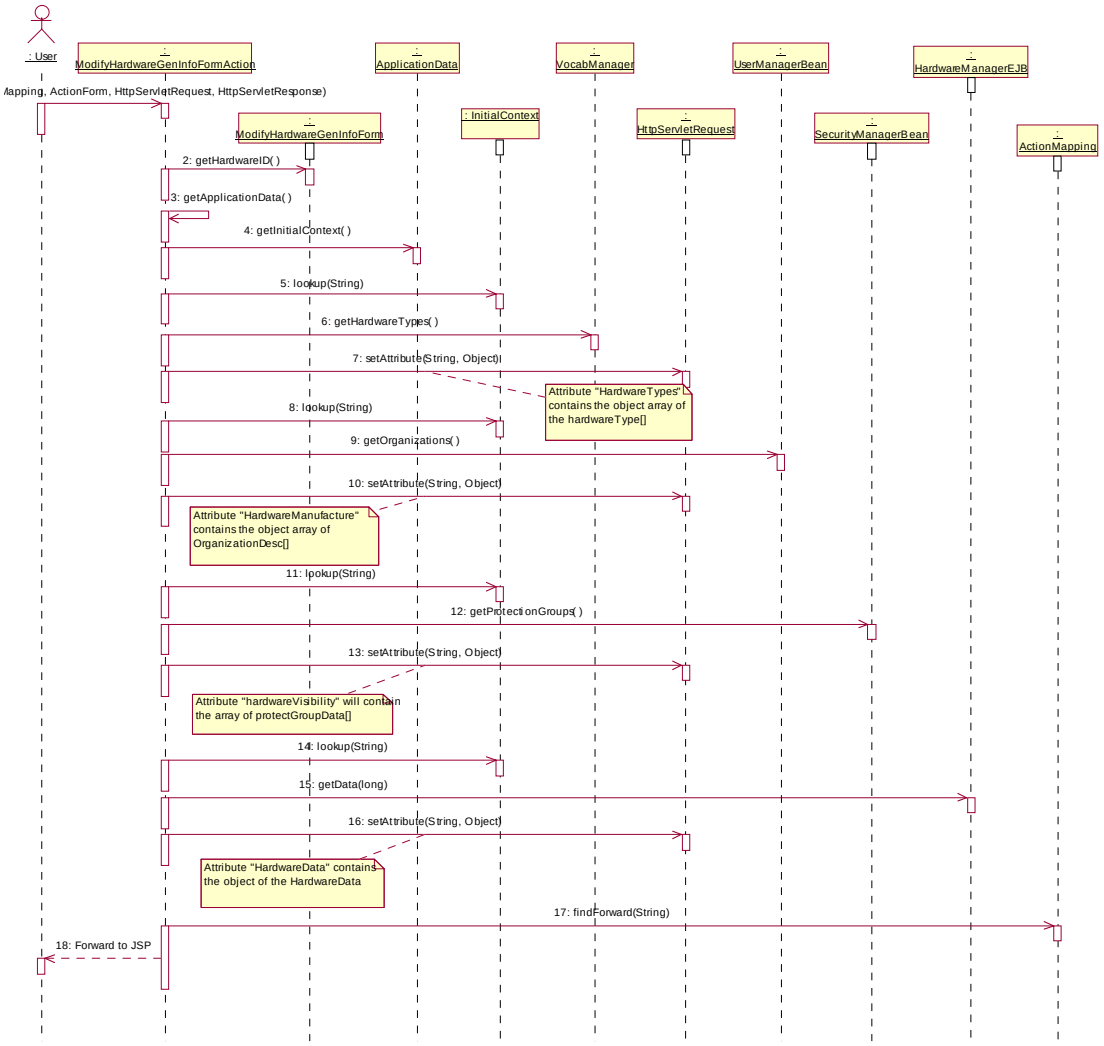
135. BuildRemoveProtocolFormAction.performAction



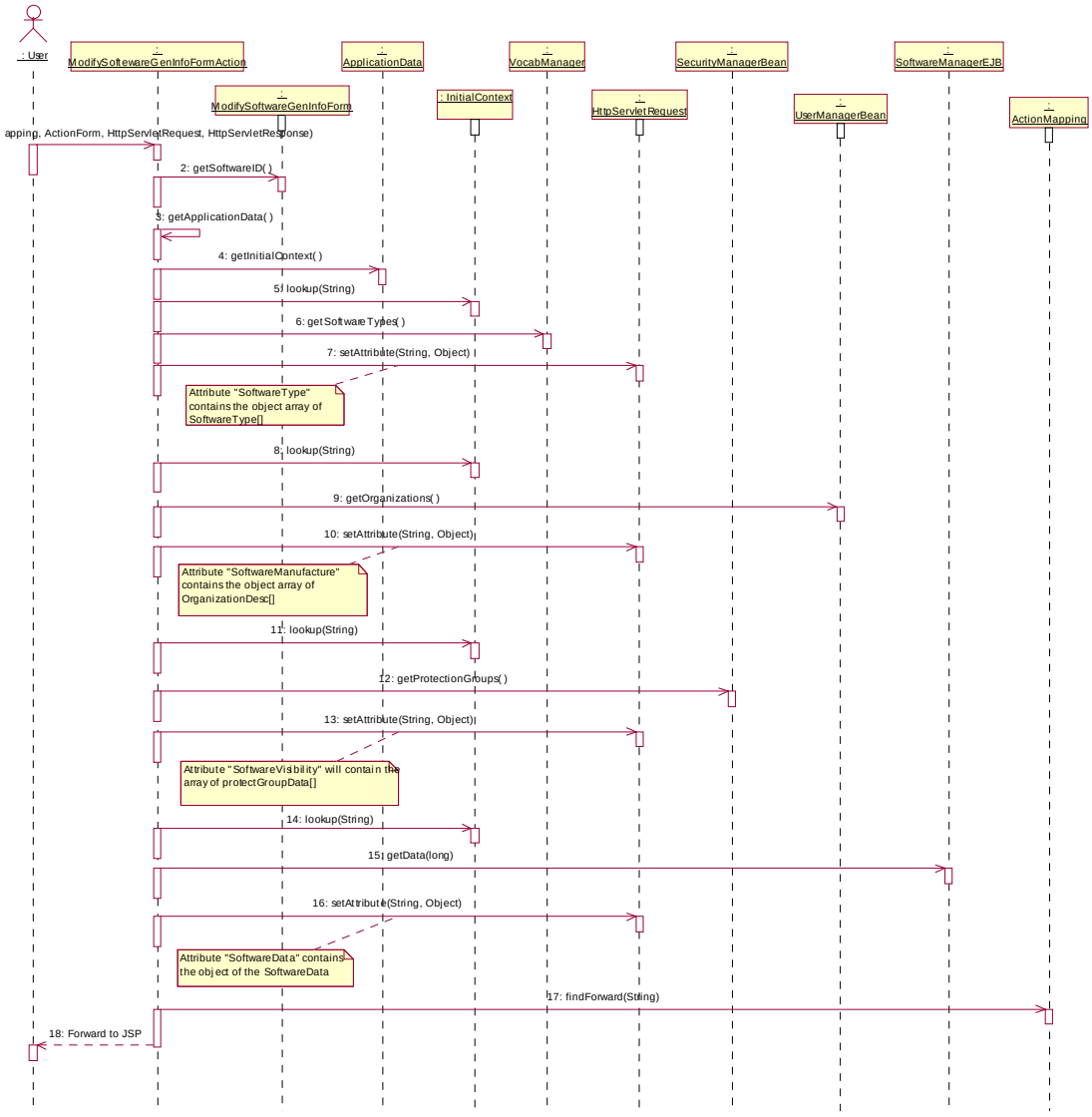
136. BuildDoneProtoclFormAction.performAction



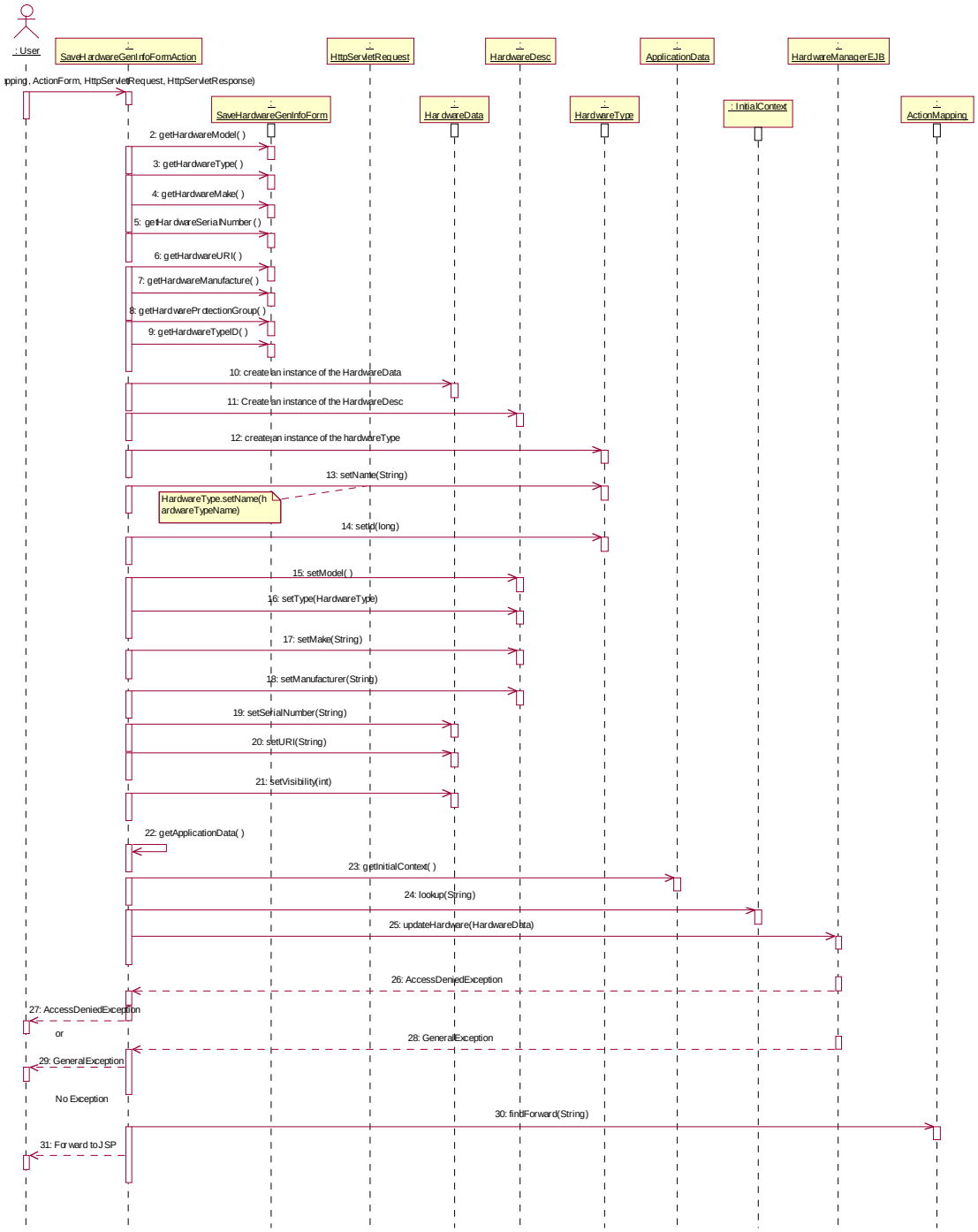
137. BuildModifyHardwareGenInfoFormAction.performAction



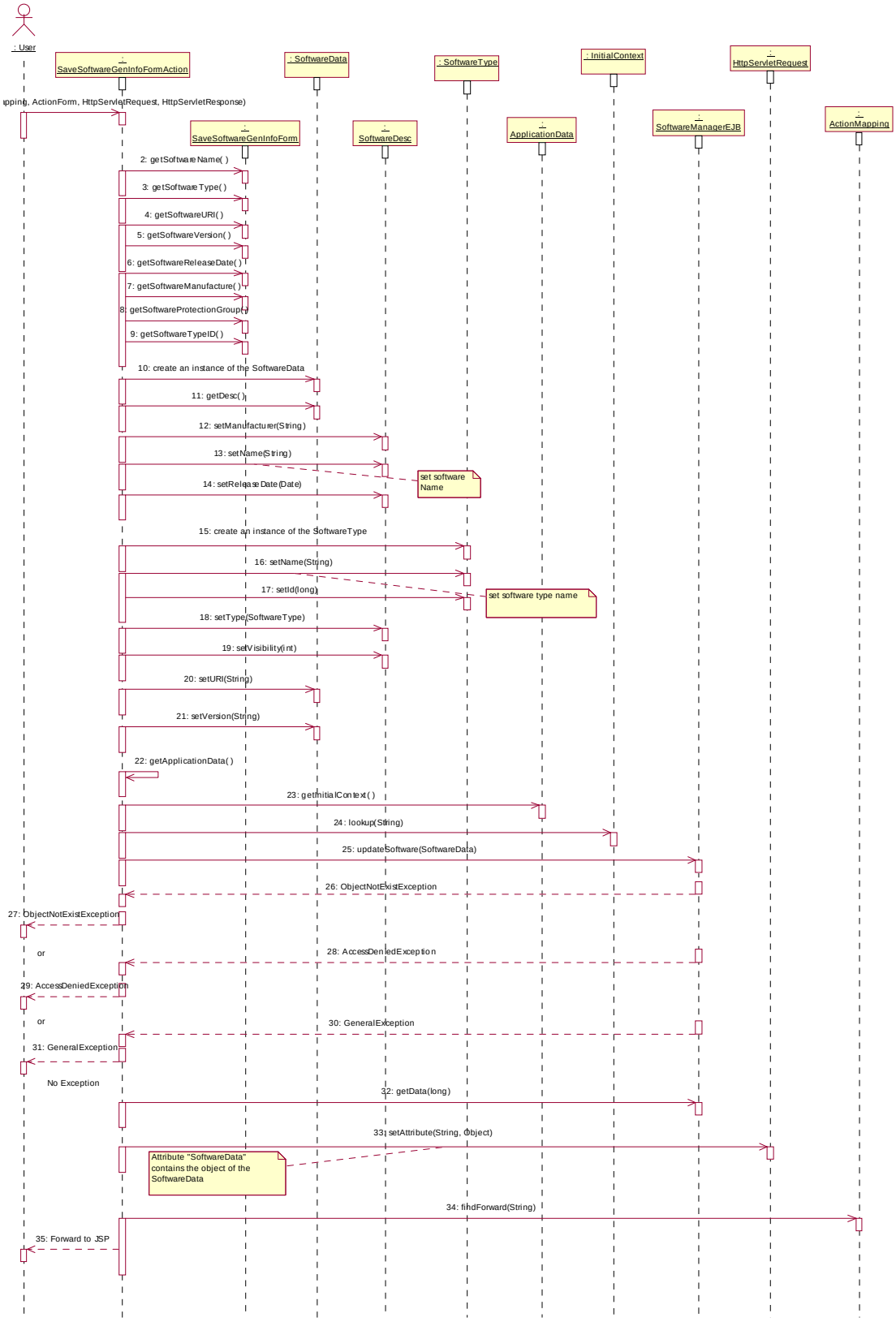
138. BuildModifySoftwareGenInfoFormAction.performAction



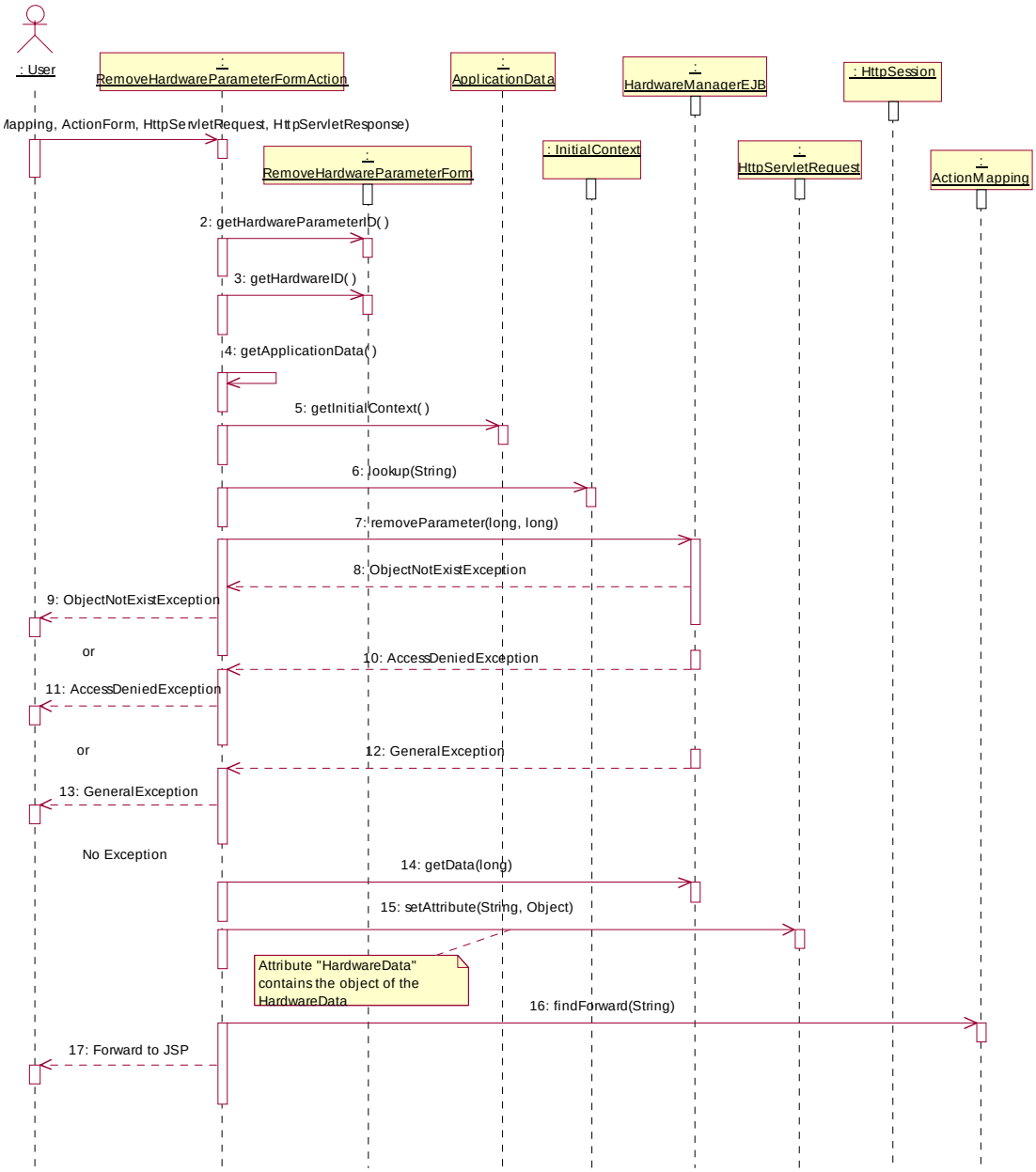
139. BuildSaveHardwareGenInfoFormAction.performAction



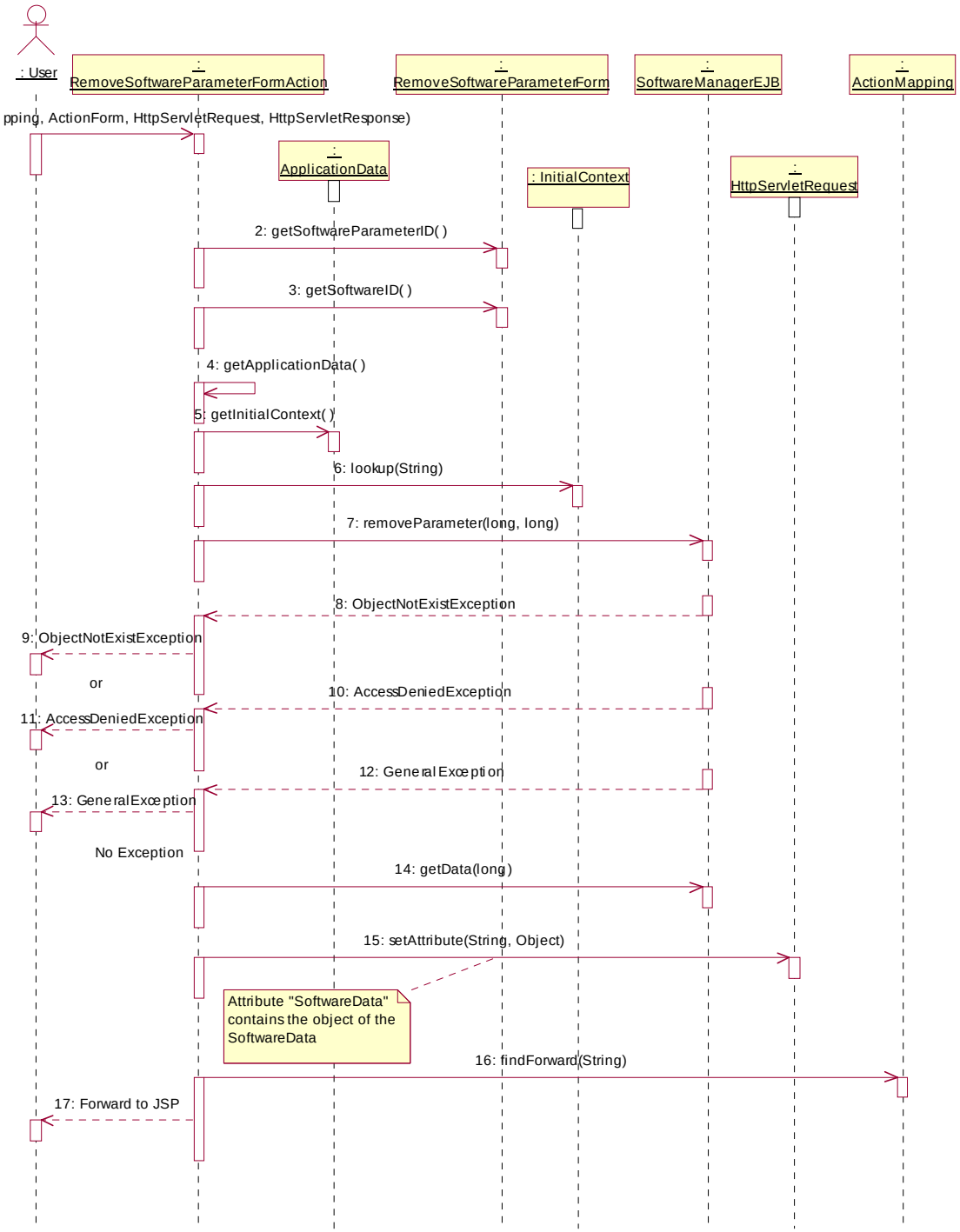
140. BuildSaveSoftwareGenInfoFormAction.performAction


: User
: SaveSoftwareGenInfoFormAction
: SoftwareData
: SoftwareType
: InitialContext
: HttpServletRequest
ipping, ActionForm, HttpServletRequest, HttpServletResponse)
: SaveSoftwareGenInfoForm
: SoftwareDesc
: ApplicationData
: SoftwareManagerEJB
: ActionMapping
2: getSoftwareName( )
3: getSoftwareType( )
4: getSoftwareURI( )
5: getSoftwareVersion( )
6: getSoftwareReleaseDate( )
7: getSoftwareManufacture( )
8: getSoftwareProtectionGroup( )
9: getSoftwareTypeID( )
10: create an instance of the SoftwareData
11: getDesc( )
12: setManufacturer(String)
13: setName(String)
set software Name
14: setReleaseDate(Date)
15: create an instance of the SoftwareType
16: setName(String)
17: setId(long)
set software type name
18: setType(SoftwareType)
19: setVisibility(int)
20: setURI(String)
21: setVersion(String)
22: getApplicationData( )
23: getInitialContext( )
24: lookup(String)
25: updateSoftware(SoftwareData)
26: ObjectNotExistException
27: ObjectNotExistException
or
28: AccessDeniedException
29: AccessDeniedException
or
30: GeneralException
31: GeneralException
No Exception
32: getData(long)
33: setAttribute(String, Object)
Attribute "SoftwareData" contains the object of the SoftwareData
34: findForward(String)
35: Forward to JSP

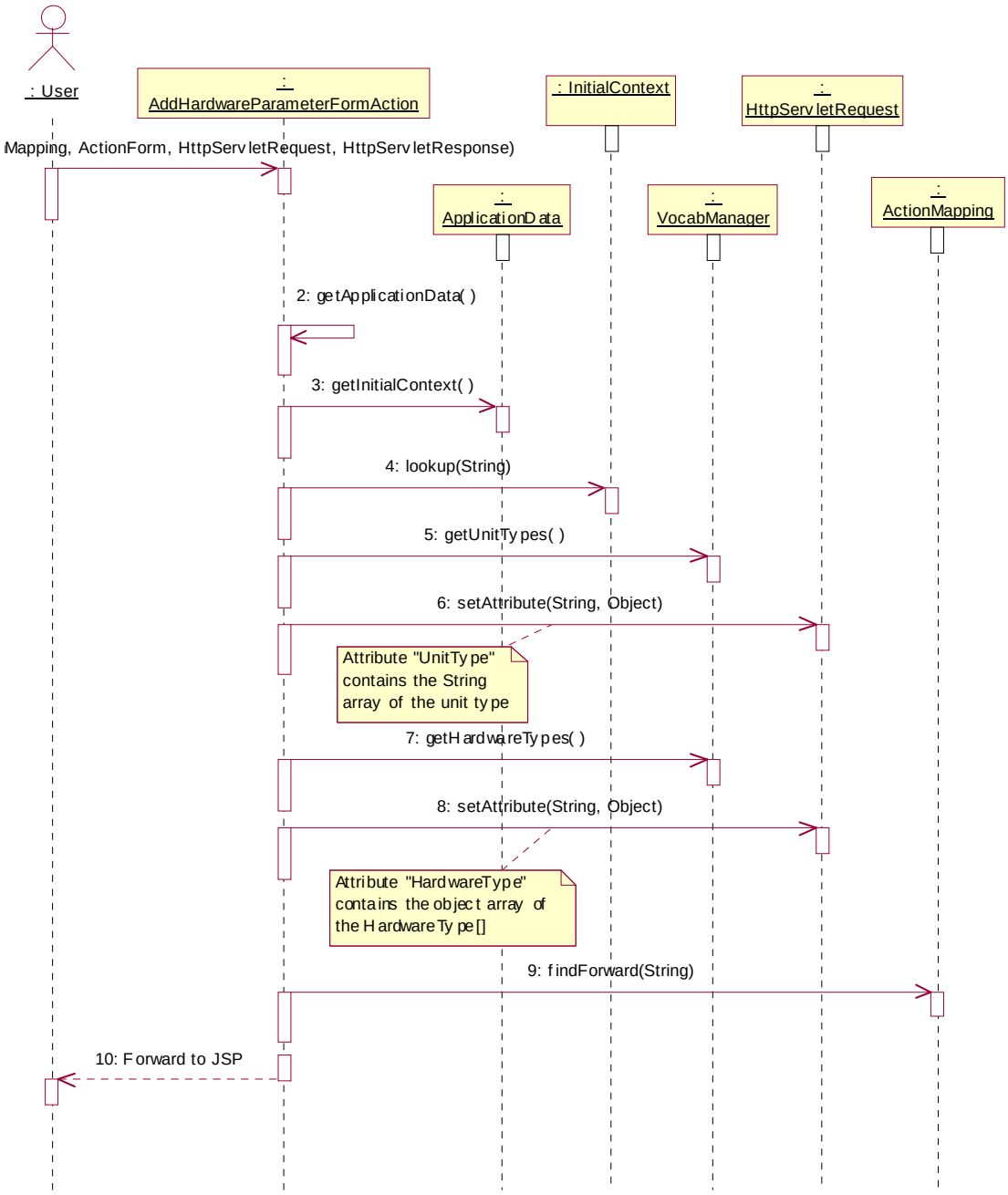
141. BuildRemoveHardwareParameterFormAction.performAction



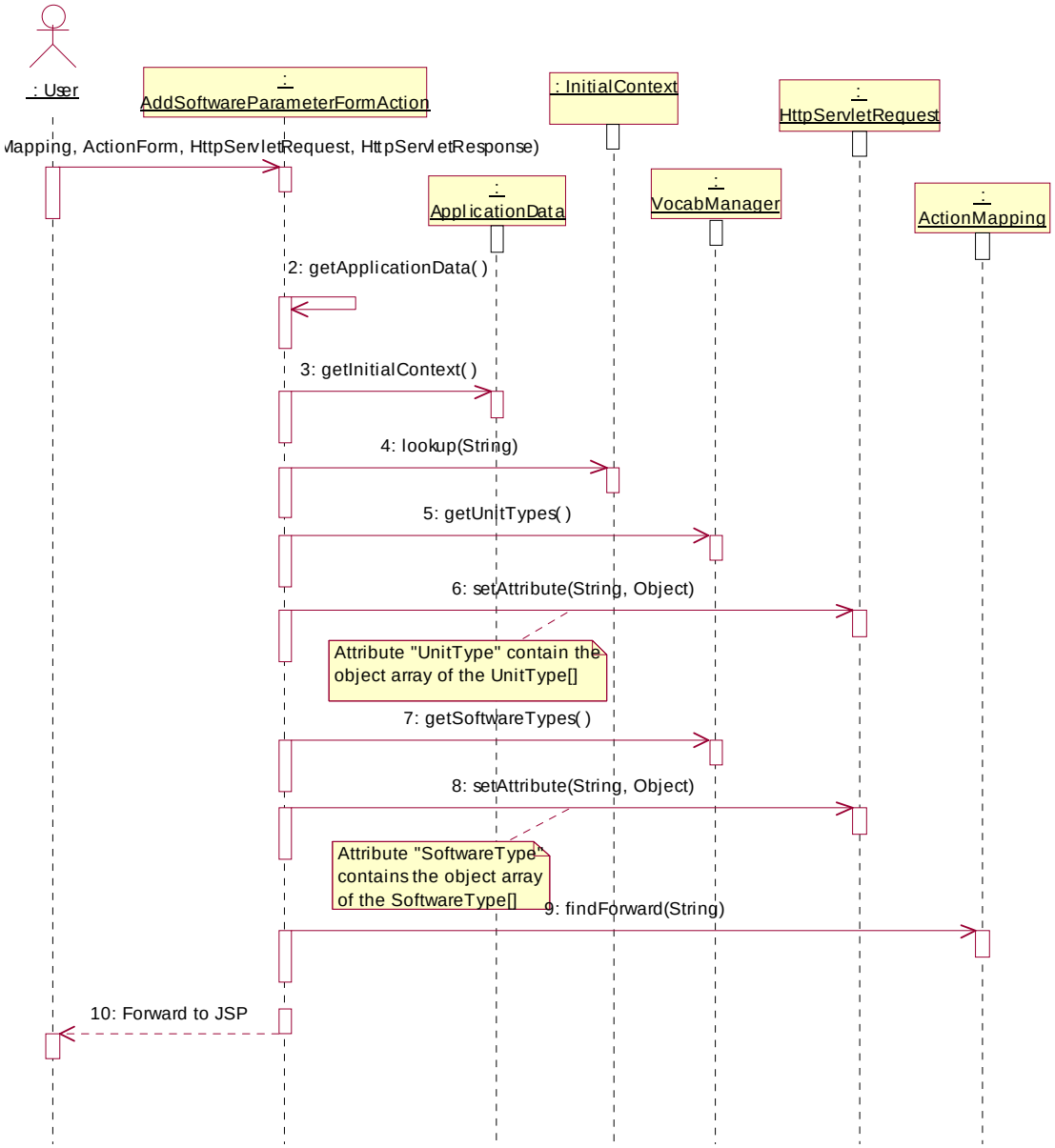
142. BuildRemoveSoftwareParameterFormAction.performAction



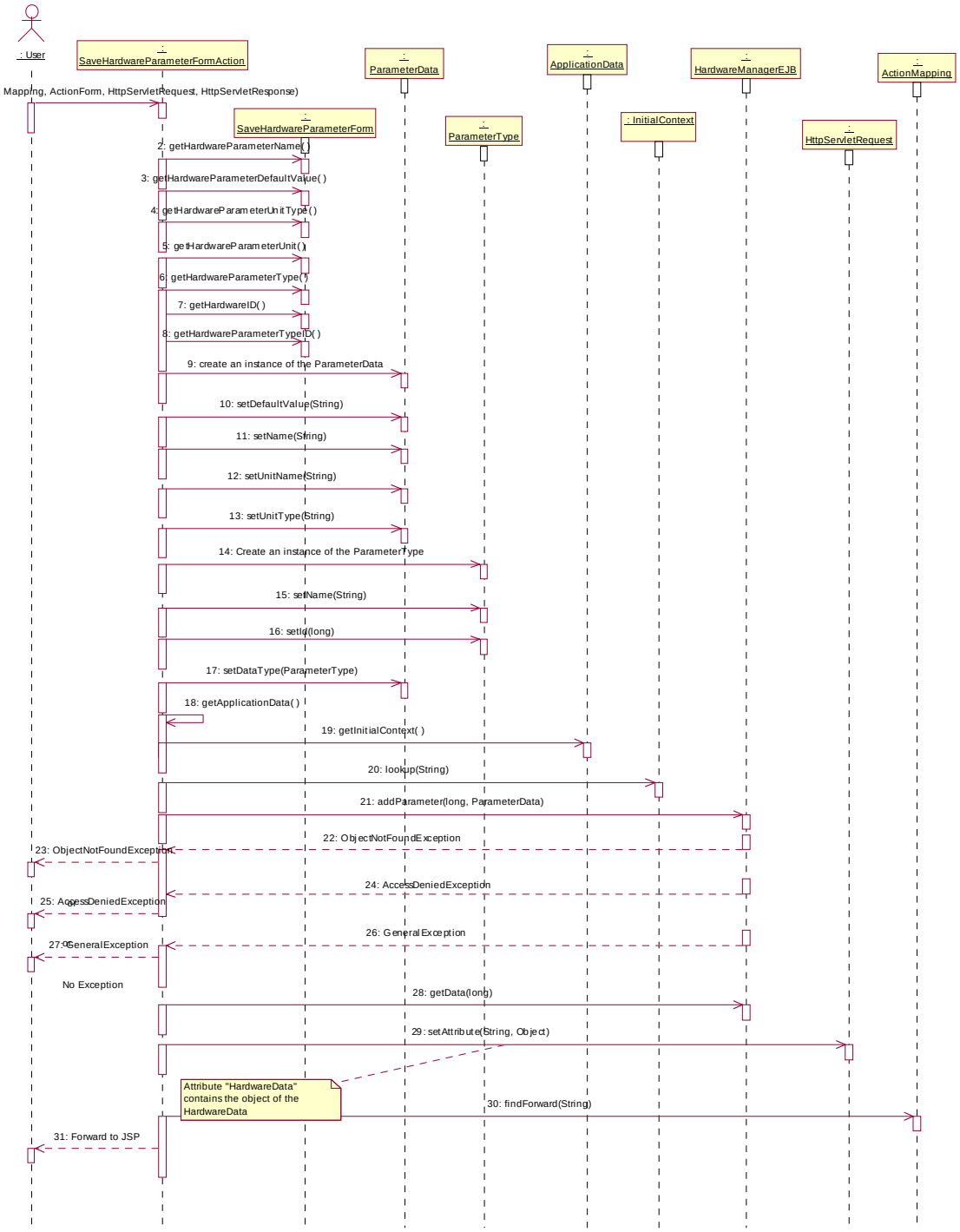
143. BuildAddHardwareParameterFormAction.performAction



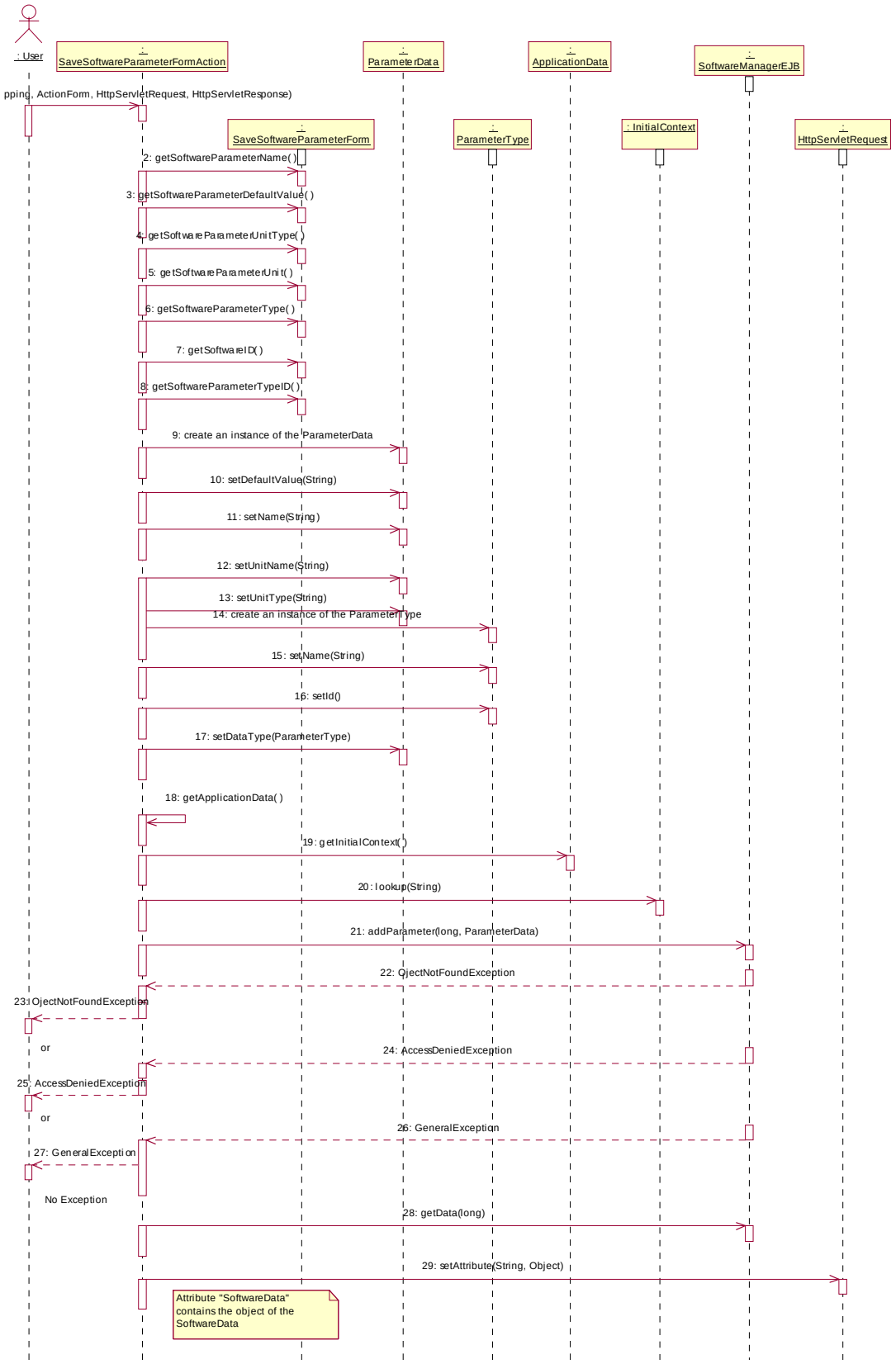
144. BuildAddSoftwareParameterFormAction.performAction



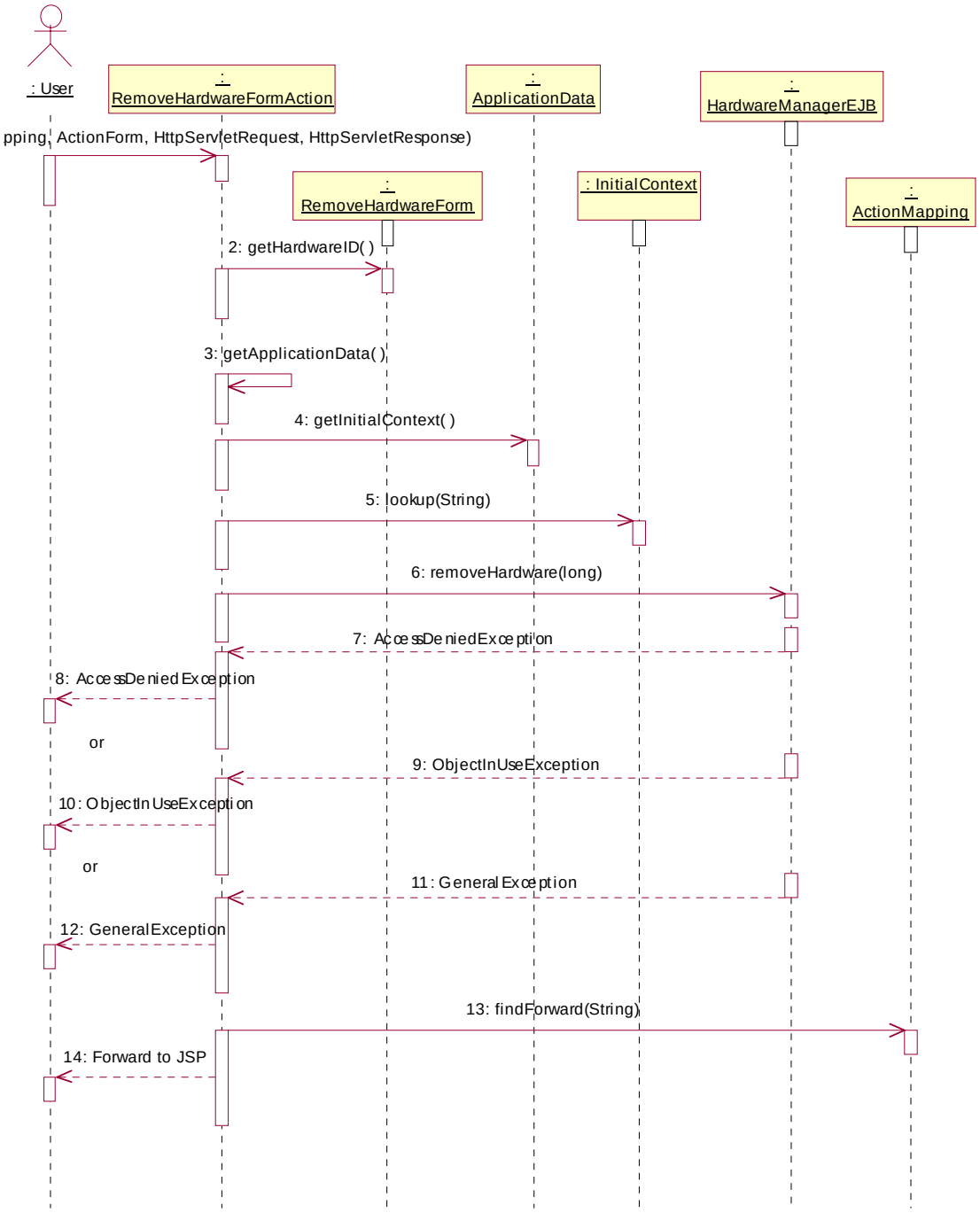
145. BuildSaveHardwareParameterFormAction.performAction



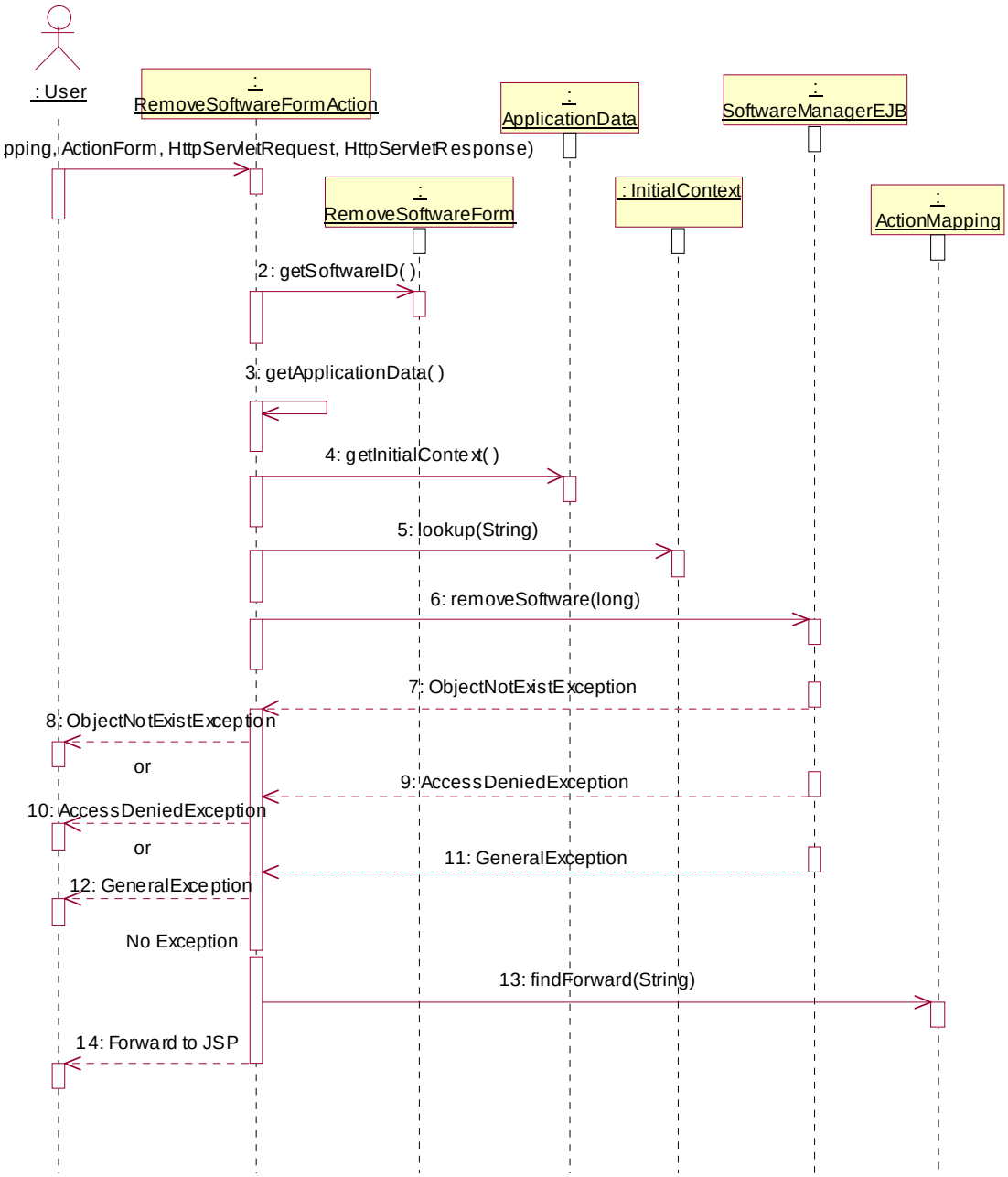
146. BuildSaveSoftwareParameterFormAction.performAction


: User
: SaveSoftwareParameterFormAction
: ParameterData
: ApplicationData
: SoftwareManagerEJB
pping, ActionForm, HttpServletRequest, HttpServletResponse)
: SaveSoftwareParameterForm
: ParameterType
: InitialContext
: HttpServletRequest
2: getSoftwareParameterName( )
3: getSoftwareParameterDefaultValue( )
4: getSoftwareParameterUnitType( )
5: getSoftwareParameterUnit( )
6: getSoftwareParameterType( )
7: getSoftwareID( )
8: getSoftwareParameterTypeID( )
9: create an instance of the ParameterData
10: setDefaultValue(String)
11: setName(String)
12: setUnitName(String)
13: setUnitType(String)
14: create an instance of the ParameterType
15: setName(String)
16: setId()
17: setDataType(ParameterType)
18: getApplicationData( )
19: getInitialContext( )
20: lookup(String)
21: addParameter(long, ParameterData)
22: OjectNotFoundException
23: OjectNotFoundException
or
24: AccessDeniedException
25: AccessDeniedException
or
26: GeneralException
27: GeneralException
No Exception
28: getData(long)
29: setAttribute(String, Object)
Attribute "SoftwareData" contains the object of the SoftwareData

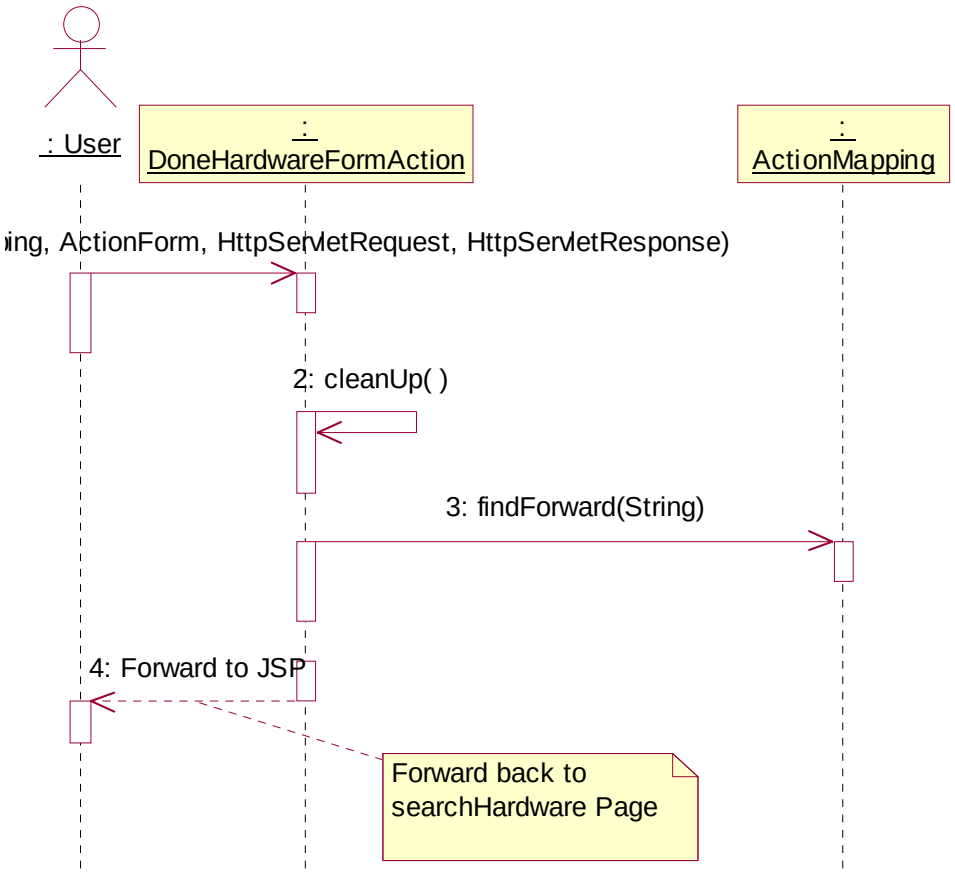
147. BuildRemoveHardwareFormAction.performAction



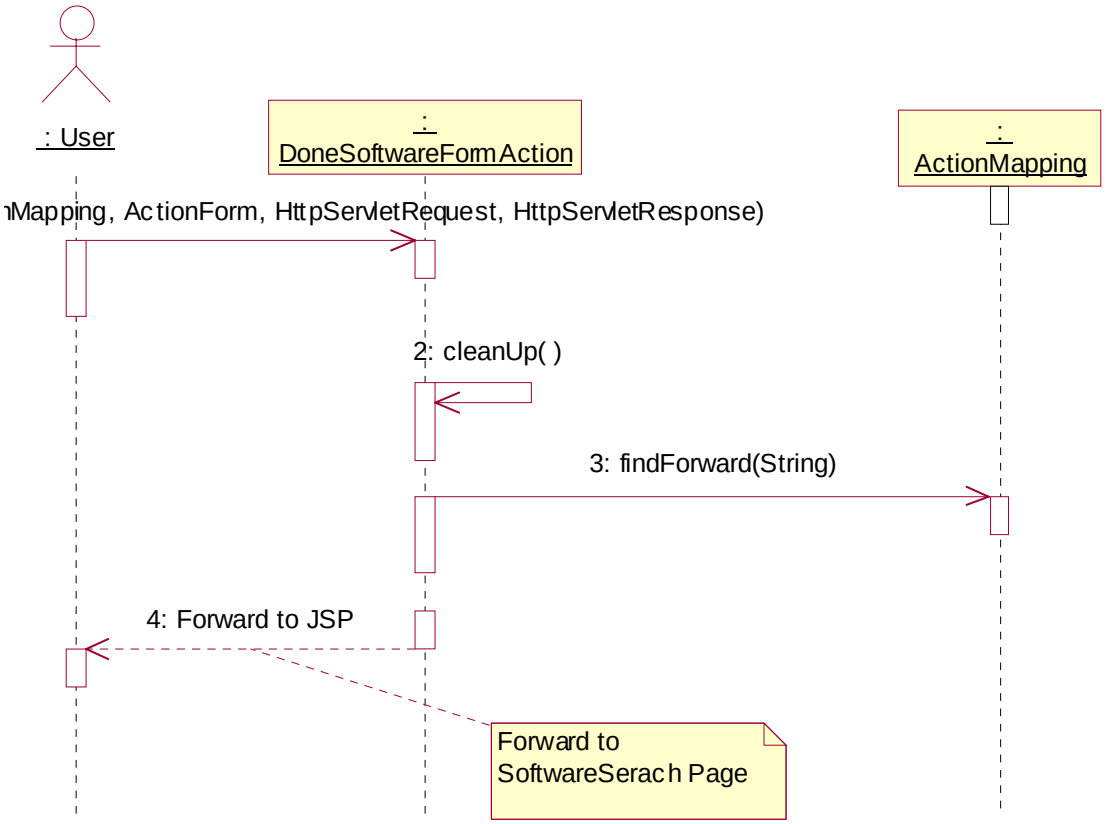
148. BuildRemoveSoftwareFormAction.peroformAction



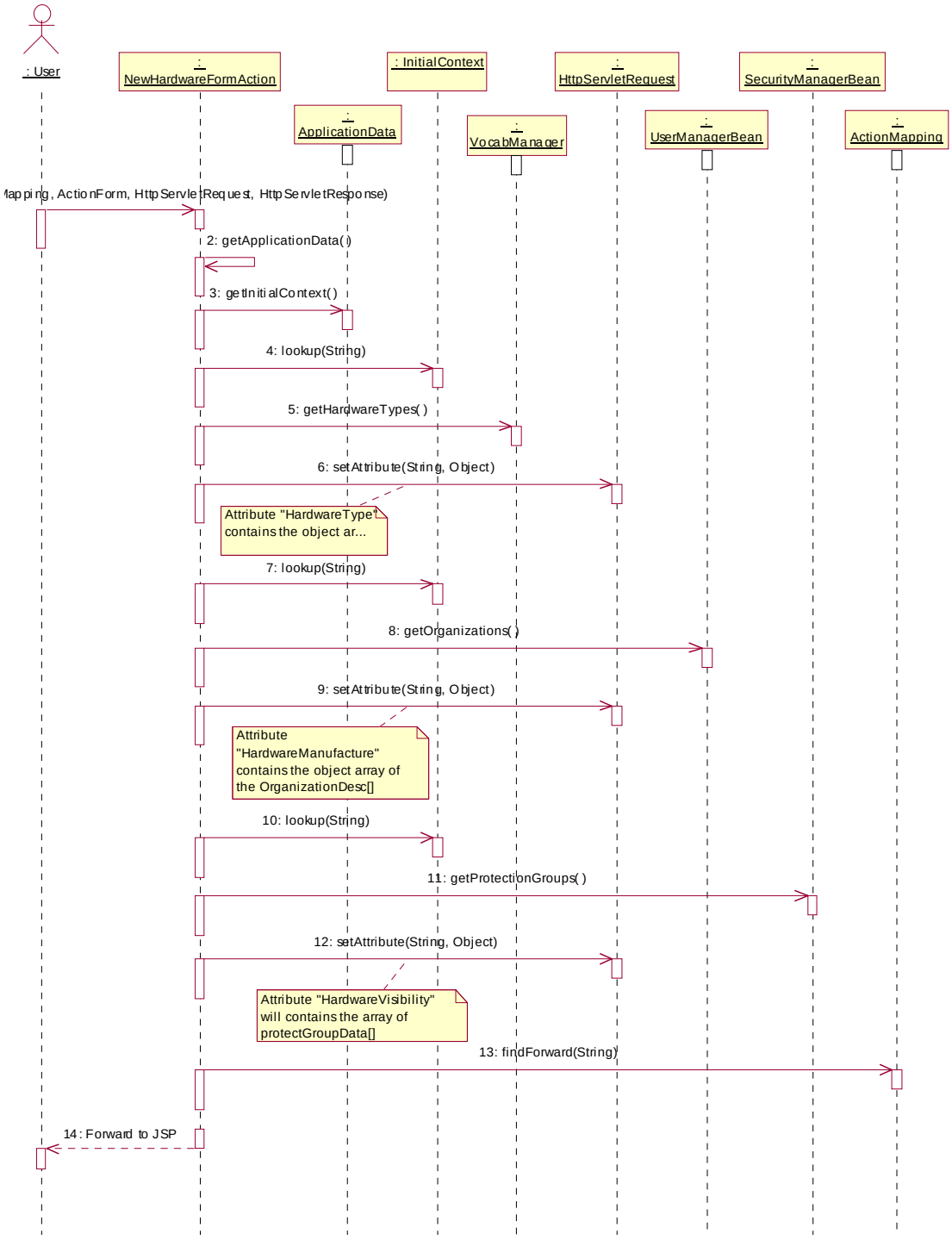
149. BuildDoneHardwareFormAction.performAction



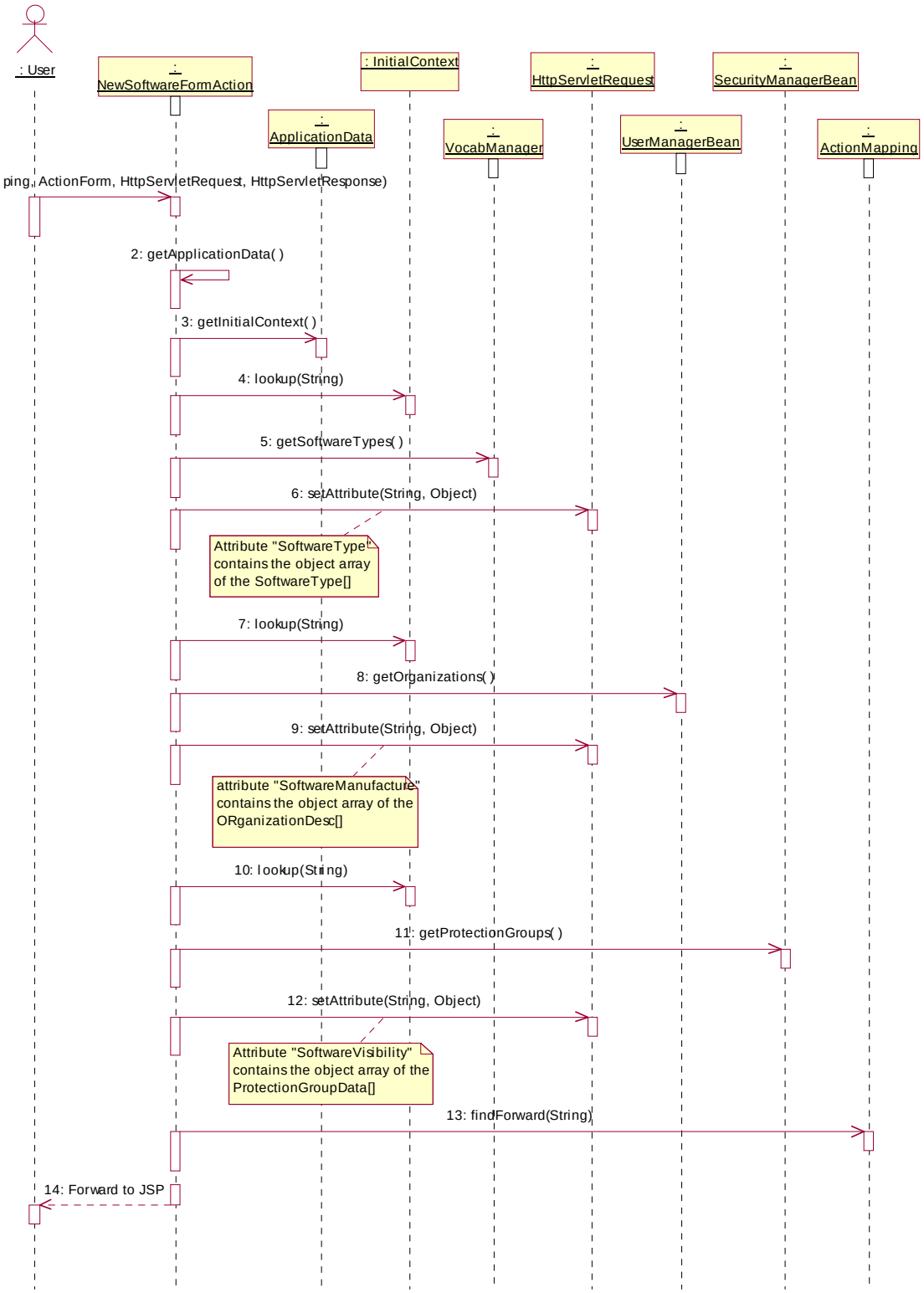
150. BuildDoneSoftwareFormAction



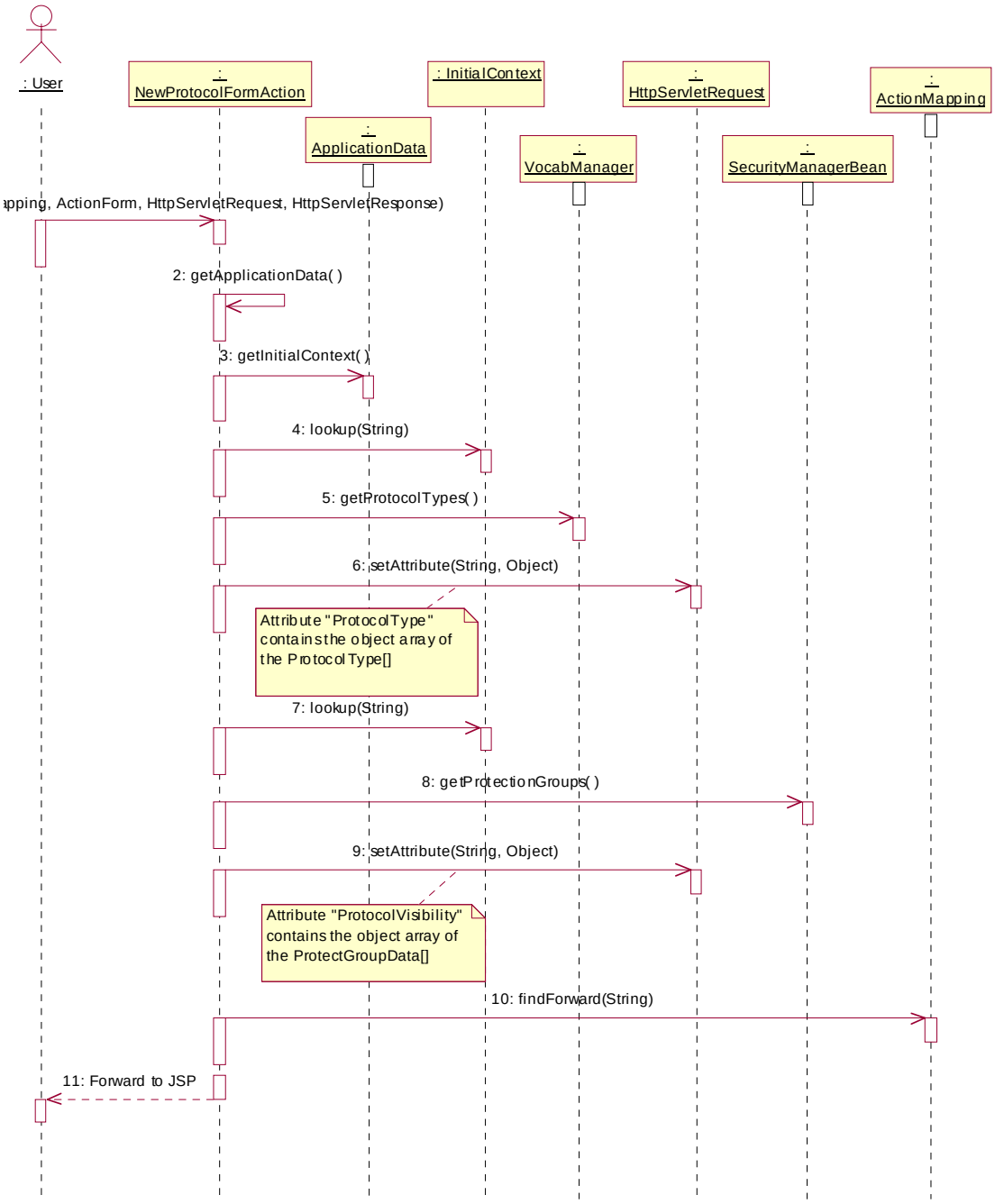
151. BuildNewHardwareFormAction.performAction



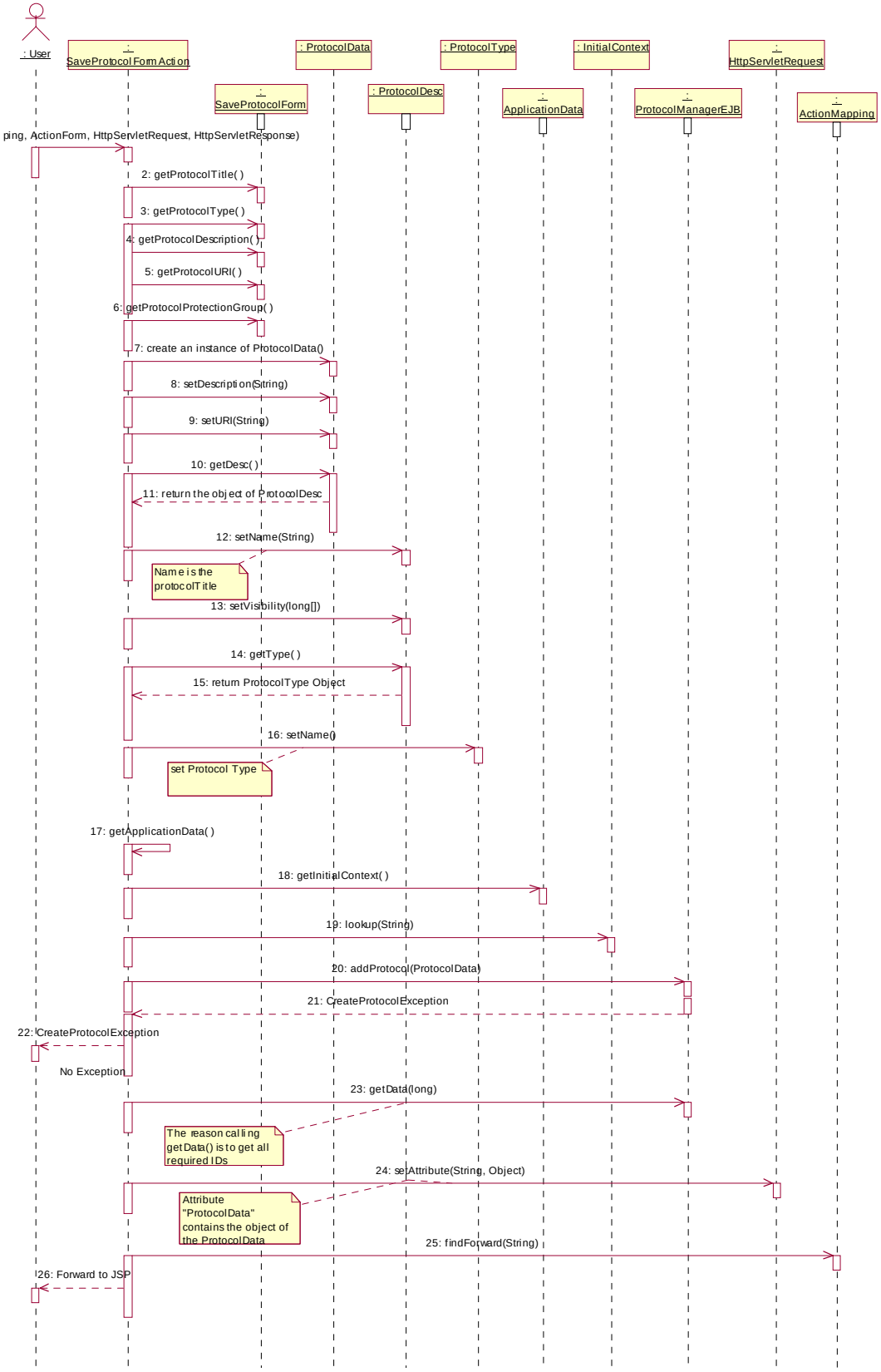
152. BuildNewSoftwareFormAction.performAction


: User
: NewSoftwareFormAction
: InitialContext
: HttpServletRequest
: SecurityManagerBean
: ApplicationData
: VocabManager
: UserManagerBean
: ActionMapping
ping, ActionForm, HttpServletRequest, HttpServletResponse)
2: getApplicationData( )
3: getInitialContext( )
4: lookup(String)
5: getSoftwareTypes( )
6: setAttribute(String, Object)
Attribute "SoftwareType" contains the object array of the SoftwareType[]
7: lookup(String)
8: getOrganizations( )
9: setAttribute(String, Object)
attribute "SoftwareManufacture" contains the object array of the ORganizationDesc[]
10: lookup(String)
11: getProtectionGroups( )
12: setAttribute(String, Object)
Attribute "SoftwareVisibility" contains the object array of the ProtectionGroupData[]
13: findForward(String)
14: Forward to JSP

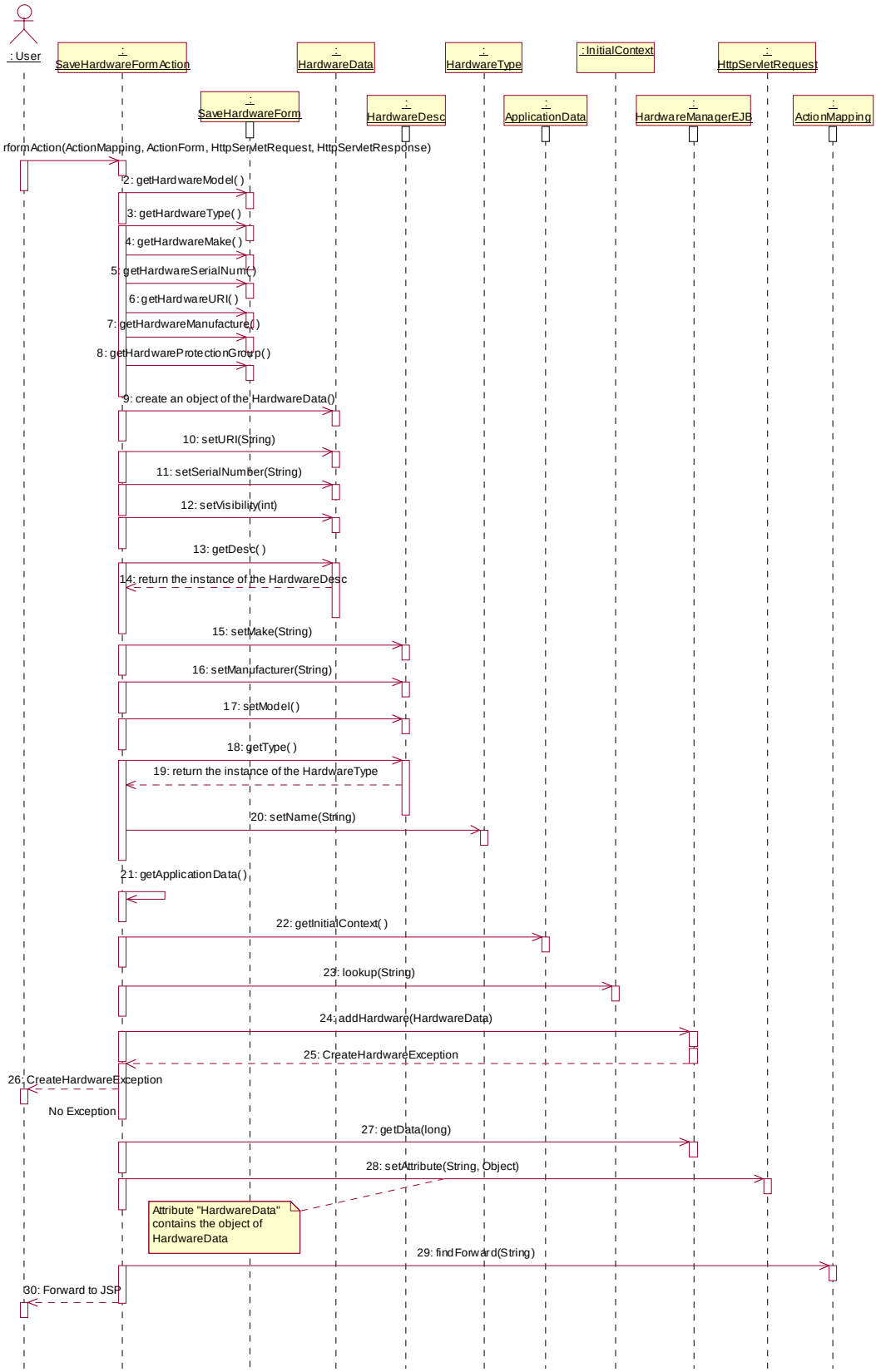
153. BuildNewProtocolFormAction.performAction



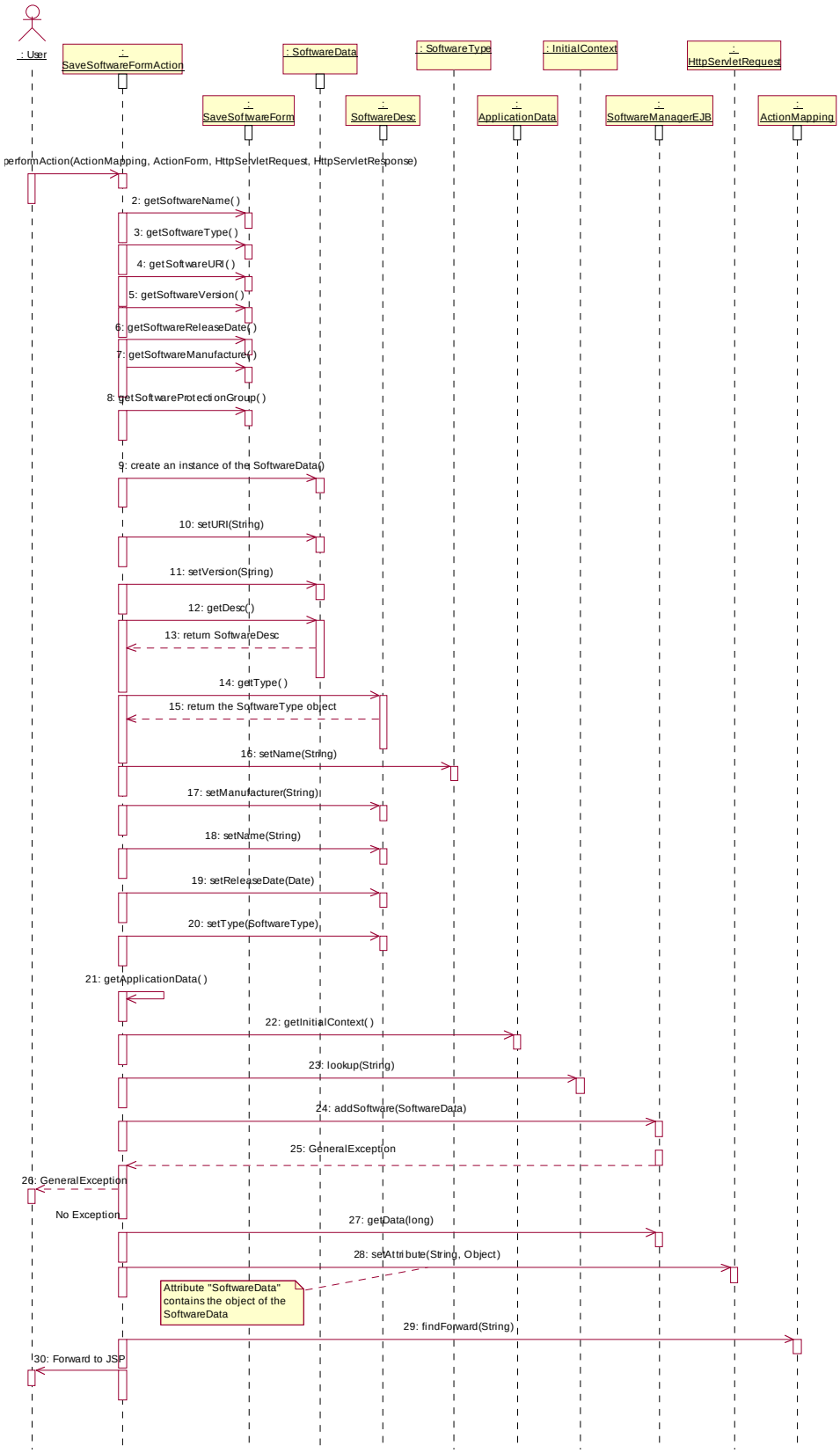
154. BuildSaveProtocolFormAction.performAction


: User
: SaveProtocolFormAction
: ProtocolData
: ProtocolType
: InitialContext
: HttpServletRequest
: SaveProtocolForm
: ProtocolDesc
: ApplicationData
: ProtocolManagerEJB
: ActionMapping
ping, ActionForm, HttpServletRequest, HttpServletResponse)
2: getProtocolTitle( )
3: getProtocolType( )
4: getProtocolDescription( )
5: getProtocolURI( )
6: getProtocolProtectionGroup( )
7: create an instance of ProtocolData()
8: setDescription(String)
9: setURI(String)
10: getDesc( )
11: return the object of ProtocolDesc
12: setName(String)
Name is the protocolTitle
13: setVisibility(long[])
14: getType( )
15: return ProtocolType Object
16: setName()
set Protocol Type
17: getApplicationData( )
18: getInitialContext( )
19: lookup(String)
20: addProtocol(ProtocolData)
21: CreateProtocolException
22: CreateProtocolException
No Exception
23: getData(long)
The reason calling getData() is to get all required IDs
24: setAttribute(String, Object)
Attribute "ProtocolData" contains the object of the ProtocolData
25: findForward(String)
26: Forward to JSP

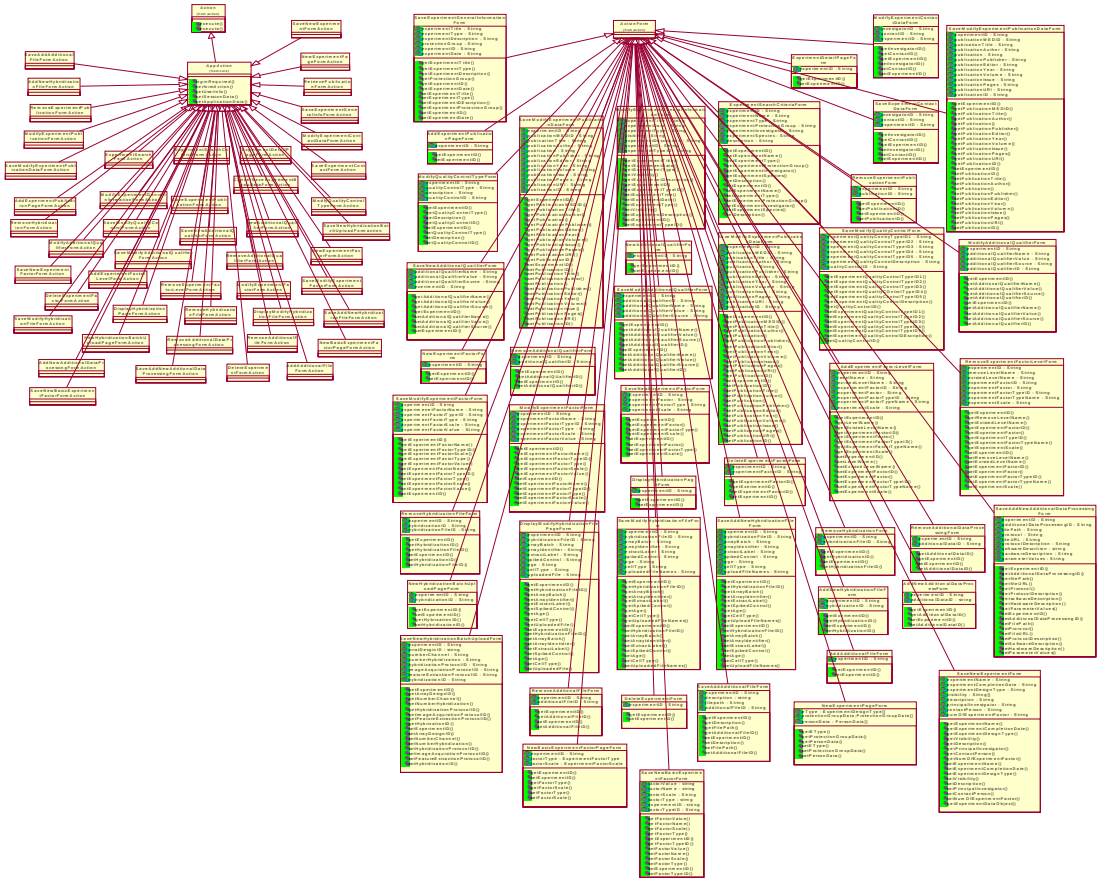
155. BuildSaveHardwareFormAction.performAction


: User
: SaveHardwareFormAction
: HardwareData
: HardwareType
: InitialContext
: HttpServletRequest
: SaveHardwareForm
: HardwareDesc
: ApplicationData
: HardwareManagerEJB
: ActionMapping
rformAction(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse)
2: getHardwareModel( )
3: getHardwareType( )
4: getHardwareMake( )
5: getHardwareSerialNum( )
6: getHardwareURI( )
7: getHardwareManufacture( )
8: getHardwareProtectionGroup( )
9: create an object of the HardwareData()
10: setURI(String)
11: setSerialNumber(String)
12: setVisibility(int)
13: getDesc( )
14: return the instance of the HardwareDesc
15: setMake(String)
16: setManufacturer(String)
17: setModel( )
18: getType( )
19: return the instance of the HardwareType
20: setName(String)
21: getApplicationData( )
22: getInitialContext( )
23: lookup(String)
24: addHardware(HardwareData)
25: CreateHardwareException
26: CreateHardwareException
No Exception
27: getData(long)
28: setAttribute(String, Object)
Attribute "HardwareData" contains the object of HardwareData
29: findForward(String)
30: Forward to JSP

156. BuildSaveSoftwareFormAction.performAction


: User
: SaveSoftwareFormAction
: SoftwareData
: SoftwareType
: InitialContext
: HttpServletRequest
: SaveSoftwareForm
: SoftwareDesc
: ApplicationData
: SoftwareManagerEJB
: ActionMapping
performAction(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse)
2: getSoftwareName( )
3: getSoftwareType( )
4: getSoftwareURI( )
5: getSoftwareVersion( )
6: getSoftwareReleaseDate( )
7: getSoftwareManufacturer( )
8: getSoftwareProtectionGroup( )
9: create an instance of the SoftwareData( )
10: setURI(String)
11: setVersion(String)
12: getDesc( )
13: return SoftwareDesc
14: getType( )
15: return the SoftwareType object
16: setName(String)
17: setManufacturer(String)
18: setName(String)
19: setReleaseDate(Date)
20: setType(SoftwareType)
21: getApplicationData( )
22: getInitialContext( )
23: lookup(String)
24: addSoftware(SoftwareData)
25: GeneralException
26: GeneralException
No Exception
27: getData(long)
28: setAttribute(String, Object)
Attribute "SoftwareData" contains the object of the SoftwareData
29: findForward(String)
30: Forward to JSP

157. Logical View::gov::nih::nci::caarray::ui::experiment



157.1 ExperimentSearchFormAction

157.2 ExperimentSearchCriteriaFormAction

157.3 ExperimentSearchCriteriaForm

157.4 ExperimentDetailPageFormAction

157.5 ExperimentDetailPageForm

157.6 ModifyExperimentGenralInformationFormAction

157.7 ModifyExperimentGeneralInformationForm

157.8 SaveExperimentGeneralInfoFormAction

157.9 SaveExperimentGeneralInformationForm

157.10 CancelSaveExperimentBasicDataFormAction

157.11 ModifyExperimentContactDataFormAction

157.12 ModifyExperimentContactDataForm

157.13 SaveExperimentContactFormAction

157.14 SaveExperimentContactDataForm

157.15 AddExperimentPublicationPageFormAction

157.16 AddExperimentPublicationPageForm

157.17 SaveExperimentPublicationFormAction

157.18 SaveModifyExperimentPublicationDataForm

157.19 RetrievePublicationFormAction

157.20 ModifyExperimentPublicationFormAction

157.21 RemoveExperimentPublicationFormAction

157.22 RemoveExperimentPublicationForm

157.23 SaveModifyExperimentPublicationDataFormAction

157.24 ModfyQualityControlTypeFormAction

157.25 ModifyQualityControlTypeForm

157.26 SaveModifyQualityControlFormAction

157.27 SaveModifyQualityControlForm

157.28 NewAdditionalQualifierFormAction

157.29 NewAdditionalQualifierForm

157.30 SaveNewAdditionalQualifierFormAction

157.31 SaveNewAdditionalQualifierForm

157.32 ModifyAdditionalQualifierFormAction

157.33 ModifyAdditionalQualifierForm

157.34 SaveModifyAdditionalQualifierFormAction

157.35 SaveModifyAdditionalQualifierForm

157.36 RemoveAddtionalQualifierFormAction

157.37 RemoveAdditionalQualifierForm

157.38 NewExperimentFactorFormAction

157.39 NewExperimentFactorForm

157.40 SaveNewExperimentFactorFormAction

157.41 SaveNewExperimentFactorForm

157.42 AddExperimentFactorLevelFormAction

157.43 RemoveExperimentFactorLevelFormAction

157.44 AddExperimentFactorLevelForm

157.45 RemoveExperimentFactorLevelForm

157.46 ModifyExperimentFactorFormAction

157.47 ModifyExperimentFactorForm

157.48 SaveModifyExperimentFactorFormAction

157.49 SaveModifyExperimentFactorForm

157.50 DeleteExperimentFactorFormAction

157.51 DeleteExperimentFactorForm

157.52 DisplayHybridizationPageFormAction

157.53 DisplayHybridizationPageForm

157.54 RemoveHybridizationFileFormAction

157.55 RemoveHybridizationFileForm

157.56 DisplayModifyHybridizationFileFormAction

157.57 DisplayModifyHybridizationFilePageForm

157.58 SaveModifyHybridizationFileFormAction

157.59 SaveModifyHybridizationFileForm

157.60 SaveAddNewHybridizationFileFormAction

157.61 SaveAddNewHybridizationFileForm

157.62 RemoveHybridizationFormAction

157.63 RemoveHybridizationForm

157.64 NewHybridizationBatchUploadPageFormAction

157.65 NewHybridizationBatchUploadPageForm

157.66 SaveNewHybridizationBatchUploadFormaction

157.67 SaveNewHybridizationBatchUploadForm

157.68 AddNewHybridizationFileFormAction

157.69 AddNewHybridizationFileForm

157.70 RemoveAdditionalDataProcessingFormAction

157.71 RemoveAdditionalDataProcessingForm

157.72 RemvoeAdditionalFileFormAction

157.73 RemoveAddtionalFileForm

157.74 AddNewAdditionalDataProcessingFormAction

157.75 AddNewAdditionalDataProcessForm

157.76 SaveAddNewAddtionalDataProcessingFormAction

157.77 SaveAddNewAdditionalDataProcessingForm

157.78 DeleteExperimentFormAction

157.79 DeleteExperimentForm

157.80 AddAdditionalFileFormAction

157.81 AddAdditionalFileForm

157.82 SaveAddAdditionalFileFormAction

157.83 SaveAddAdditionalFileForm

157.84 NewExperimentPageFormAction

157.85 NewExperimentPageForm

157.86 SaveNewExperimentFormAction

157.87 SaveNewExperimentForm

157.88 NewBasicExperimentFactorPageFormAction

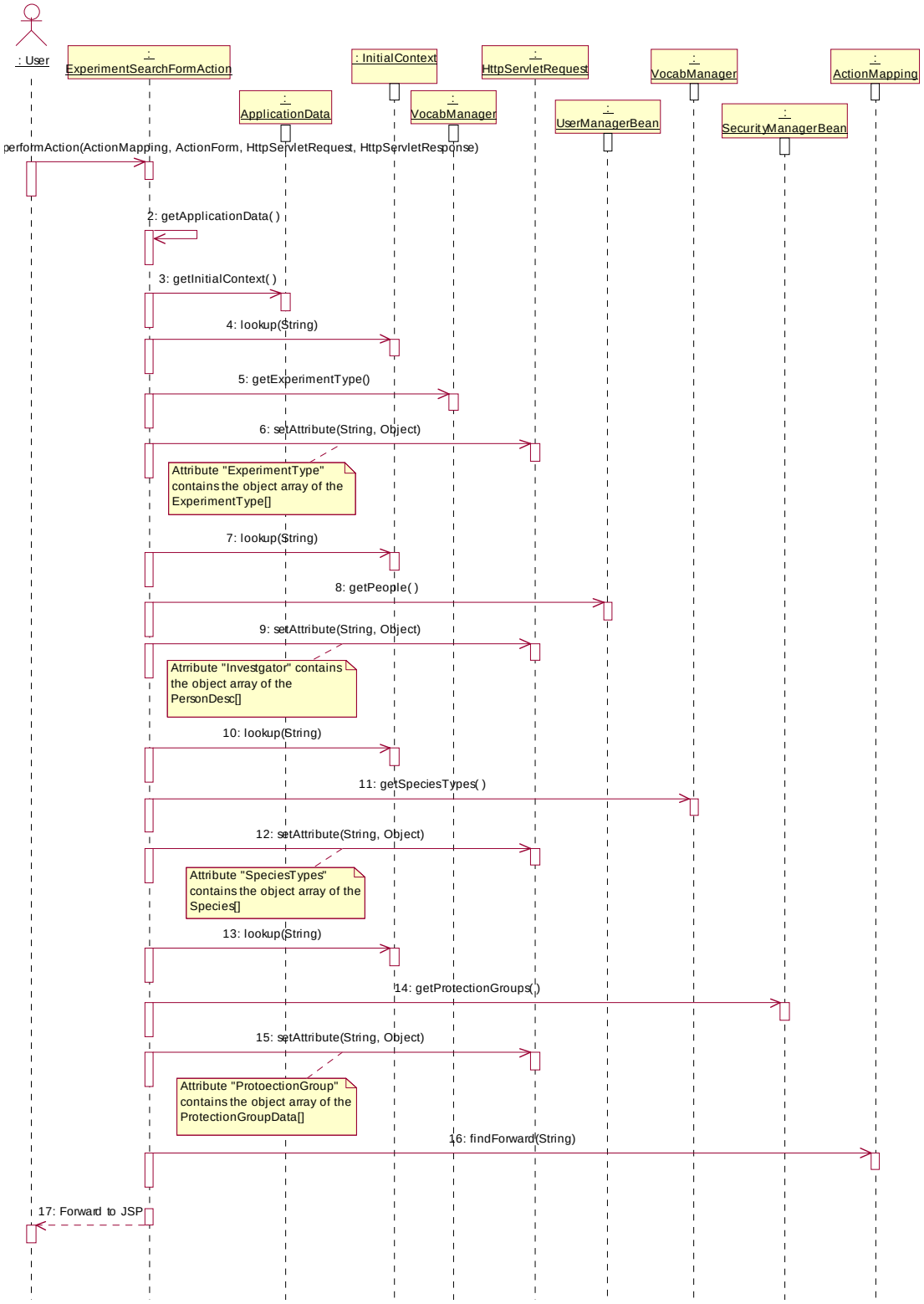
157.89 NewBasicExperimentFactorPageForm

157.90 SaveNewBasicExperimentFactorFormAction

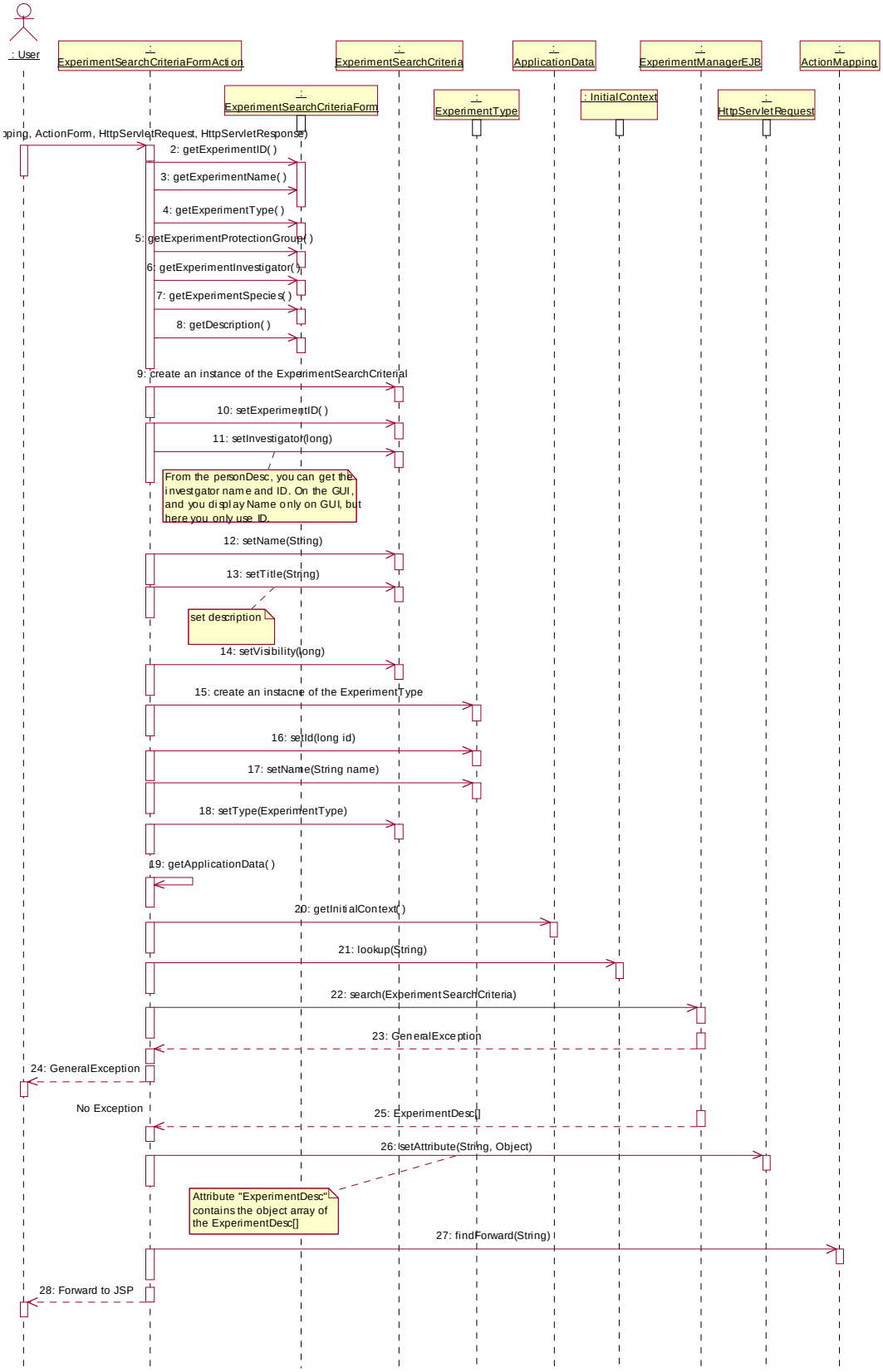
157.91 SaveNewBasicExperimentFactorForm

157.92

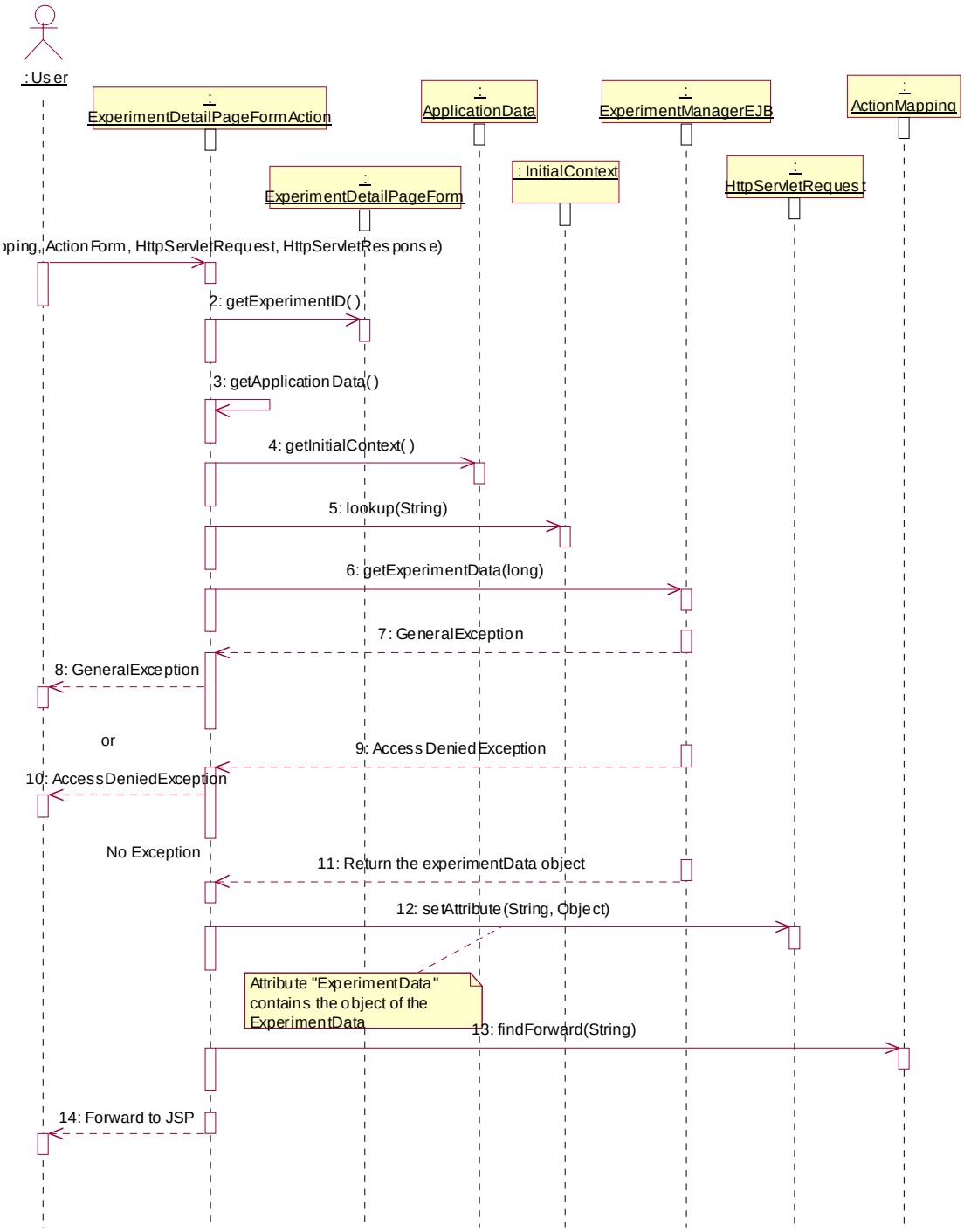
158. BuildExperimentSearchFormAction.performAction


: User
: ExperimentSearchFormAction
: ApplicationData
: InitialContext
: VocabManager
: HttpServletRequest
: UserManagerBean
: VocabManager
: SecurityManagerBean
: ActionMapping
performAction(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse)
2: getApplicationData( )
3: getInitialContext( )
4: lookup(String)
5: getExperimentType()
6: setAttribute(String, Object)
Attribute "ExperimentType" contains the object array of the ExperimentType[]
7: lookup(String)
8: getPeople( )
9: setAttribute(String, Object)
Atrribute "Investgator" contains the object array of the PersonDesc[]
10: lookup(String)
11: getSpeciesTypes( )
12: setAttribute(String, Object)
Attribute "SpeciesTypes" contains the object array of the Species[]
13: lookup(String)
14: getProtectionGroups( )
15: setAttribute(String, Object)
Attribute "ProtoectionGroup" contains the object array of the ProtectionGroupData[]
16: findForward(String)
17: Forward to JSP

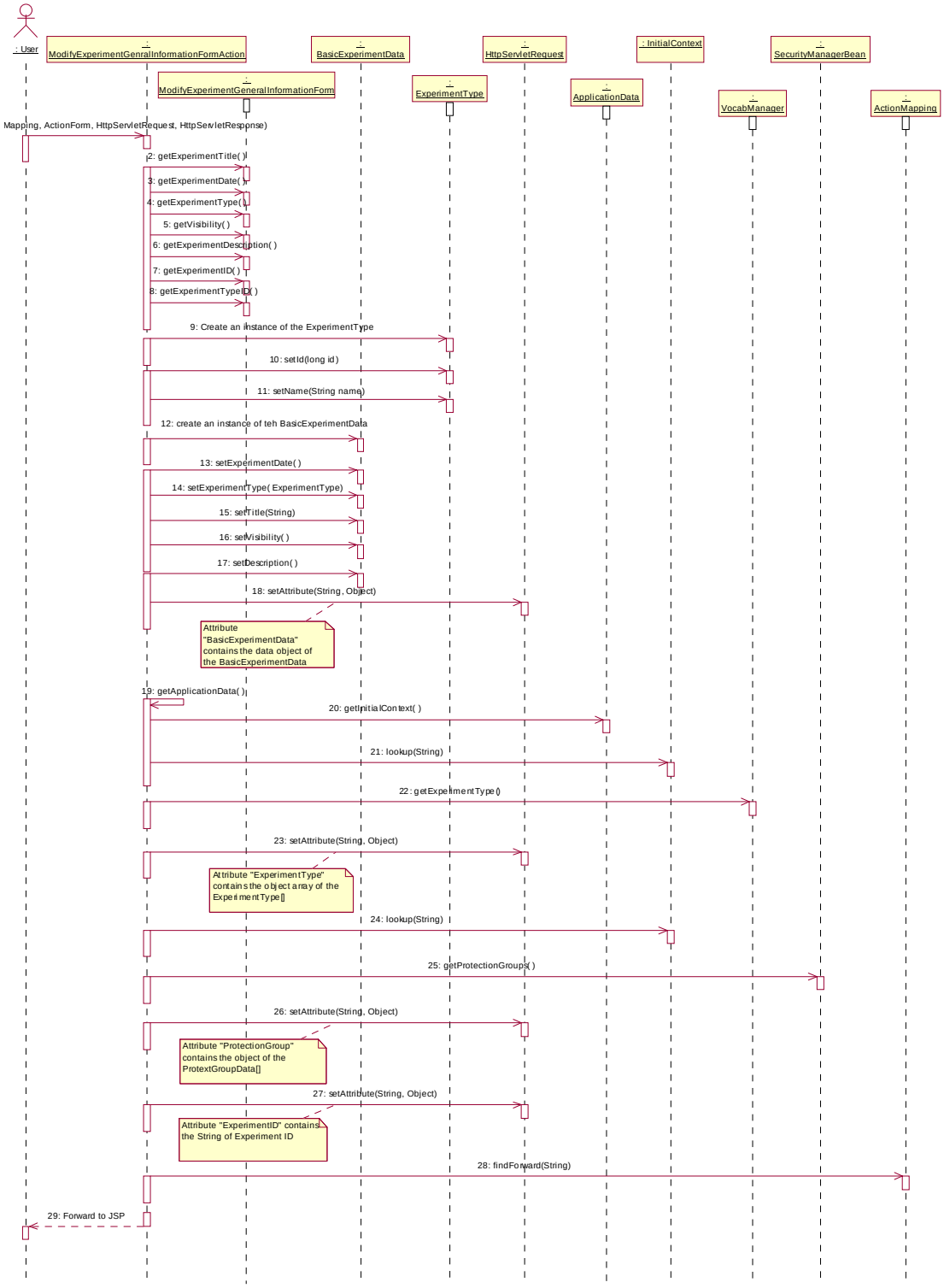
159. BuildExperimentSearchCriterialFormAction.performAction


: User
: ExperimentSearchCriteriaFormAction
: ExperimentSearchCriteria
: ApplicationData
: ExperimentManagerEJB
: ActionMapping
: ExperimentSearchCriteriaForm
: ExperimentType
: InitialContext
: HttpServletRequest
pping, ActionForm, HttpServletRequest, HttpServletResponse)
2: getExperimentID( )
3: getExperimentName( )
4: getExperimentType( )
5: getExperimentProtectionGroup( )
6: getExperimentInvestigator( )
7: getExperimentSpecies( )
8: getDescription( )
9: create an instance of the ExperimentSearchCriteria
10: setExperimentID( )
11: setInvestigator(long)
From the personDesc, you can get the investigator name and ID. On the GUI, and you display Name only on GUI, but here you only use ID.
12: setName(String)
13: setTitle(String)
set description
14: setVisibility(long)
15: create an instacne of the ExperimentType
16: setId(long id)
17: setName(String name)
18: setType(ExperimentType)
19: getApplicationData( )
20: getInitialContext( )
21: lookup(String)
22: search(ExperimentSearchCriteria)
23: GeneralException
24: GeneralException
No Exception
25: ExperimentDesc[]
26: setAttribute(String, Object)
Attribute "ExperimentDesc" contains the object array of the ExperimentDesc[]
27: findForward(String)
28: Forward to JSP

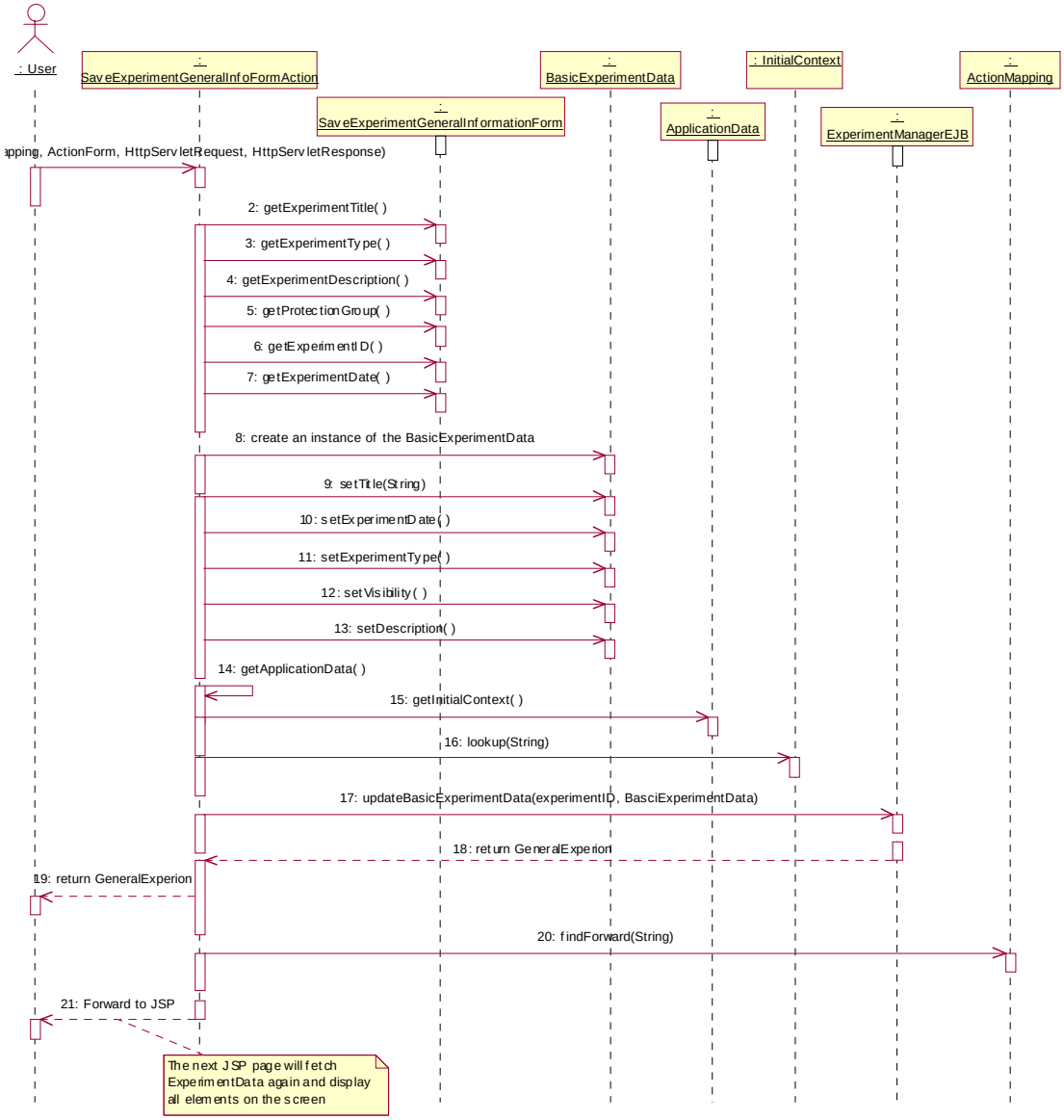
160. BuildExperimentDetailPageFormAction.performAction



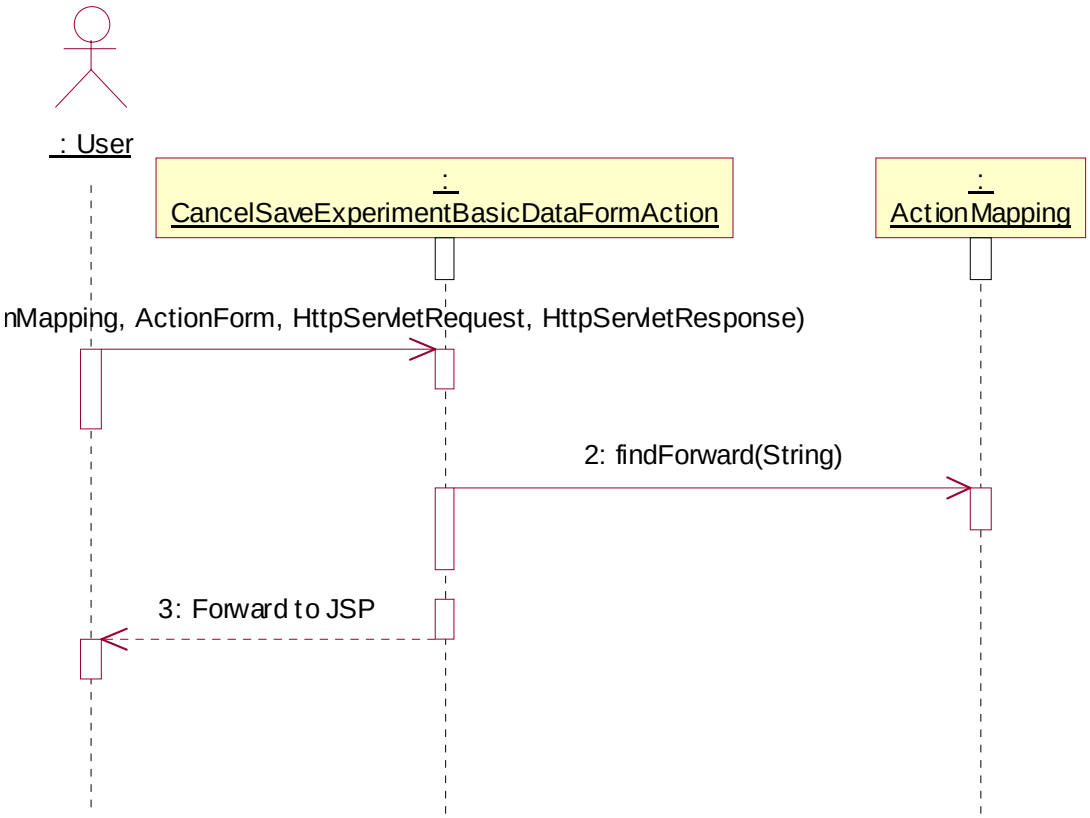
161. BuildExperimentGeneralInfomationPageFormAction.performAction



162. BuildSaveExperimentGeneralInformationFormAction.performAction

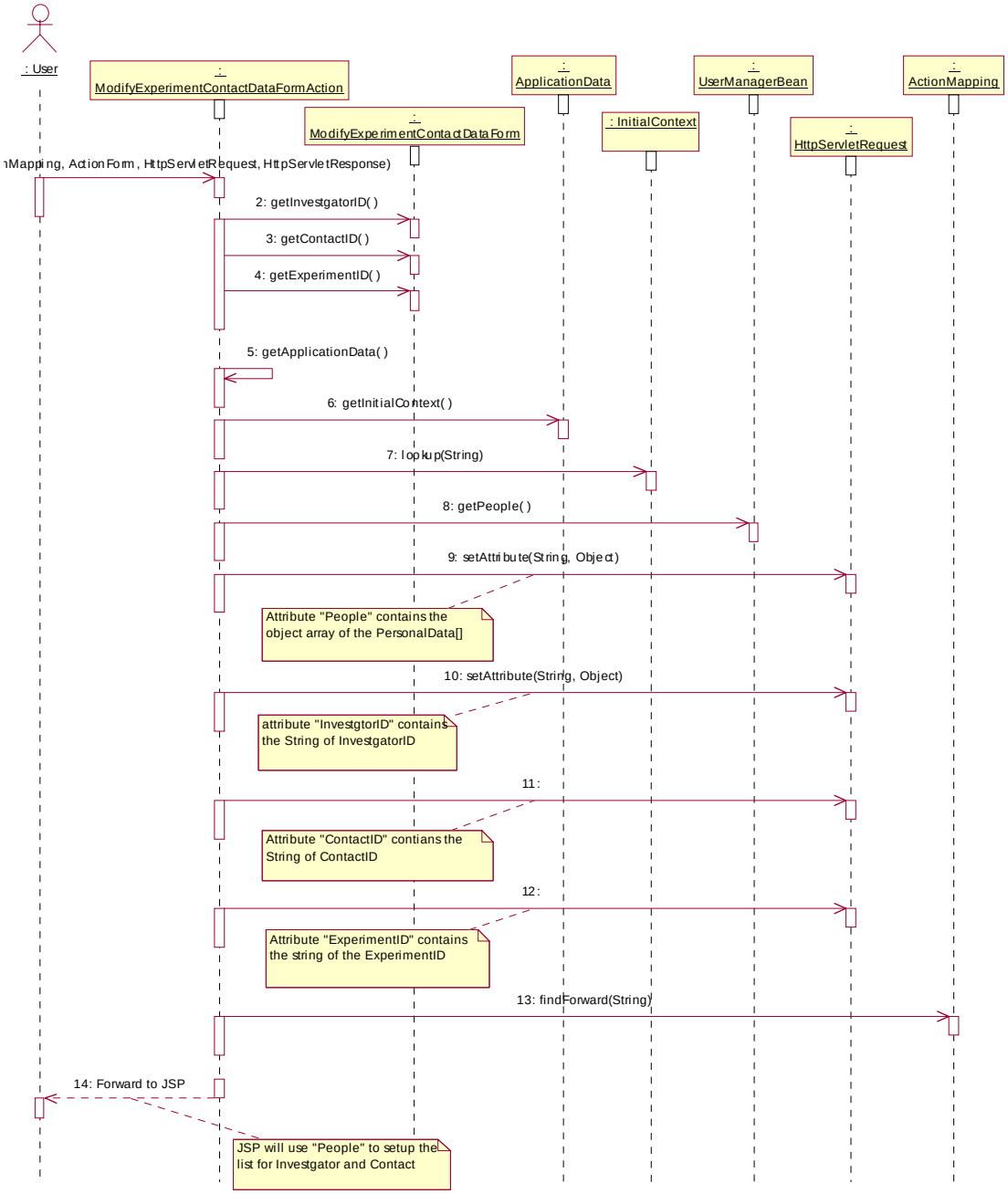


163. BuildCancelSaveBasicExperimentDataFormAction.performAction

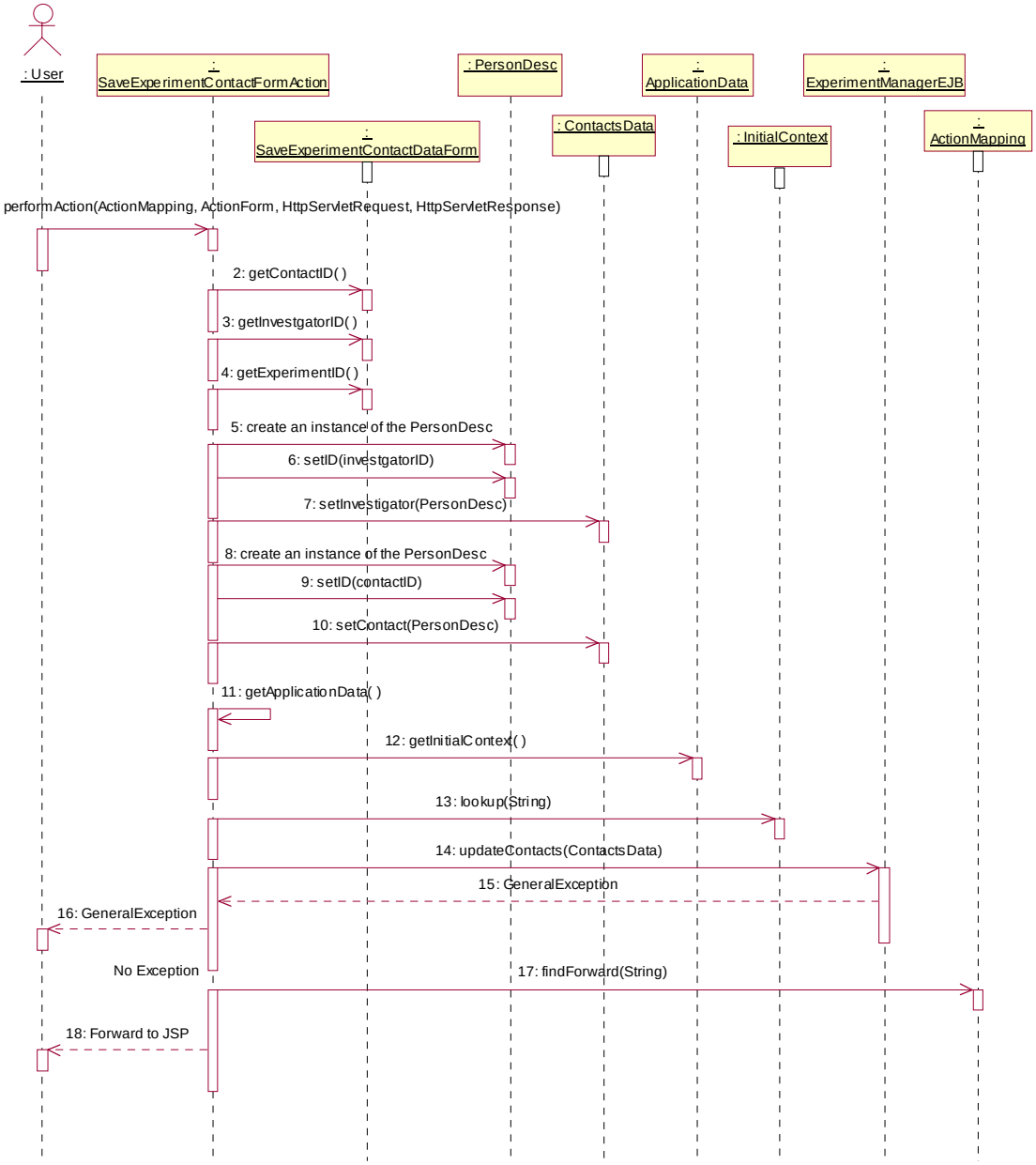


For the rest of Cancellation action, they are just like this one

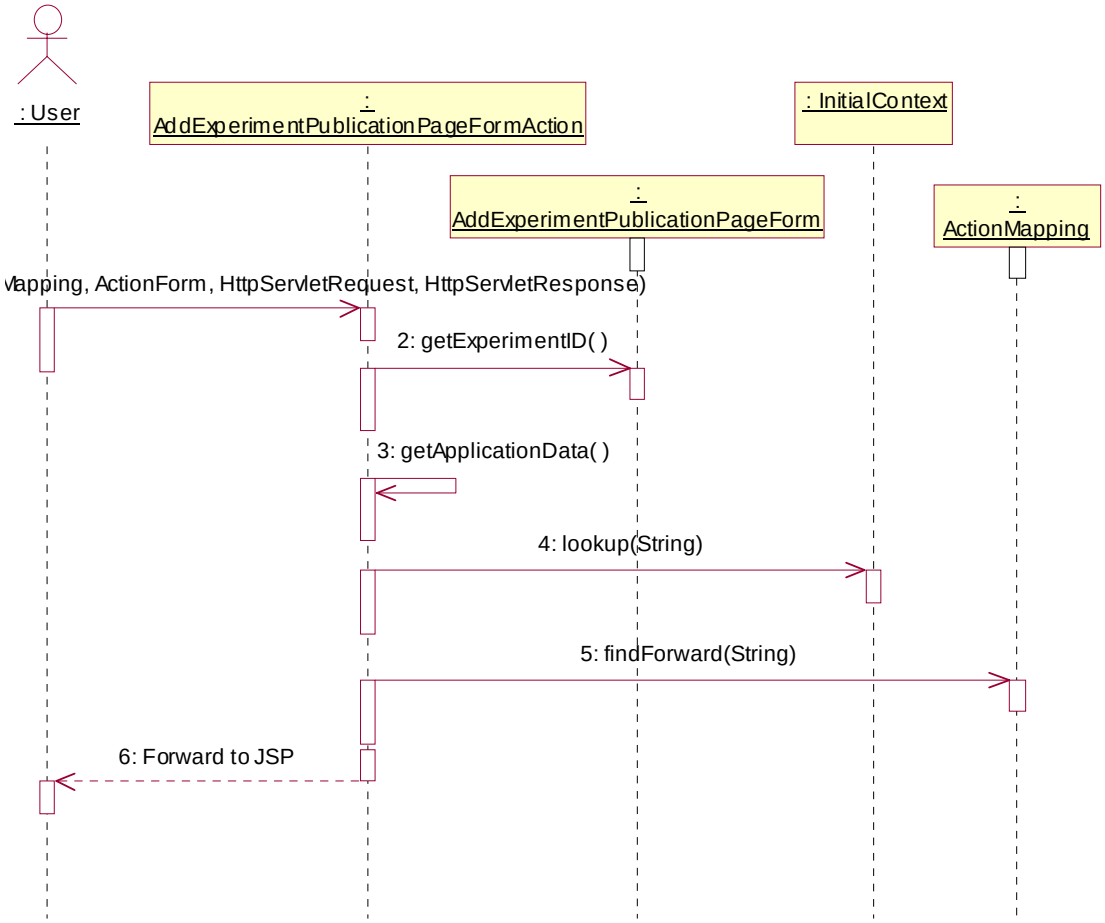
164. BuildExperimentContactDataPageFormAction.performAction



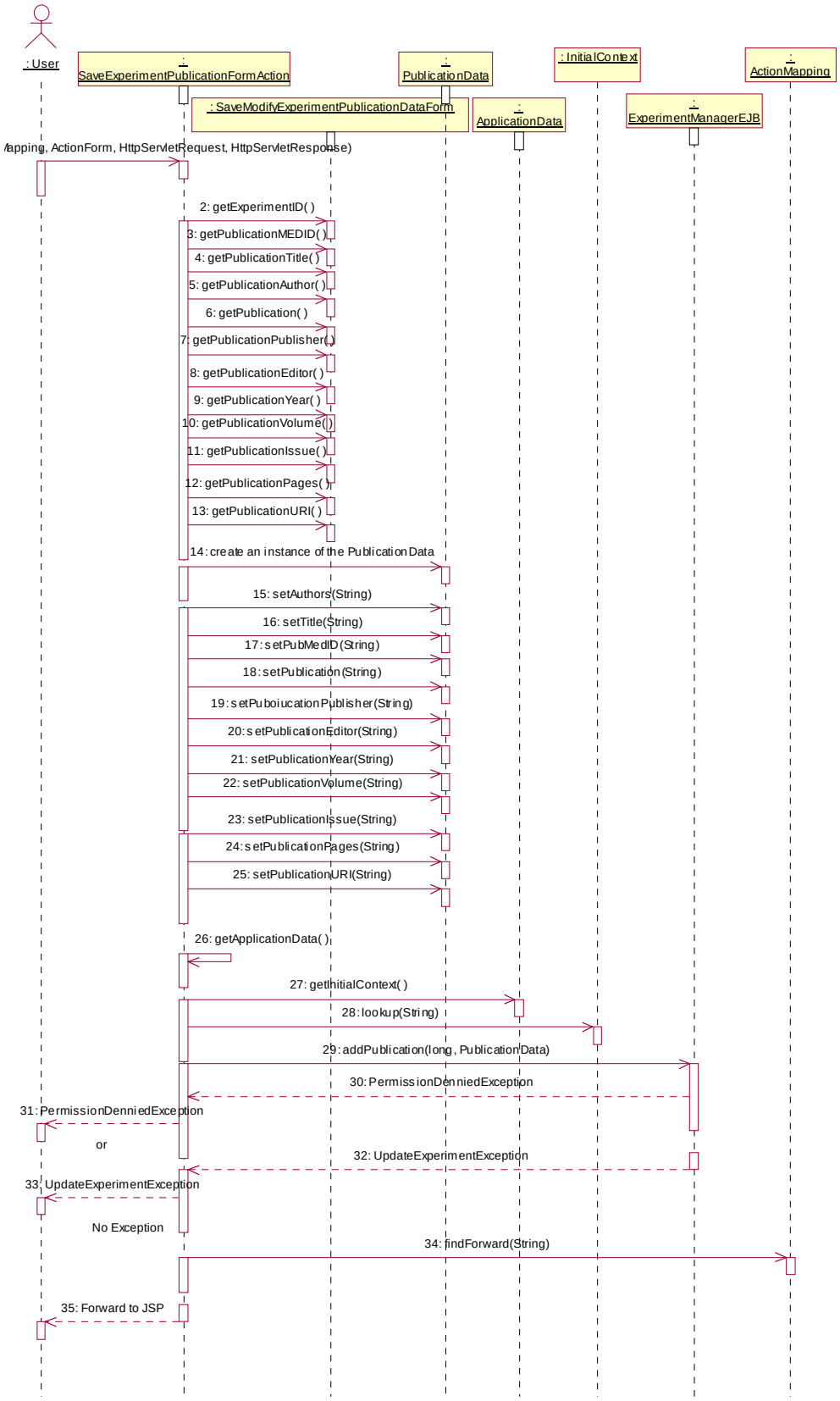
165. BuildSaveExperimentContactFormAction.performAction



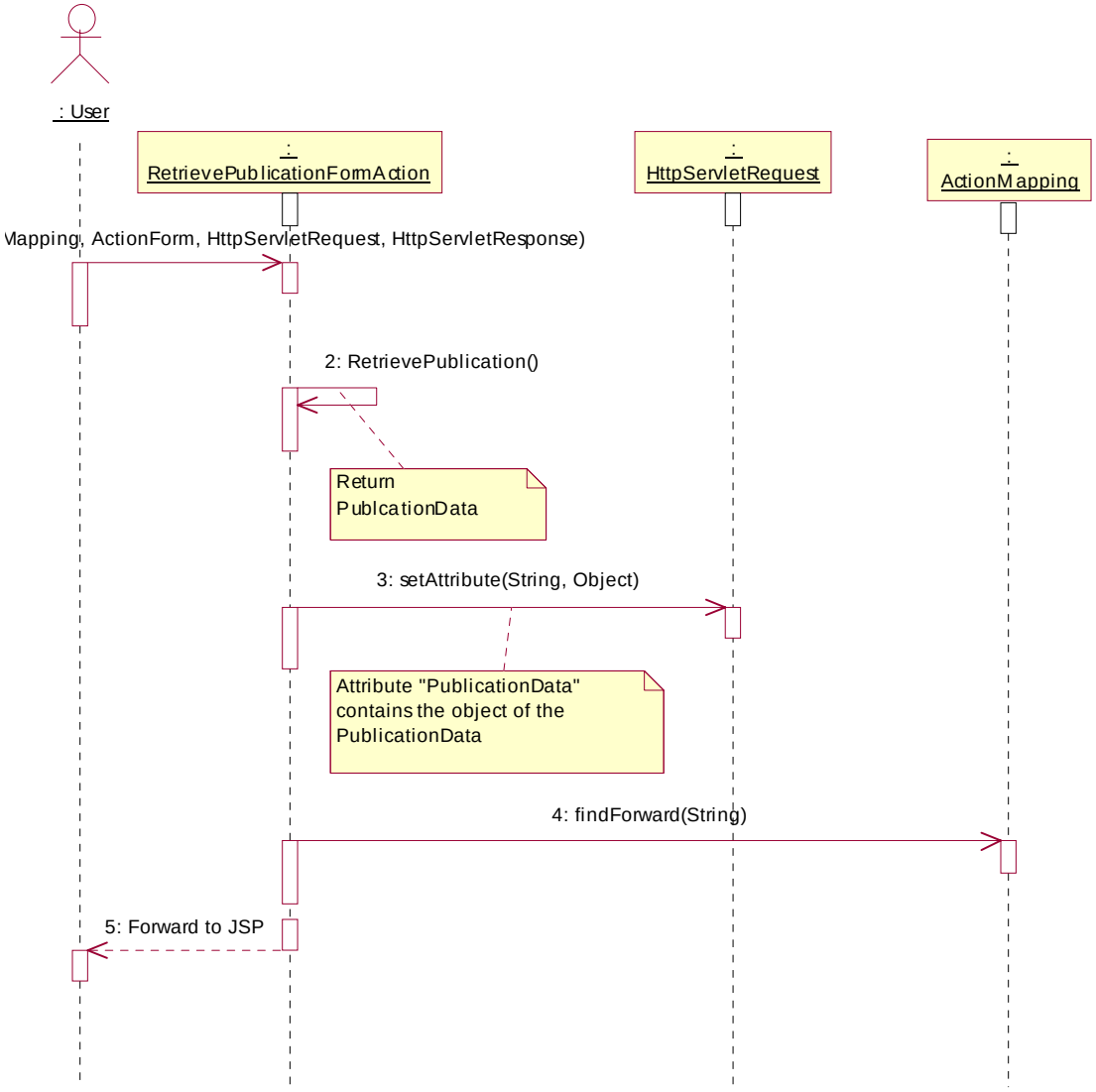
166. BuildAddExperimentPublicationPageFormAction.performAction



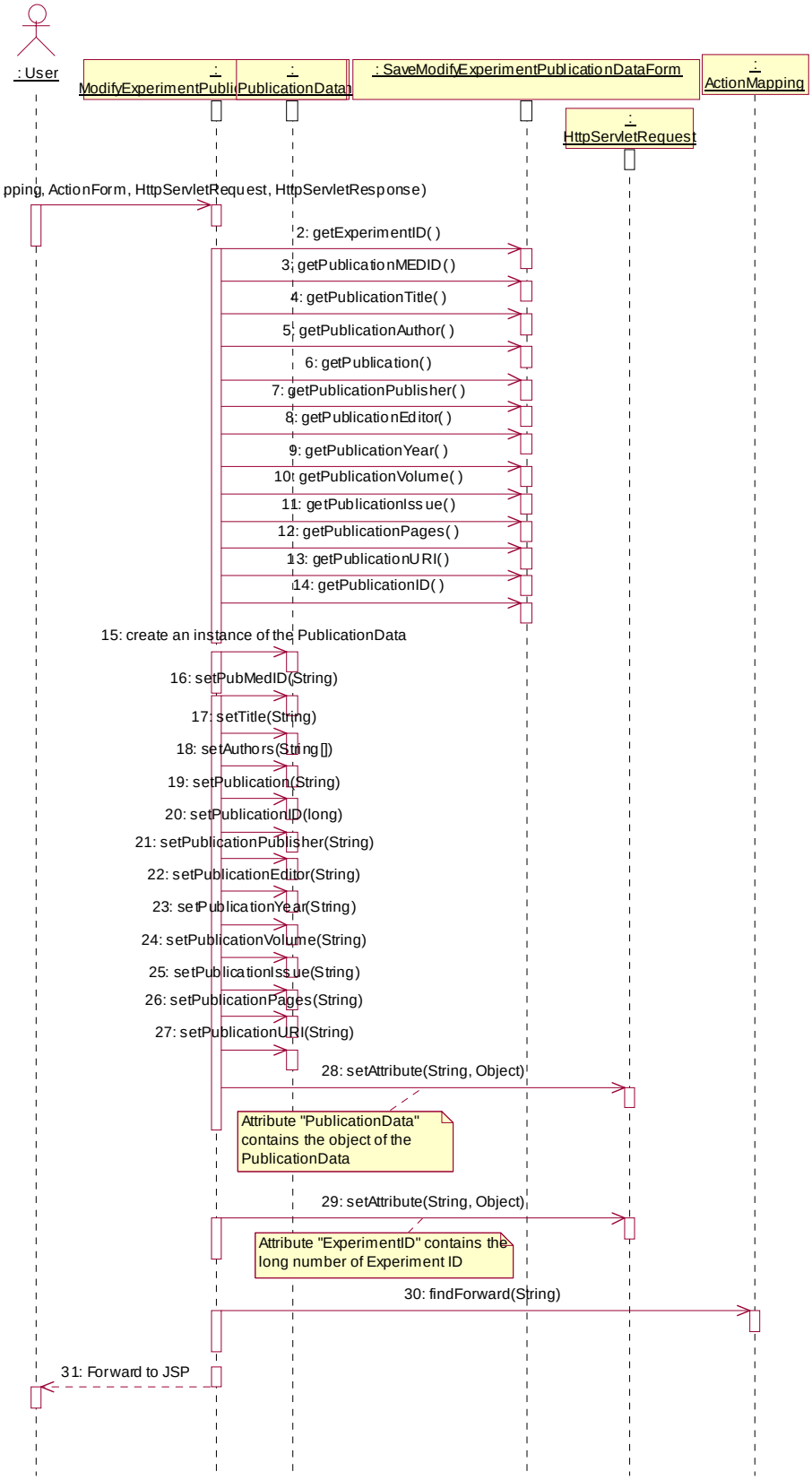
167. BuildSaveExperimentPublicationDataFormAction.performAction


: User
: SaveExperimentPublicationFormAction
: PublicationData
: InitialContext
: ActionMapping
: SaveModifyExperimentPublicationDataForm
: ApplicationData
: ExperimentManagerEJB
/mapping, ActionForm, HttpServletRequest, HttpServletResponse)
2: getExperimentID( )
3: getPublicationMEDID( )
4: getPublicationTitle( )
5: getPublicationAuthor( )
6: getPublication( )
7: getPublicationPublisher( )
8: getPublicationEditor( )
9: getPublicationYear( )
10: getPublicationVolume( )
11: getPublicationIssue( )
12: getPublicationPages( )
13: getPublicationURI( )
14: create an instance of the PublicationData
15: setAuthors(String)
16: setTitle(String)
17: setPubMedID(String)
18: setPublication(String)
19: setPuboiucationPublisher(String)
20: setPublicationEditor(String)
21: setPublicationYear(String)
22: setPublicationVolume(String)
23: setPublicationIssue(String)
24: setPublicationPages(String)
25: setPublicationURI(String)
26: getApplicationData( )
27: getInitialContext( )
28: lookup(String)
29: addPublication(long, PublicationData)
30: PermissionDenniedException
31: PermissionDenniedException
or
32: UpdateExperimentException
33: UpdateExperimentException
No Exception
34: findForward(String)
35: Forward to JSP

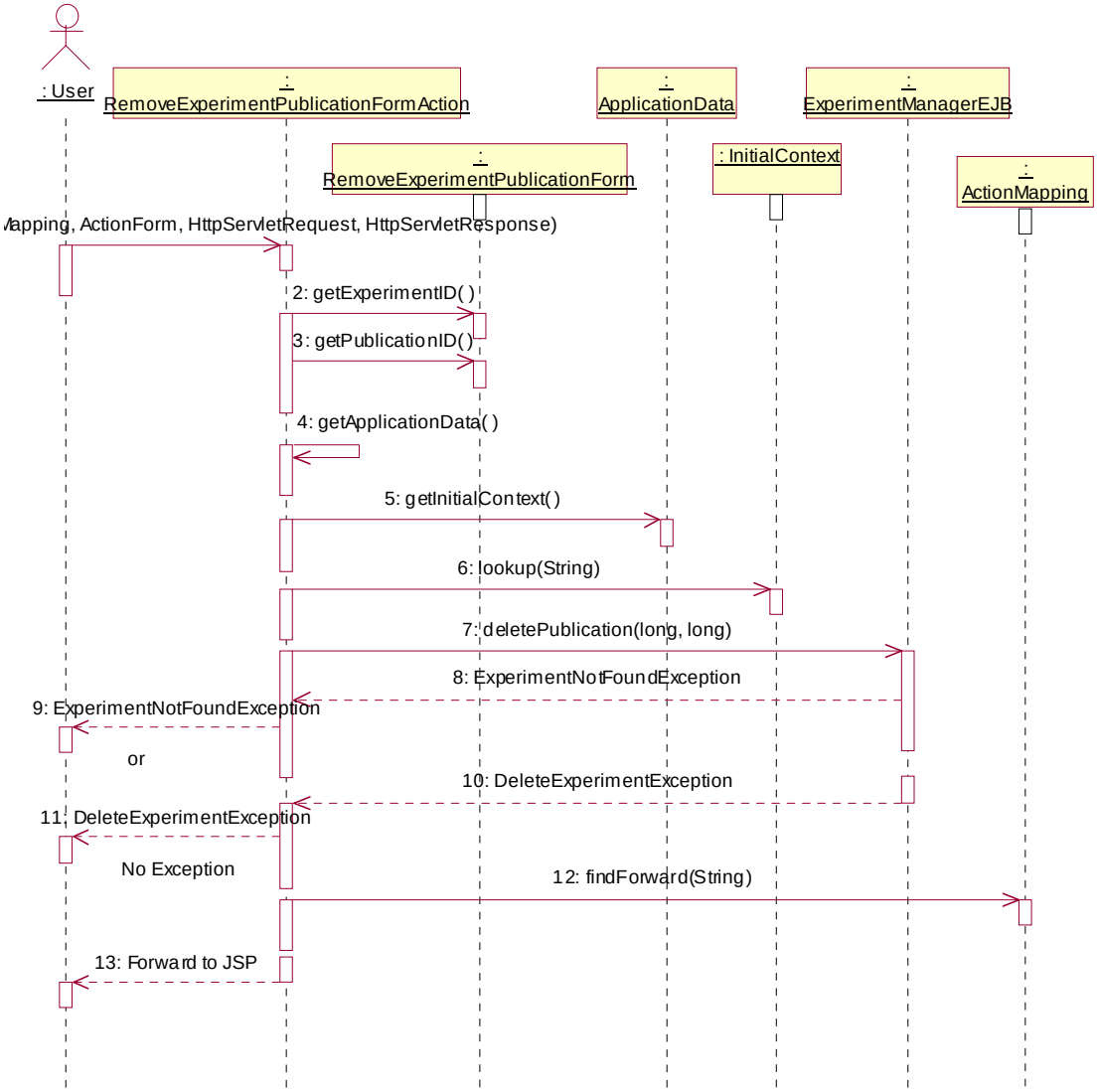
168. BuildRetrievePublicationFormAction.performAction



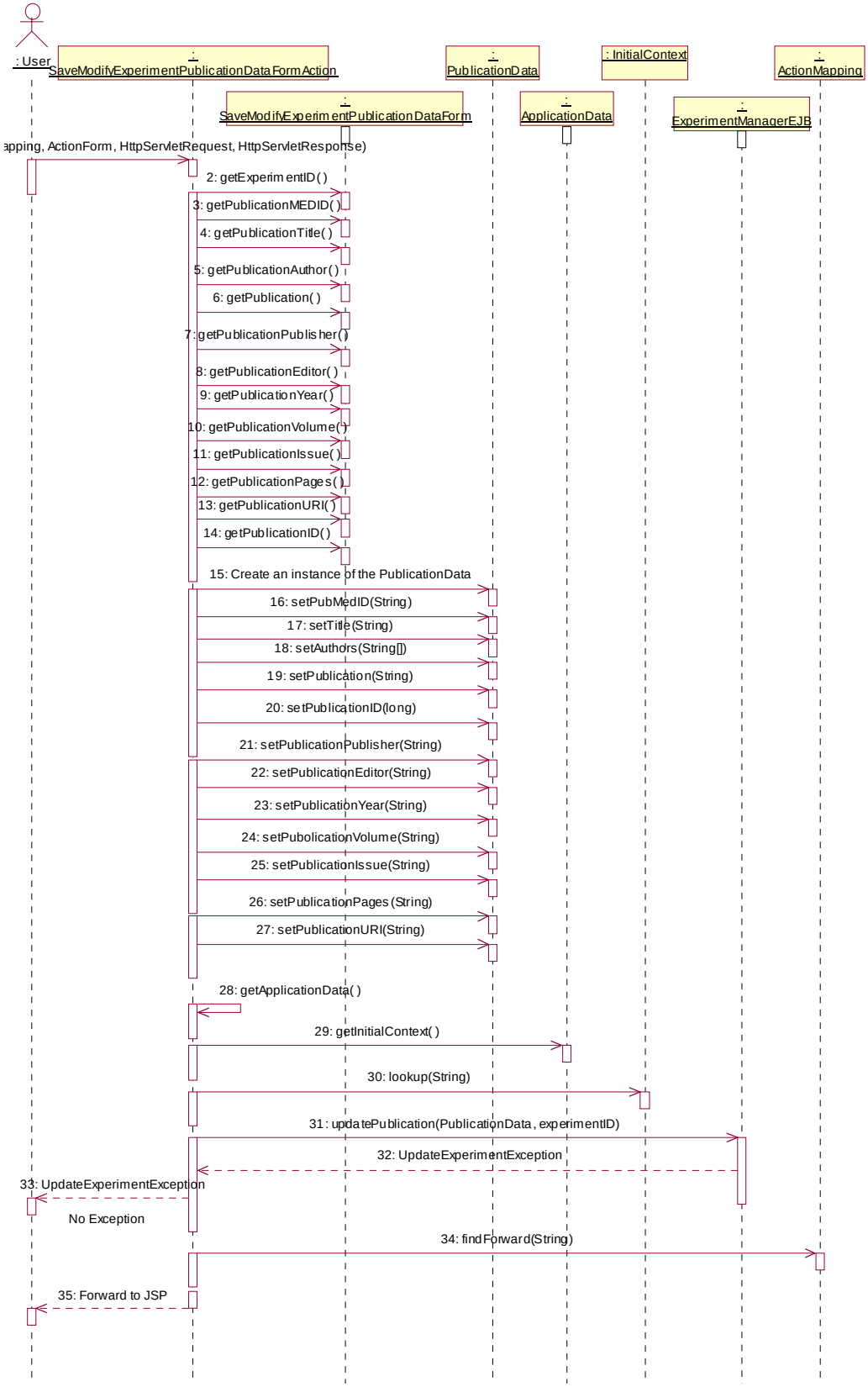
169. BuildModifyExperimentPublicationPageFormAction.perofrmAction


: User
: ModifyExperimentPublicationData
: PublicationData
: SaveModifyExperimentPublicationDataForm
: ActionMapping
: HttpServletRequest
pping, ActionForm, HttpServletRequest, HttpServletResponse)
2: getExperimentID( )
3: getPublicationMEDID( )
4: getPublicationTitle( )
5: getPublicationAuthor( )
6: getPublication( )
7: getPublicationPublisher( )
8: getPublicationEditor( )
9: getPublicationYear( )
10: getPublicationVolume( )
11: getPublicationIssue( )
12: getPublicationPages( )
13: getPublicationURI( )
14: getPublicationID( )
15: create an instance of the PublicationData
16: setPubMedID(String)
17: setTitle(String)
18: setAuthors(String[])
19: setPublication(String)
20: setPublicationID(long)
21: setPublicationPublisher(String)
22: setPublicationEditor(String)
23: setPublicationYear(String)
24: setPublicationVolume(String)
25: setPublicationIssue(String)
26: setPublicationPages(String)
27: setPublicationURI(String)
28: setAttribute(String, Object)
Attribute "PublicationData" contains the object of the PublicationData
29: setAttribute(String, Object)
Attribute "ExperimentID" contains the long number of Experiment ID
30: findForward(String)
31: Forward to JSP

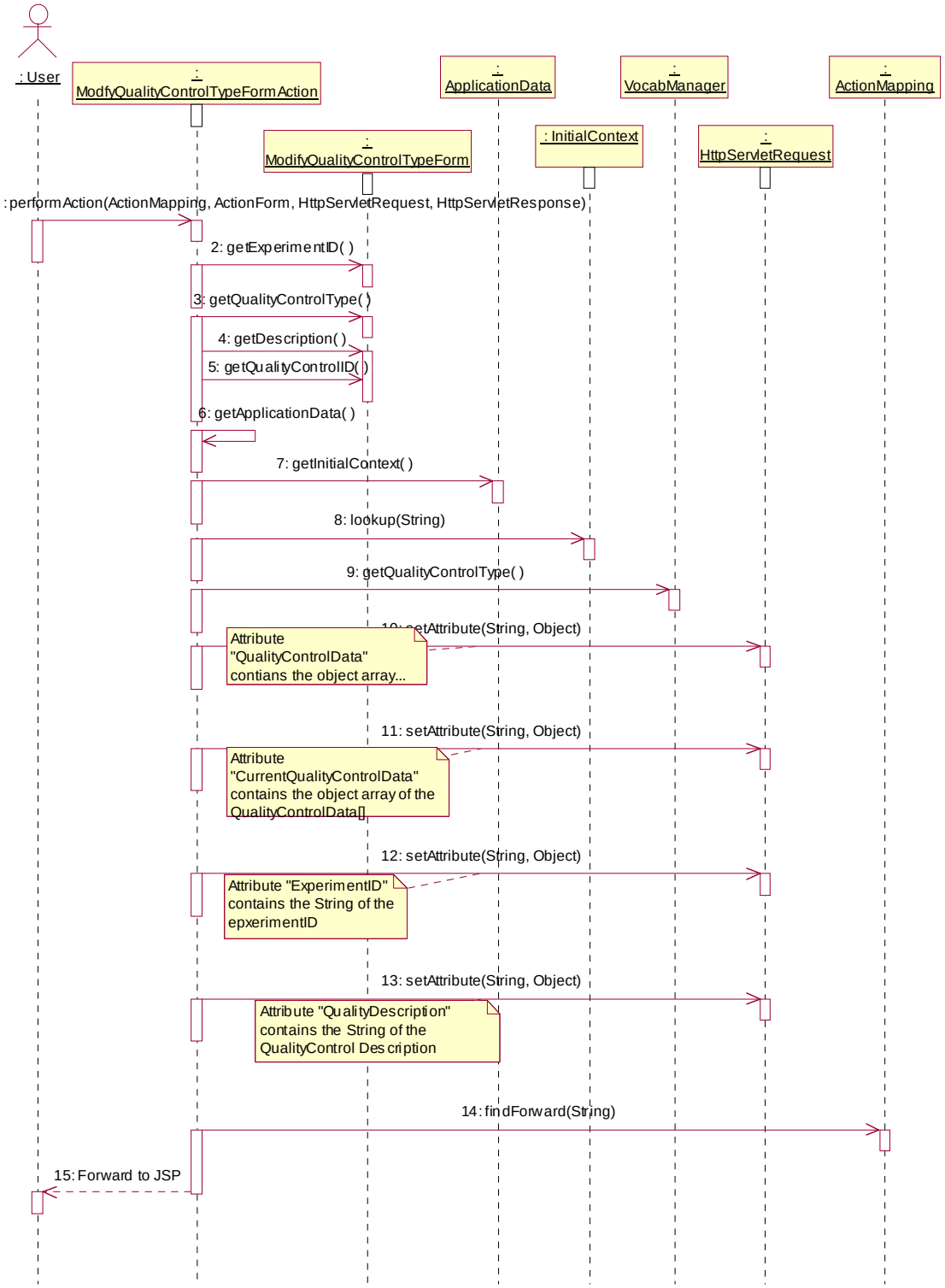
170. BuildRemoveExperimentPublicationFormAction.performAction



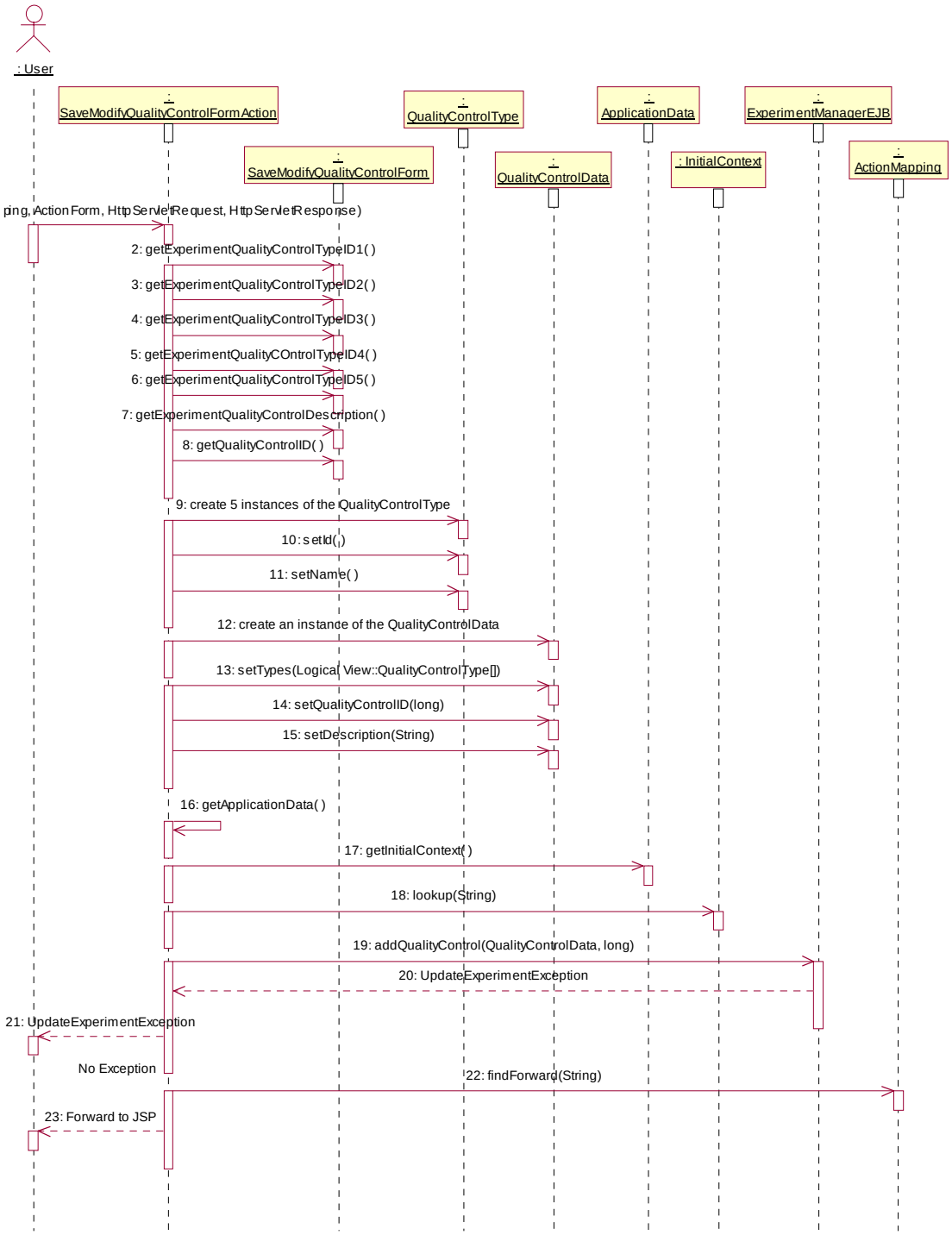
171. BuildSaveModifyExperimentPublicationDataFormAction.performAction


: User
: SaveModifyExperimentPublicationDataFormAction
: PublicationData
: InitialContext
: ActionMapping
: SaveModifyExperimentPublicationDataForm
: ApplicationData
: ExperimentManagerEJB
apping, ActionForm, HttpServletRequest, HttpServletResponse)
2: getExperimentID( )
3: getPublicationMEDID( )
4: getPublicationTitle( )
5: getPublicationAuthor( )
6: getPublication( )
7: getPublicationPublisher( )
8: getPublicationEditor( )
9: getPublicationYear( )
10: getPublicationVolume( )
11: getPublicationIssue( )
12: getPublicationPages( )
13: getPublicationURI( )
14: getPublicationID( )
15: Create an instance of the PublicationData
16: setPubMedID(String)
17: setTitle(String)
18: setAuthors(String[])
19: setPublication(String)
20: setPublicationID(long)
21: setPublicationPublisher(String)
22: setPublicationEditor(String)
23: setPublicationYear(String)
24: setPubolicationVolume(String)
25: setPublicationIssue(String)
26: setPublicationPages(String)
27: setPublicationURI(String)
28: getApplicationData( )
29: getInitialContext( )
30: lookup(String)
31: updatePublication(PublicationData, experimentID)
32: UpdateExperimentException
33: UpdateExperimentException
No Exception
34: findForward(String)
35: Forward to JSP

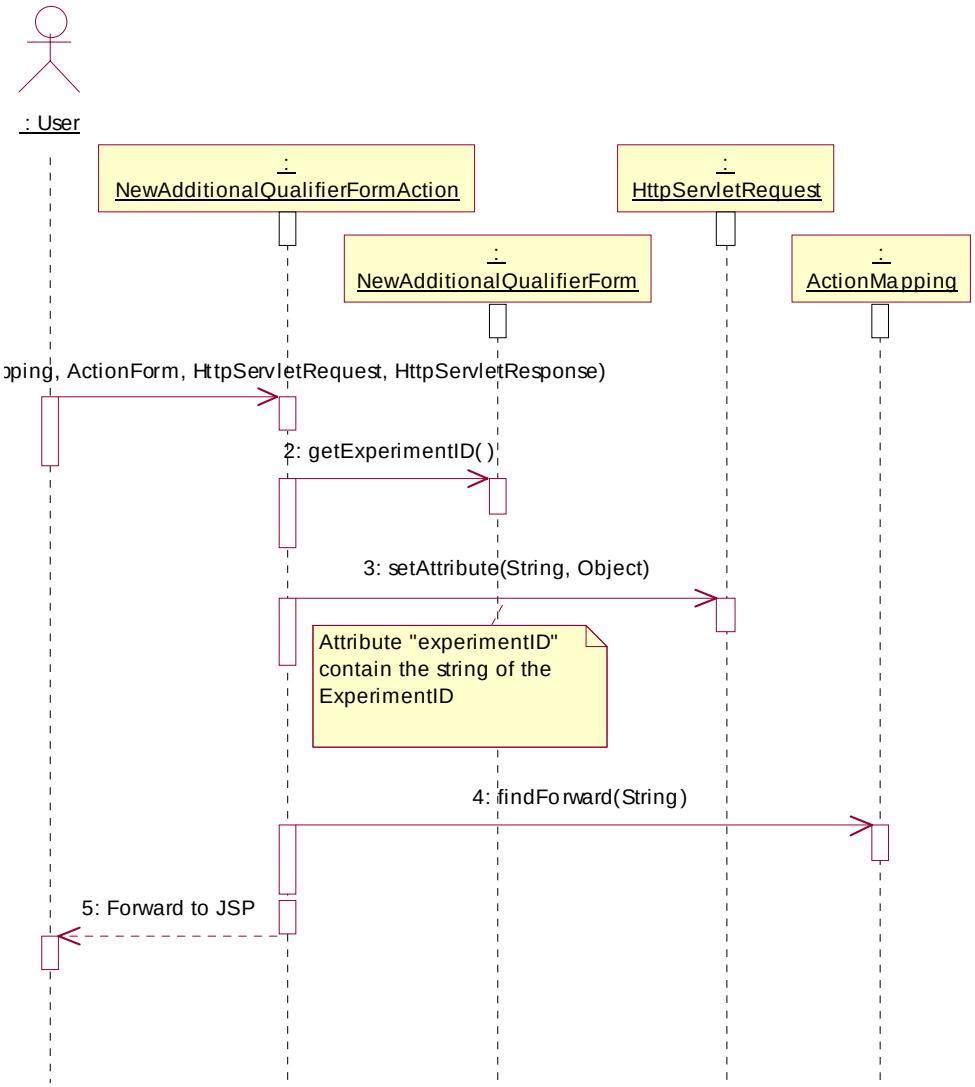
172. BuildModifyQualityControlTypeFormAction.performAction


: User
: ModfyQualityControlTypeFormAction
: ApplicationData
: VocabManager
: ActionMapping
: ModifyQualityControlTypeForm
: InitialContext
: HttpServletRequest
: performAction(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse)
2: getExperimentID( )
3: getQualityControlType( )
4: getDescription( )
5: getQualityControlID( )
6: getApplicationData( )
7: getInitialContext( )
8: lookup(String)
9: getQualityControlType( )
10: setAttribute(String, Object)
Attribute "QualityControlData" contians the object array...
11: setAttribute(String, Object)
Attribute "CurrentQualityControlData" contains the object array of the QualityControlData[]
12: setAttribute(String, Object)
Attribute "ExperimentID" contains the String of the epxerimentID
13: setAttribute(String, Object)
Attribute "QualityDescription" contains the String of the QualityControl Description
14: findForward(String)
15: Forward to JSP

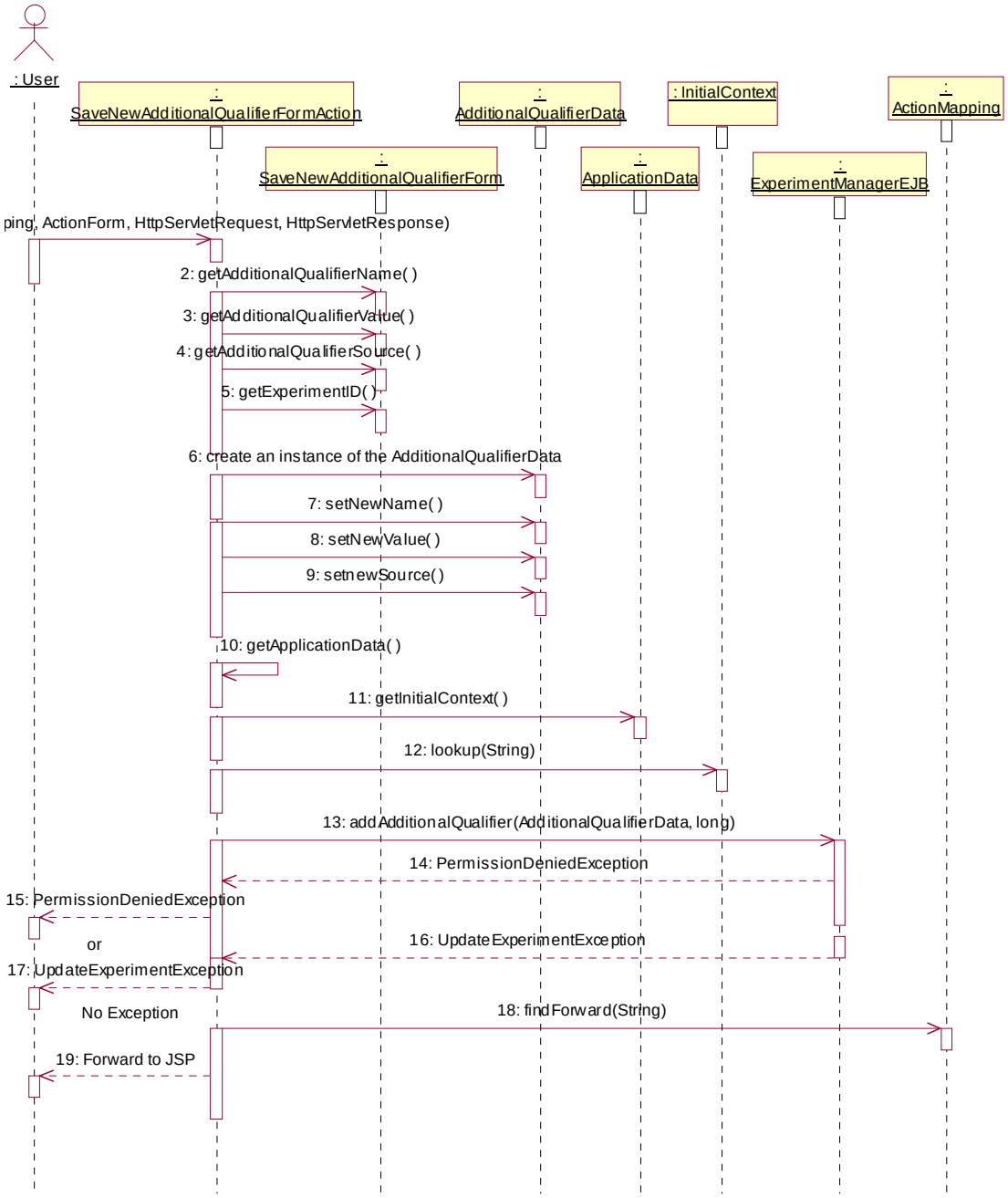
173. BuildSaveModifyQualityControlFormAction.performAction



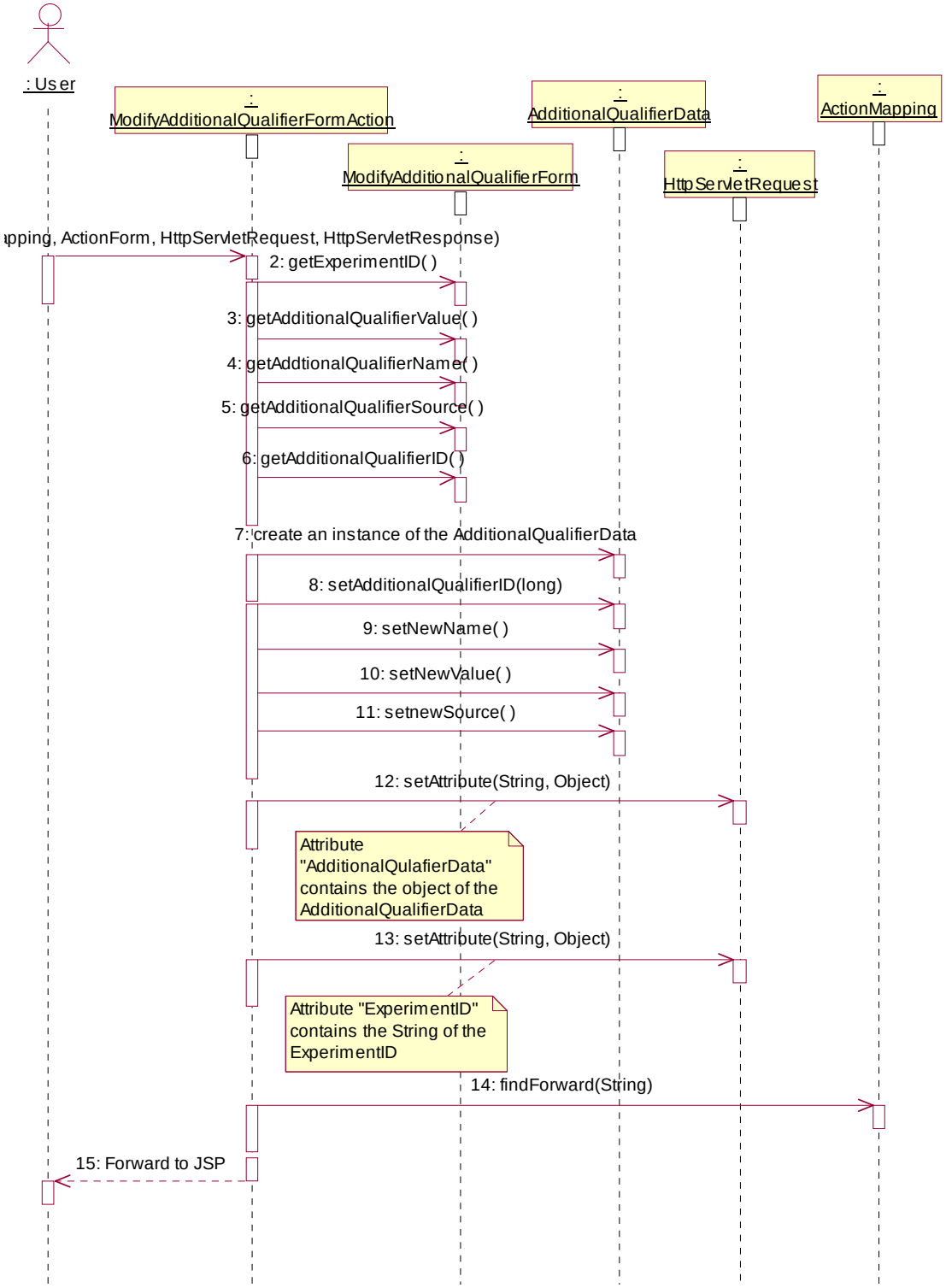
174. BuildNewAdditionalQulifierFormAction.performAction



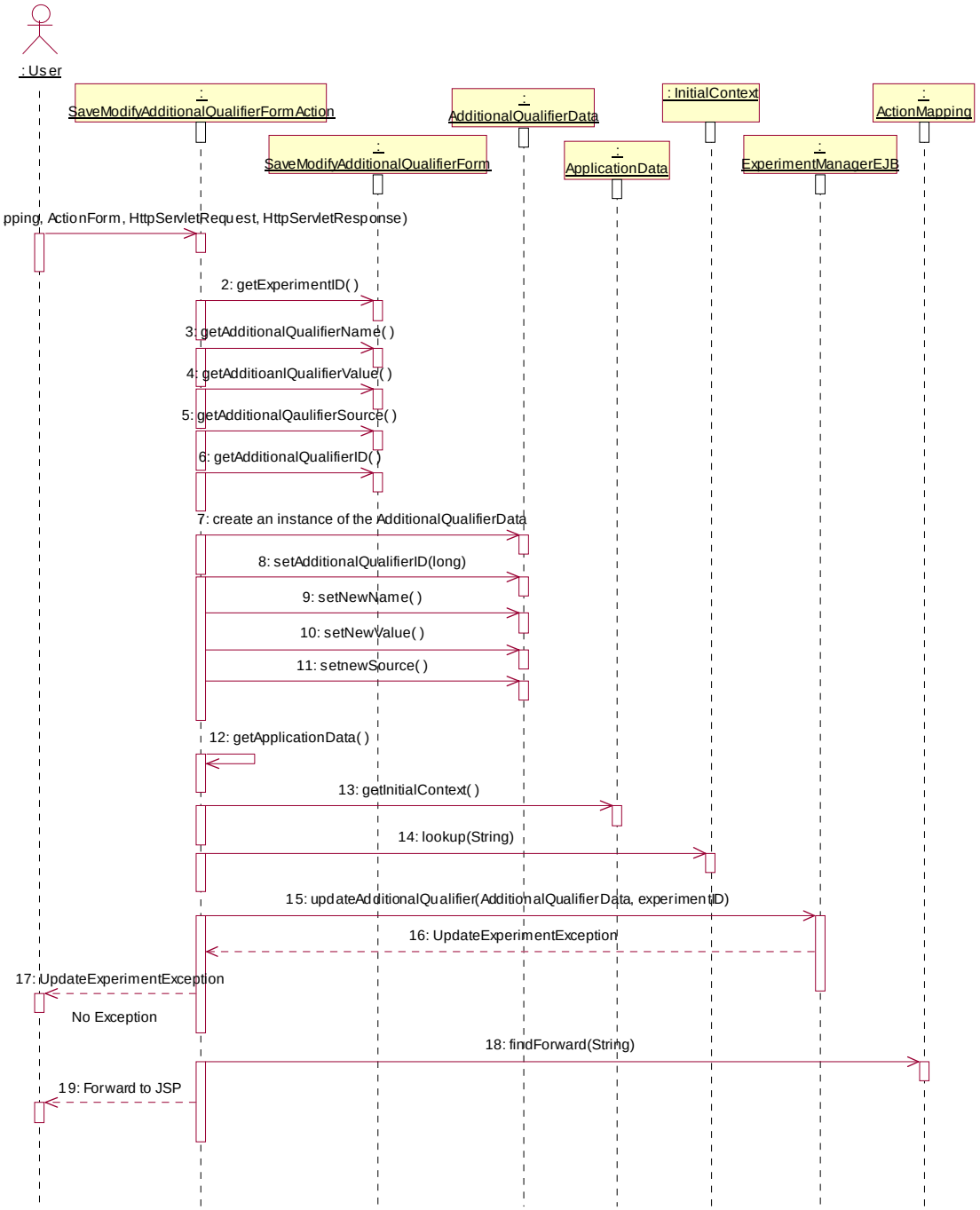
175. BuildSaveNewAdditionalQualifierFormAction.performAction



176. BuildModifyAdditionalQualifierFormAction.performAction

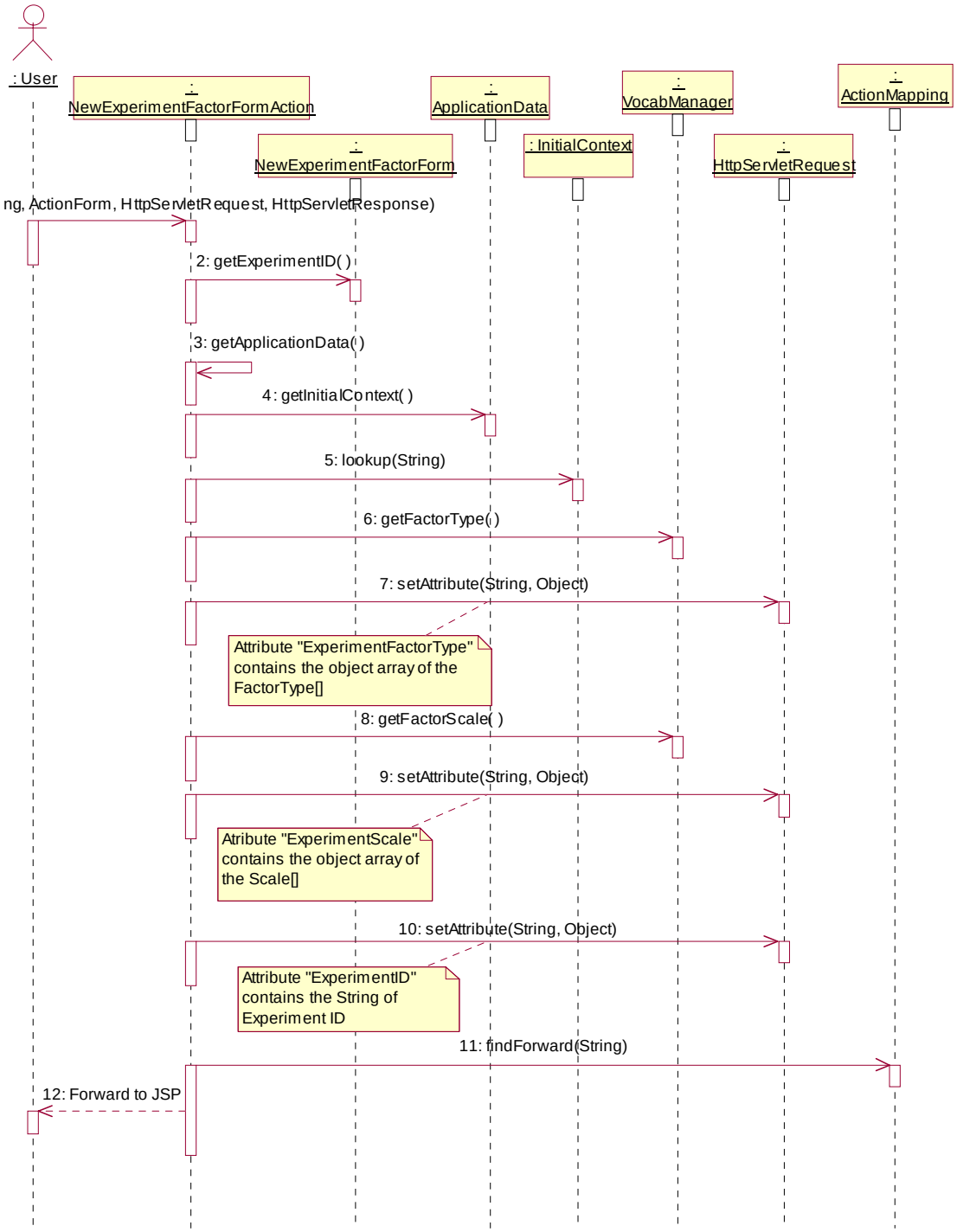

: User
: ModifyAdditionalQualifierFormAction
: AdditionalQualifierData
: ActionMapping
: ModifyAdditionalQualifierForm
: HttpServletRequest
pping, ActionForm, HttpServletRequest, HttpServletResponse)
2: getExperimentID( )
3: getAdditionalQualifierValue( )
4: getAddtionalQualifierName( )
5: getAdditionalQualifierSource( )
6: getAdditionalQualifierID( )
7: create an instance of the AdditionalQualifierData
8: setAdditionalQualifierID(long)
9: setNewName( )
10: setNewValue( )
11: setnewSource( )
12: setAttribute(String, Object)
Attribute "AdditionalQulafierData" contains the object of the AdditionalQualifierData
13: setAttribute(String, Object)
Attribute "ExperimentID" contains the String of the ExperimentID
14: findForward(String)
15: Forward to JSP

177. BuildSaveModifyAdditionalQualifierFormAction.performAction

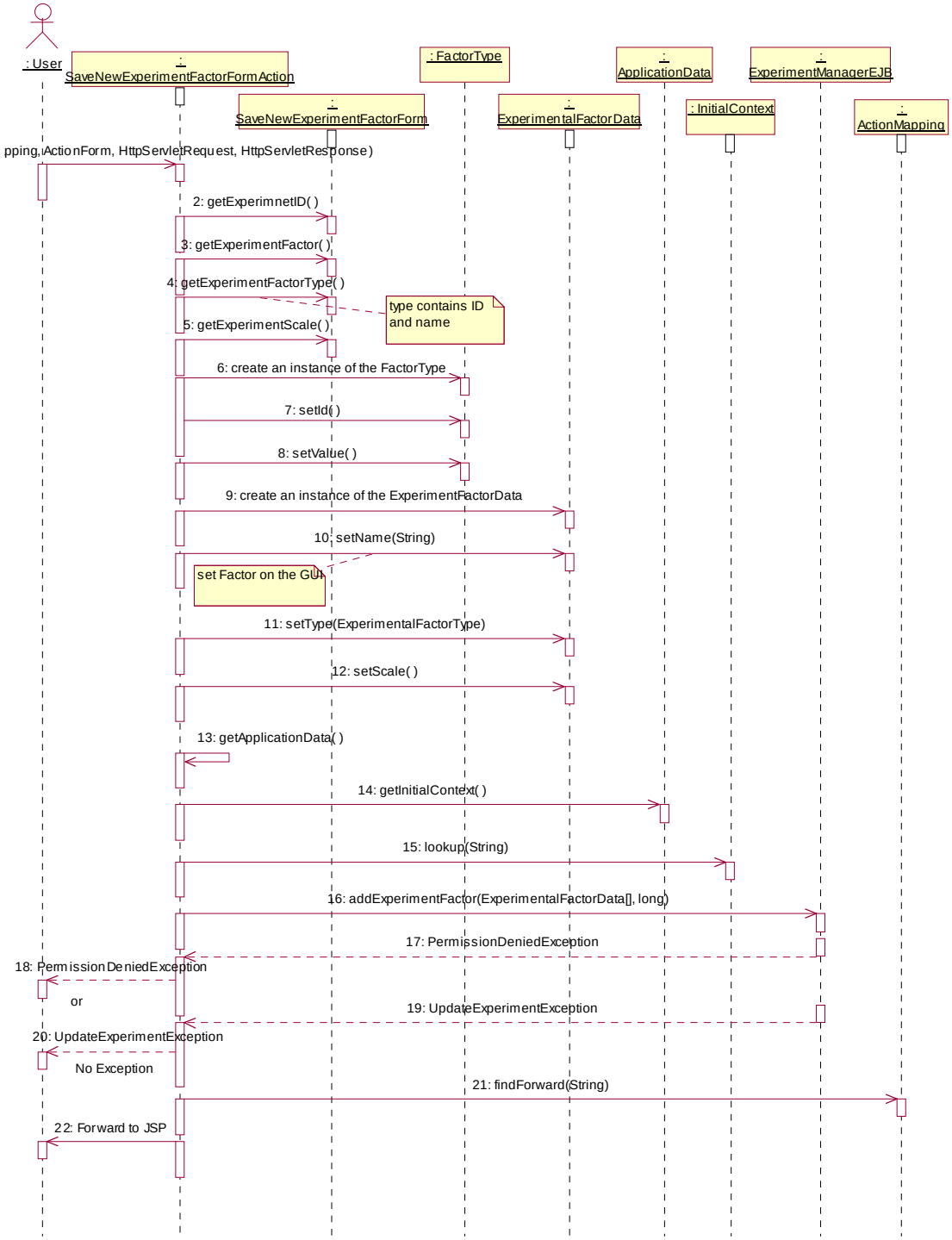


178. BuildRemoveAdditionalQualifierFormAction.performAction

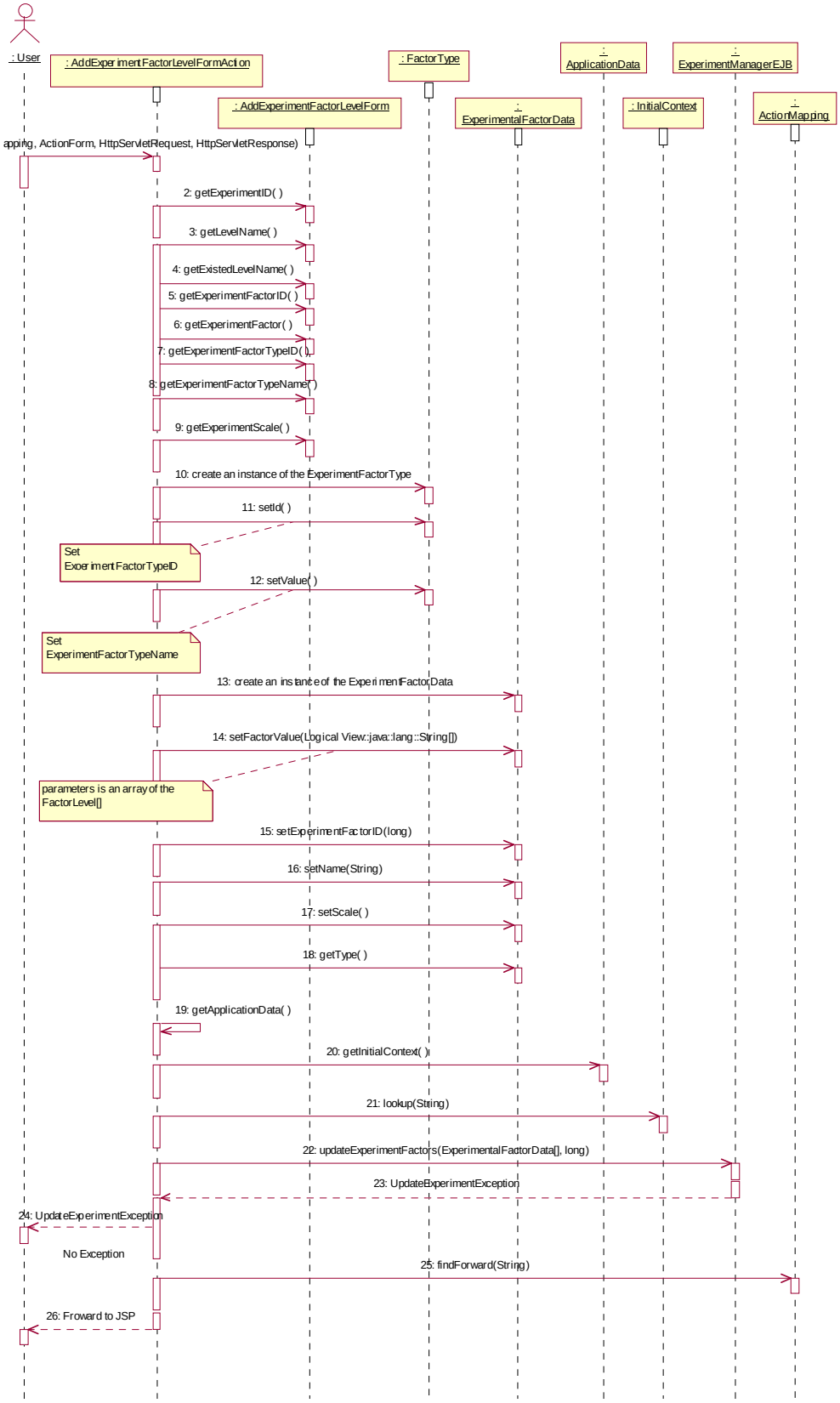
179. BuildNewExperimentFactorFormAction.performAction



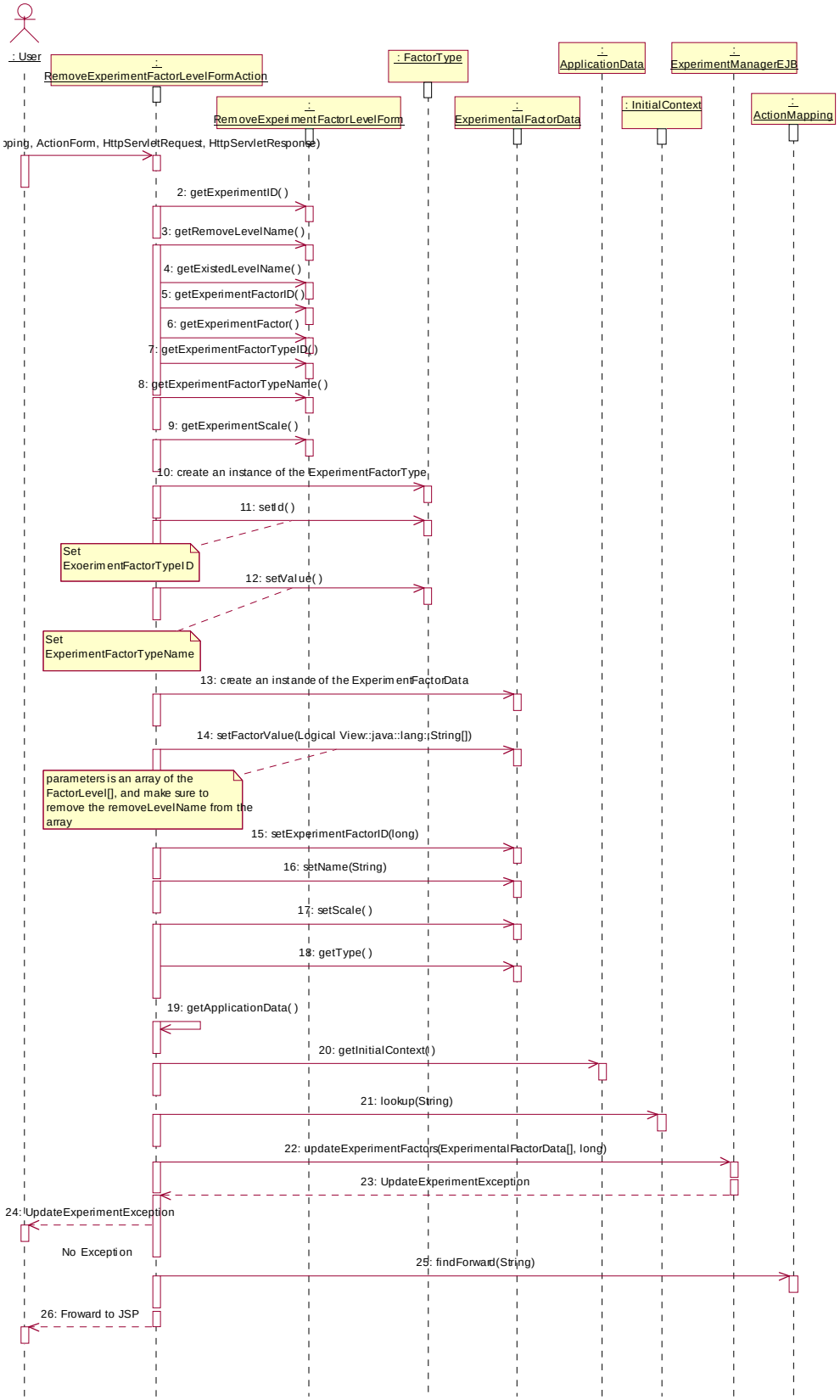
180. BuildSaveNewExperimentFactorFormAction.performAction



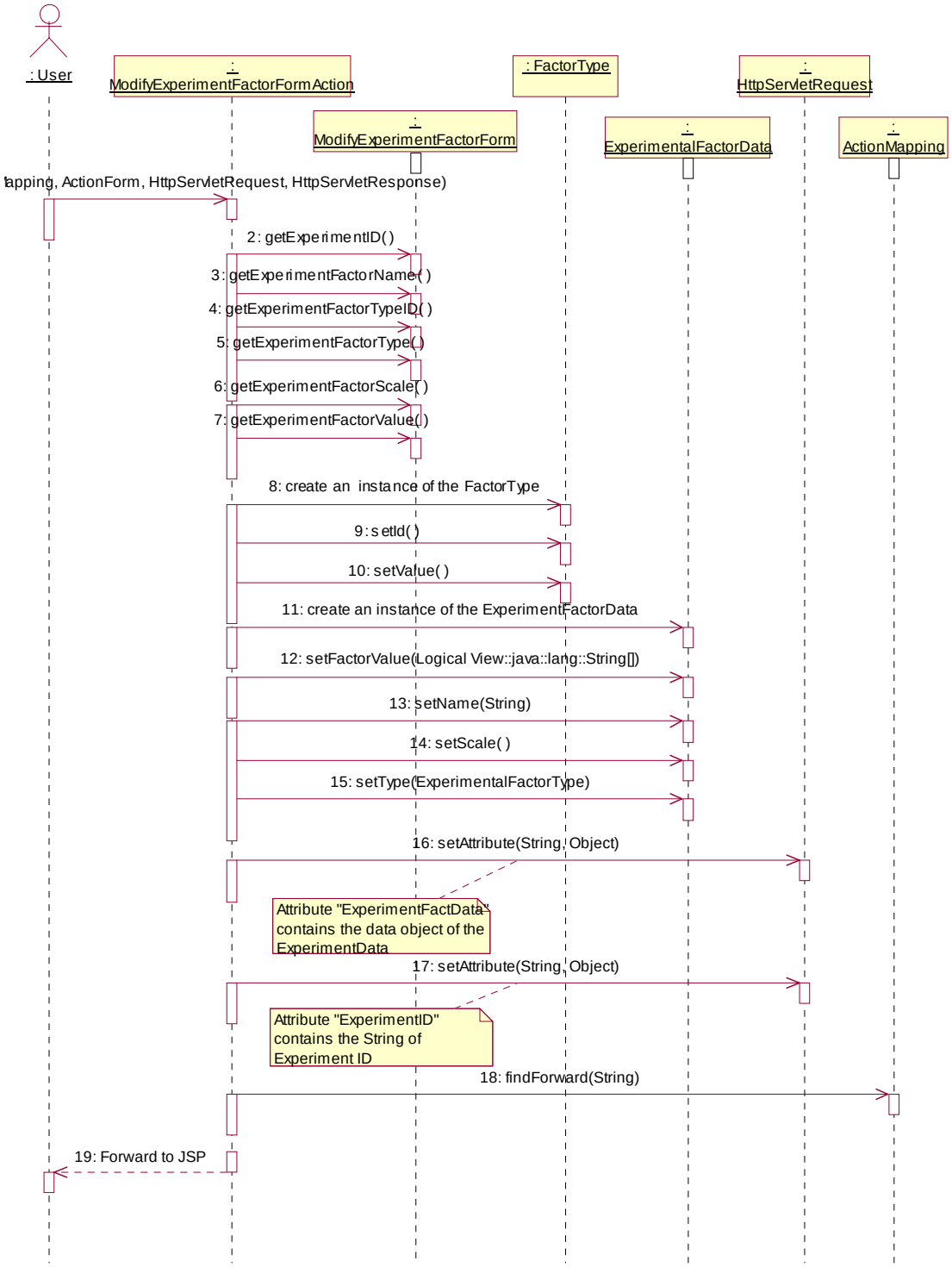
181. BuildAddExperimentFactorLevelFormAction.performAction


: User
: AddExperimentFactorLevelFormAction
: FactorType
: ApplicationData
: ExperimentManagerEJB
: AddExperimentFactorLevelForm
: ExperimentalFactorData
: InitialContext
: ActionMapping
apping, ActionForm, HttpServletRequest, HttpServletResponse)
2: getExperimentID( )
3: getLevelName( )
4: getExistedLevelName( )
5: getExperimentFactorID( )
6: getExperimentFactor( )
7: getExperimentFactorTypeID( )
8: getExperimentFactorTypeName( )
9: getExperimentScale( )
10: create an instance of the ExperimentFactorType
11: setId( )
Set
ExperimentFactorTypeID
12: setValue( )
Set
ExperimentFactorTypeName
13: create an instance of the ExperimentFactorData
14: setFactorValue(Logical View::java::lang::String[])
parameters is an array of the
FactorLevel[]
15: setExperimentFactorID(long)
16: setName(String)
17: setScale( )
18: getType( )
19: getApplicationData( )
20: getInitialContext( )
21: lookup(String)
22: updateExperimentFactors(ExperimentalFactorData[], long)
23: UpdateExperimentException
24: UpdateExperimentException
No Exception
25: findForward(String)
26: Froward to JSP

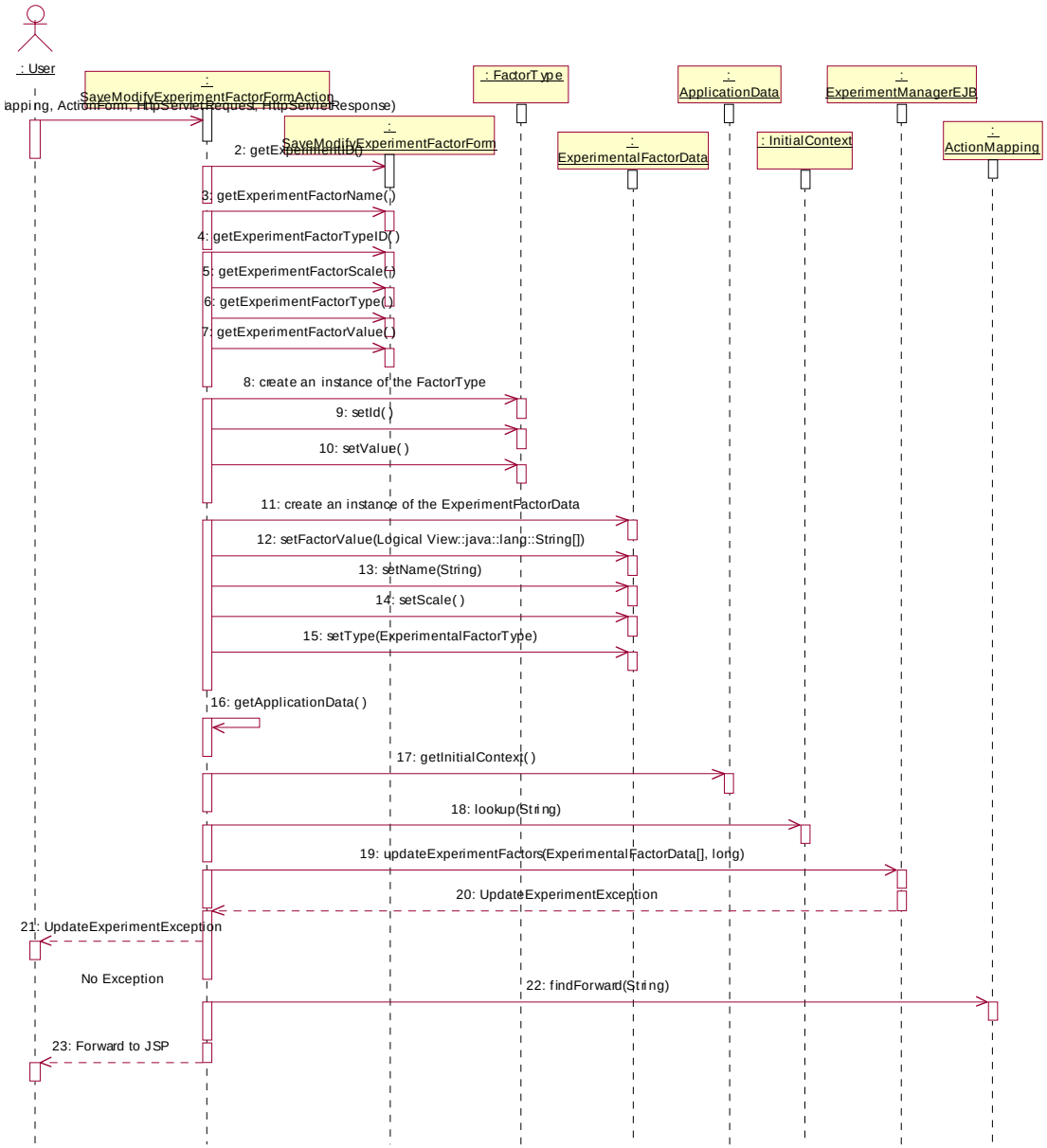
182. BuildRemoveExperimentFactorLevelFormAction.performAction


: User
: RemoveExperimentFactorLevelFormAction
: RemoveExperimentFactorLevelForm
: FactorType
: ExperimentalFactorData
: ApplicationData
: InitialContext
: ExperimentManagerEJB
: ActionMapping
ping, ActionForm, HttpServletRequest, HttpServletResponse)
2: getExperimentID( )
3: getRemoveLevelName( )
4: getExistedLevelName( )
5: getExperimentFactorID( )
6: getExperimentFactor( )
7: getExperimentFactorTypeID( )
8: getExperimentFactorTypeName( )
9: getExperimentScale( )
10: create an instance of the ExperimentFactorType
11: setId( )
Set ExoerimentFactorTypeID
12: setValue( )
Set ExperimentFactorTypeName
13: create an instance of the ExperimentFactorData
14: setFactorValue(Logical View::java::lang::String[])
parameters is an array of the FactorLevel[], and make sure to remove the removeLevelName from the array
15: setExperimentFactorID(long)
16: setName(String)
17: setScale( )
18: getType( )
19: getApplicationData( )
20: getInitialContext( )
21: lookup(String)
22: updateExperimentFactors(ExperimentalFactorData[], long)
23: UpdateExperimentException
24: UpdateExperimentException
No Exception
25: findForward(String)
26: Froward to JSP

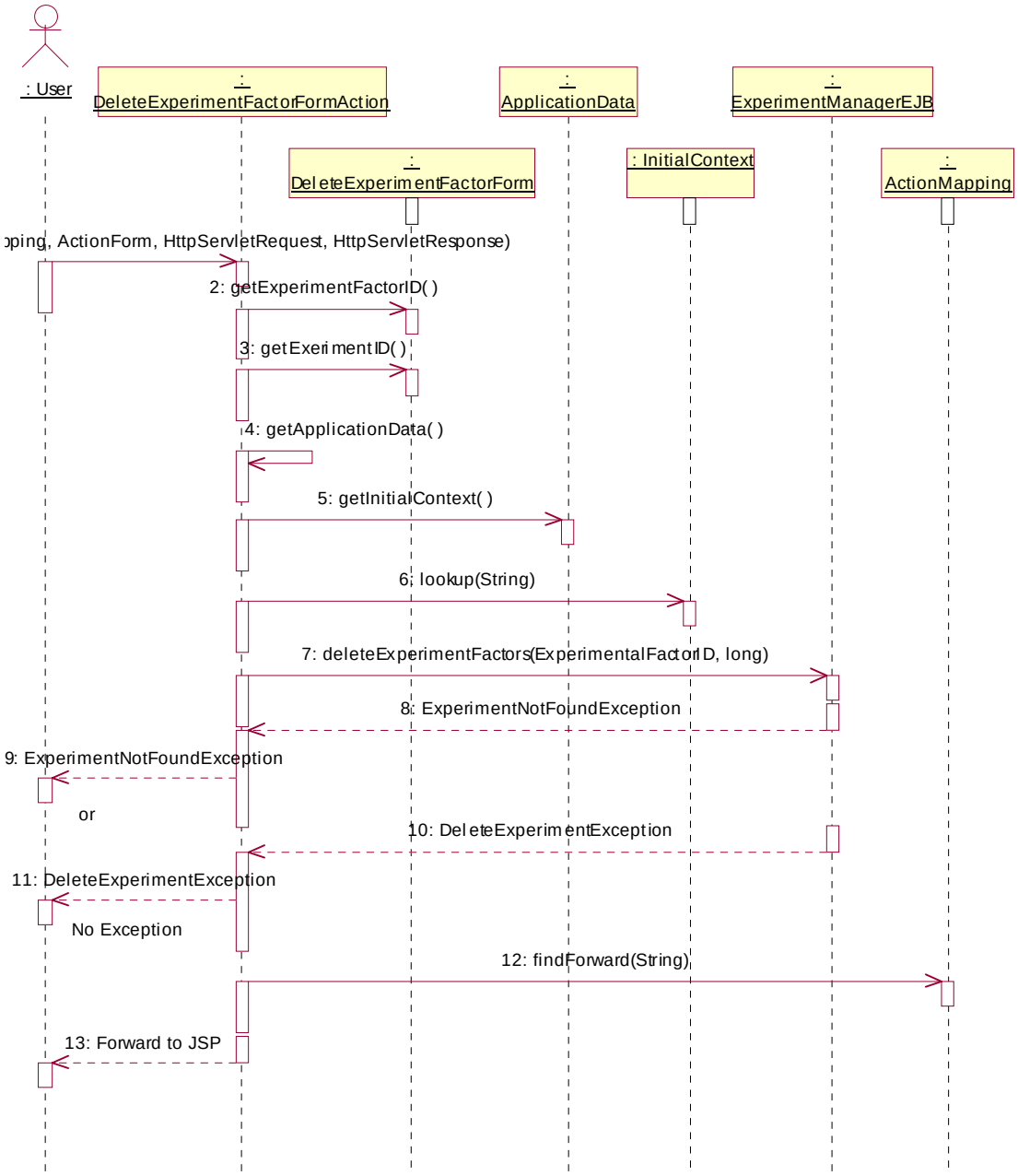
183. BuildModifyExperimentFactorPageFormAction.performAction


: User
: ModifyExperimentFactorFormAction
: FactorType
: HttpServletRequest
: ModifyExperimentFactorForm
: ExperimentalFactorData
: ActionMapping
1apping, ActionForm, HttpServletRequest, HttpServletResponse)
2: getExperimentID( )
3: getExperimentFactorName( )
4: getExperimentFactorTypeID( )
5: getExperimentFactorType( )
6: getExperimentFactorScaleT( )
7: getExperimentFactorValue( )
8: create an instance of the FactorType
9: setId( )
10: setValue( )
11: create an instance of the ExperimentFactorData
12: setFactorValue(Logical View::java::lang::String[])
13: setName(String)
14: setScale( )
15: setType(ExperimentalFactorType)
16: setAttribute(String, Object)
Attribute "ExperimentFactData" contains the data object of the ExperimentData
17: setAttribute(String, Object)
Attribute "ExperimentID" contains the String of Experiment ID
18: findForward(String)
19: Forward to JSP

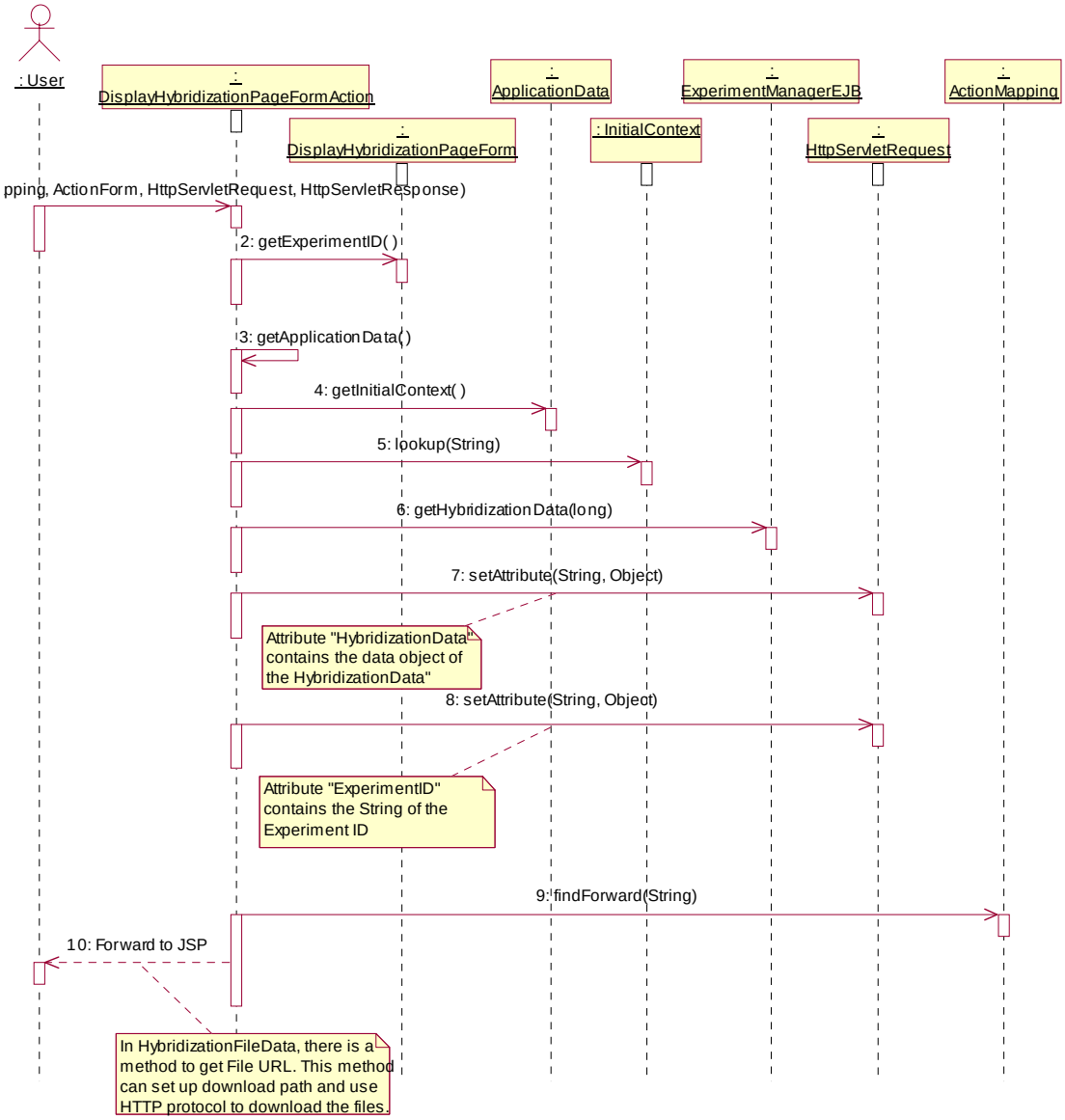
184. BuildSaveModifyExperimentFactorFormAction.performAction



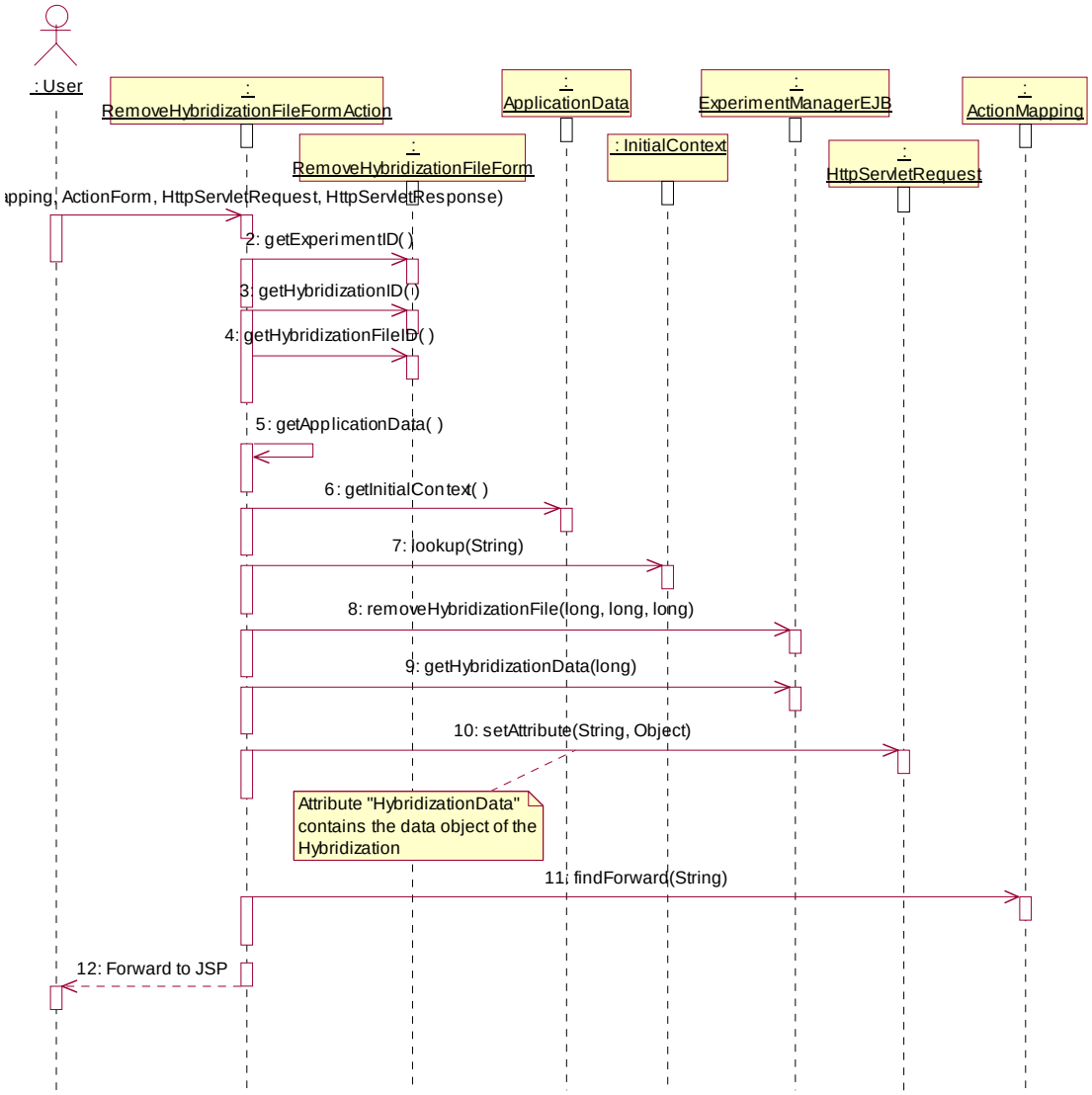
185. BuildDeleteExperimentFactorFormAction.peroformAction



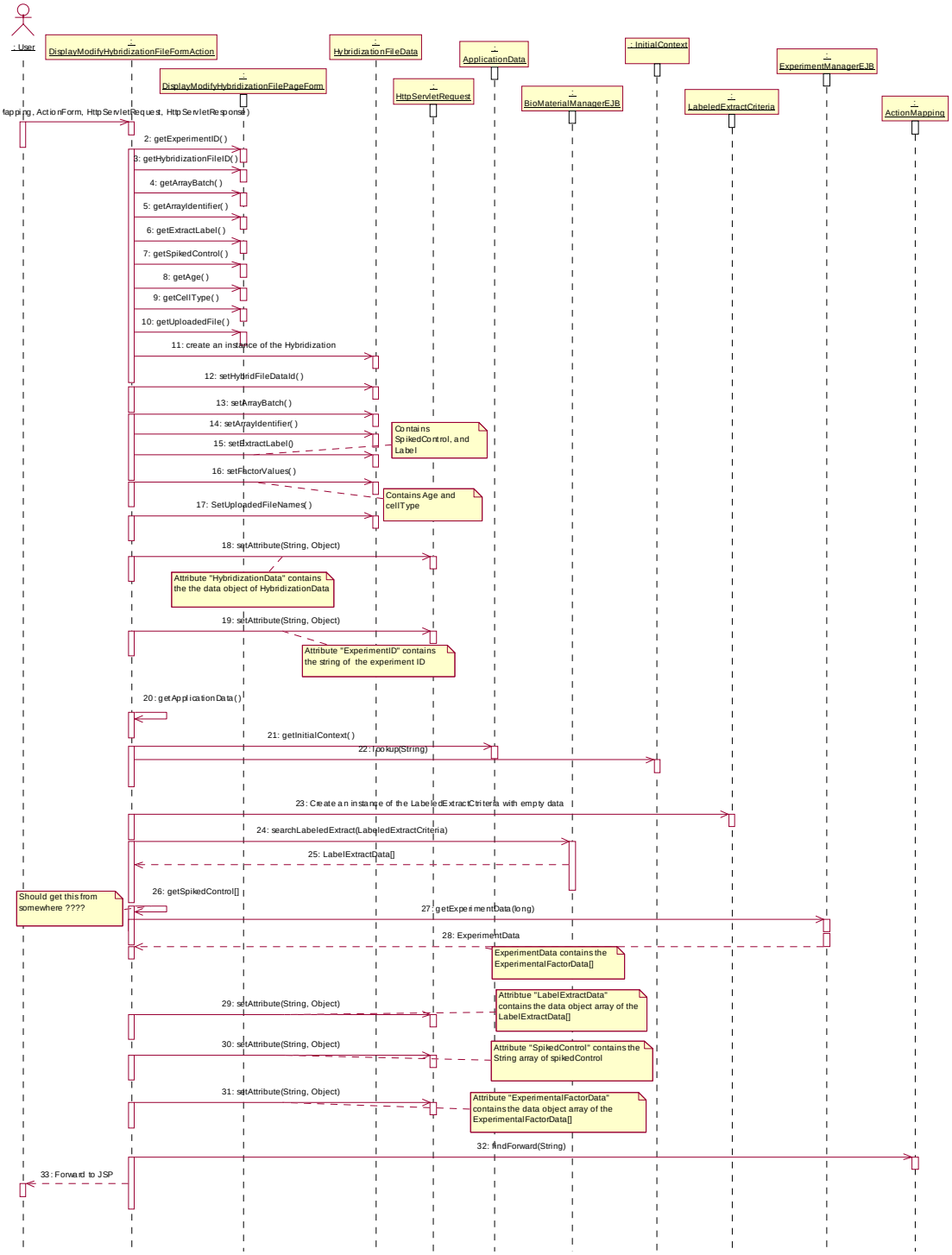
186. BuildDisplayHybridizationPageFormAction.performAction



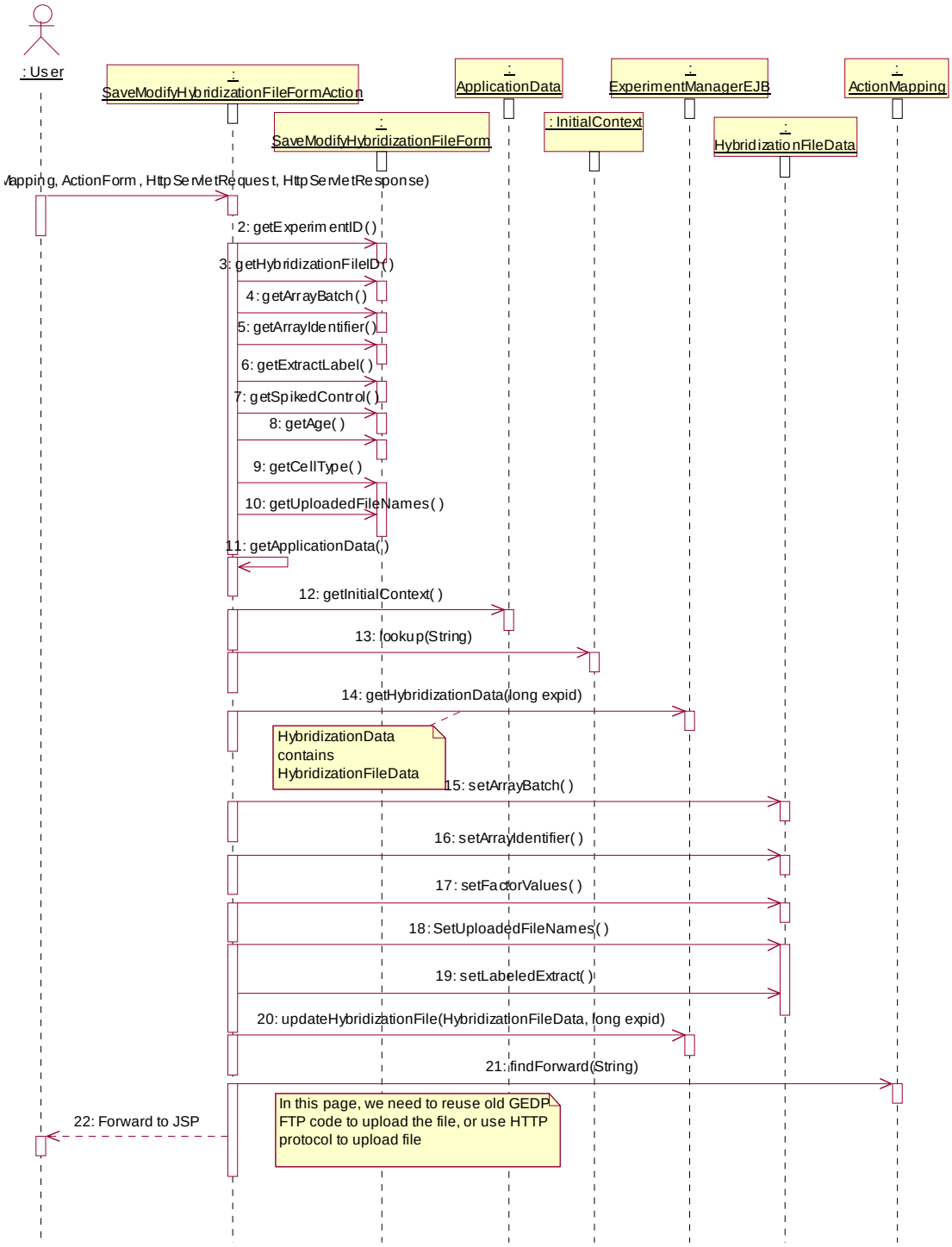
187. BuildRemoveHybridizationFileFormAction.performAction



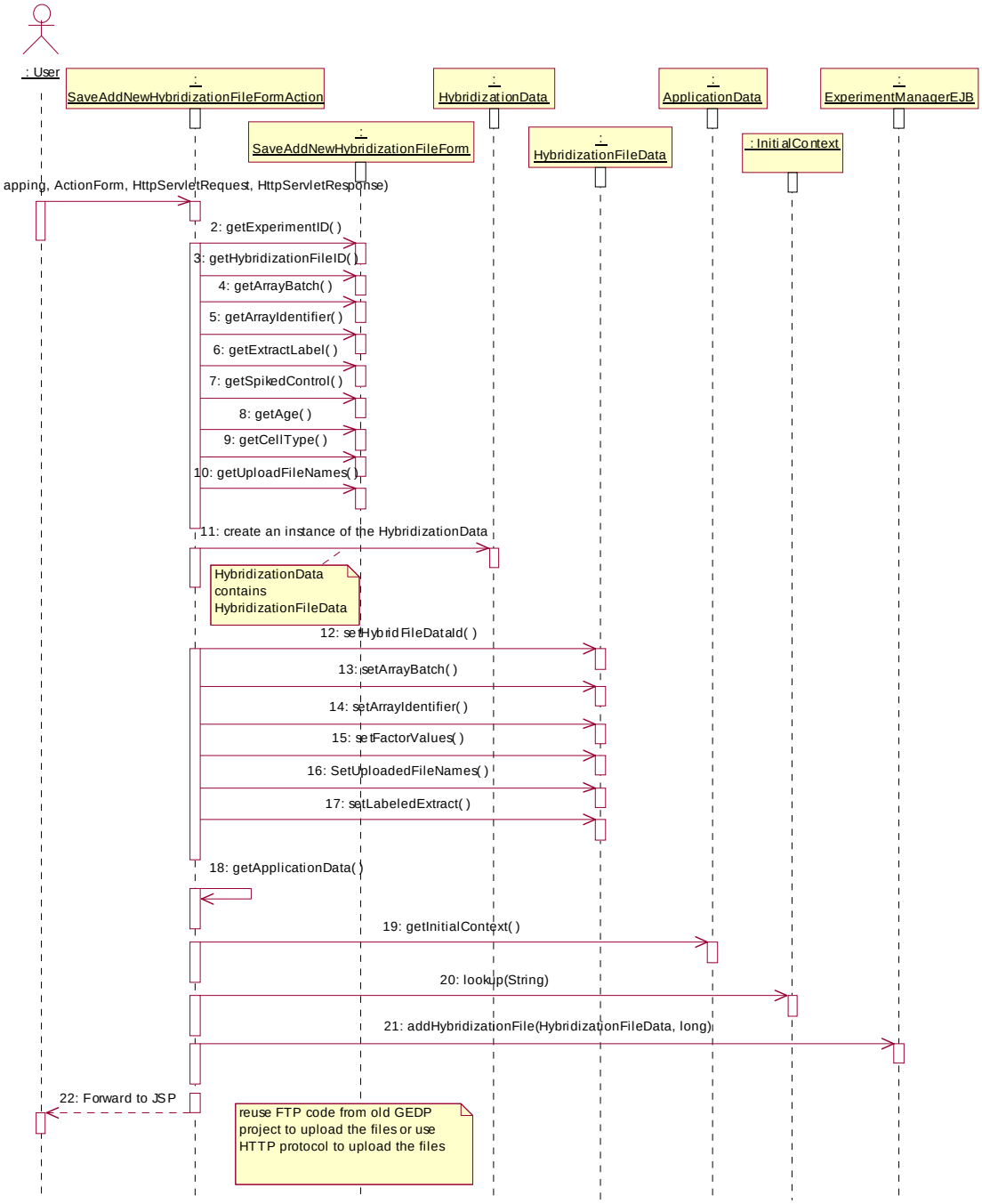
188. BuildDisplayModifyHybridizationFormAction.peroformAction


: User
DisplayModifyHybridizationFileFormAction
DisplayModifyHybridizationFilePageForm
HybridizationFileData
HttpServletRequest
ApplicationData
BioMaterialManagerEJB
: InitialContext
LabeledExtractCriteria
ExperimentManagerEJB
ActionMapping
(mapping, ActionForm, HttpServletRequest, HttpServletResponse)
2: getExperimentID( )
3: getHybridizationFileID( )
4: getArrayBatch( )
5: getArrayIdentifier( )
6: getExtractLabel( )
7: getSpikedControl( )
8: getAge( )
9: getCellType( )
10: getUploadedFile( )
11: create an instance of the Hybridization
12: setHybridFileDataId( )
13: setArrayBatch( )
14: setArrayIdentifier( )
15: setExtractLabel()
Contains SpikedControl, and Label
16: setFactorValues( )
Contains Age and cellType
17: SetUploadedFileNames( )
18: setAttribute(String, Object)
Attribute "HybridizationData" contains the the data object of HybridizationData
19: setAttribute(String, Object)
Attribute "ExperimentID" contains the string of the experiment ID
20: getApplicationData( )
21: getInitialContext( )
22: lookup(String)
23: Create an instance of the LabeledExtractCriteria with empty data
24: searchLabeledExtract(LabeledExtractCriteria)
25: LabelExtractData[]
26: getSpikedControl[]
Should get this from somewhere ????
27: getExperimentData(long)
28: ExperimentData
ExperimentData contains the ExperimentalFactorData[]
29: setAttribute(String, Object)
Attribtue "LabelExtractData" contains the data object array of the LabelExtractData[]
30: setAttribute(String, Object)
Attribute "SpikedControl" contains the String array of spikedControl
31: setAttribute(String, Object)
Attribute "ExperimentalFactorData" contains the data object array of the ExperimentalFactorData[]
32: findForward(String)
33: Forward to JSP

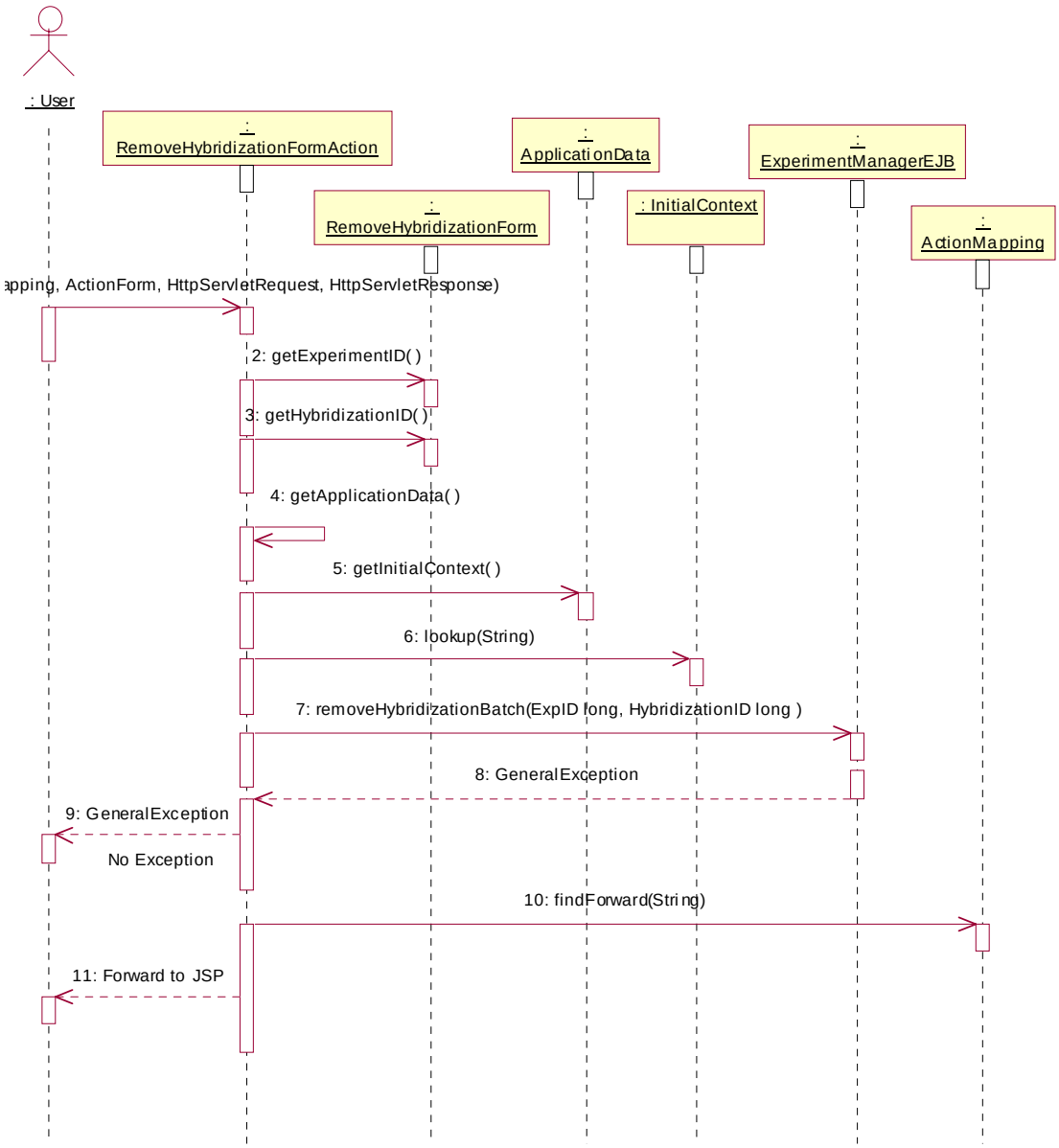
189. BuildSaveModifyHybridizationFileFormAction.performAction



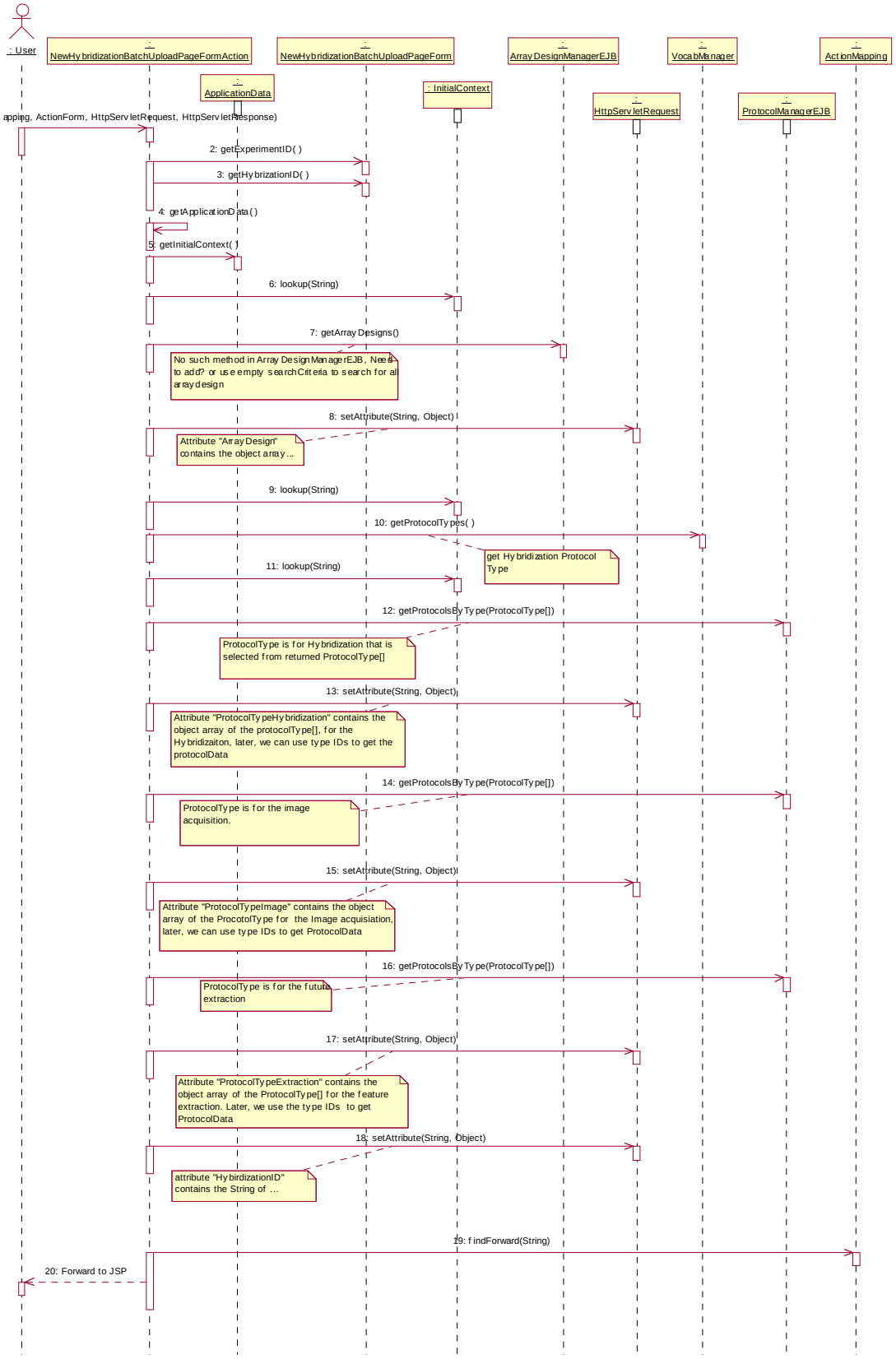
190. BuilldSaveAddNewHybridizationFileFormAction.performAction



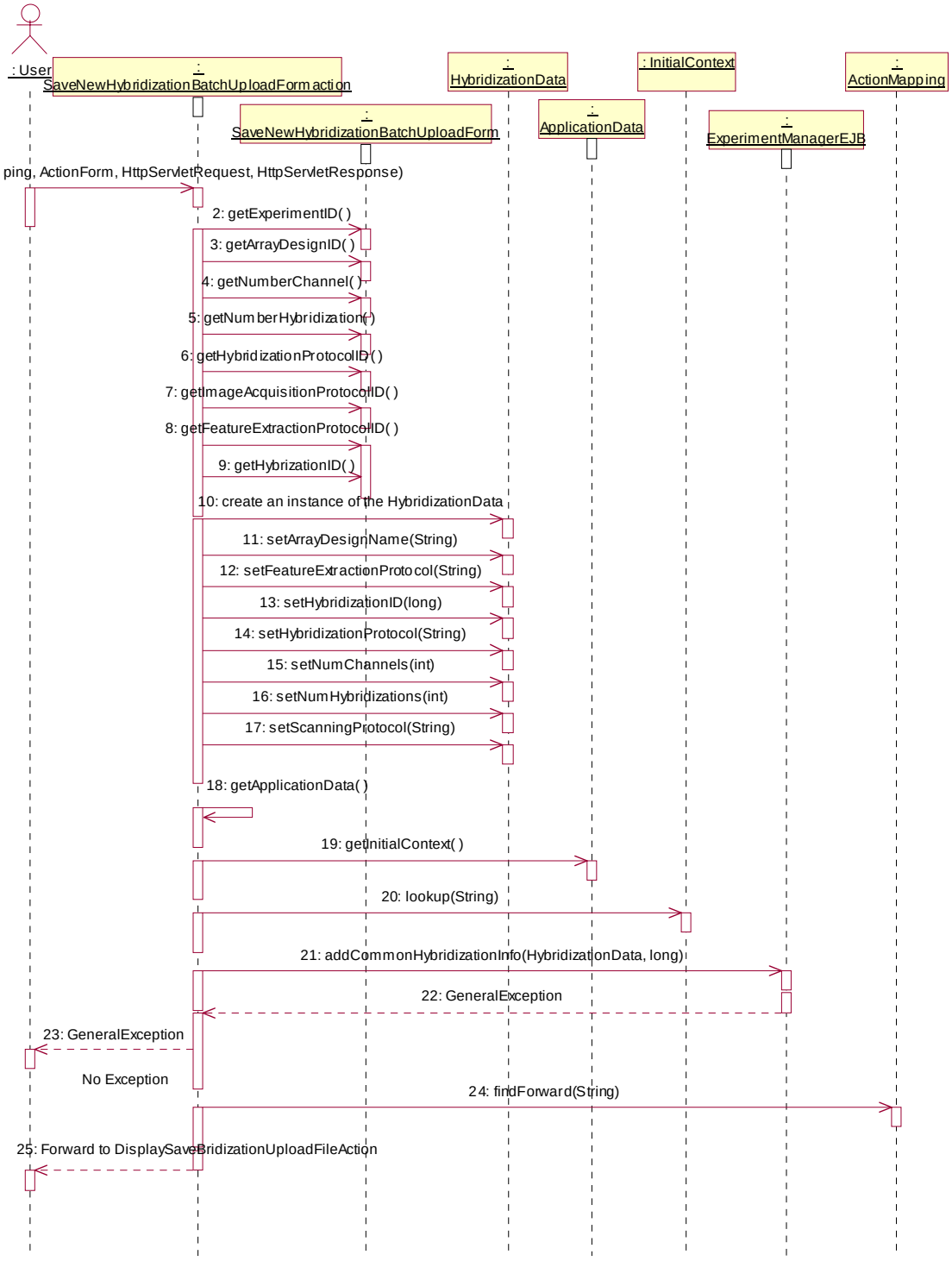
191. BuildRemoveHybridizationBatchFormAction.performAction



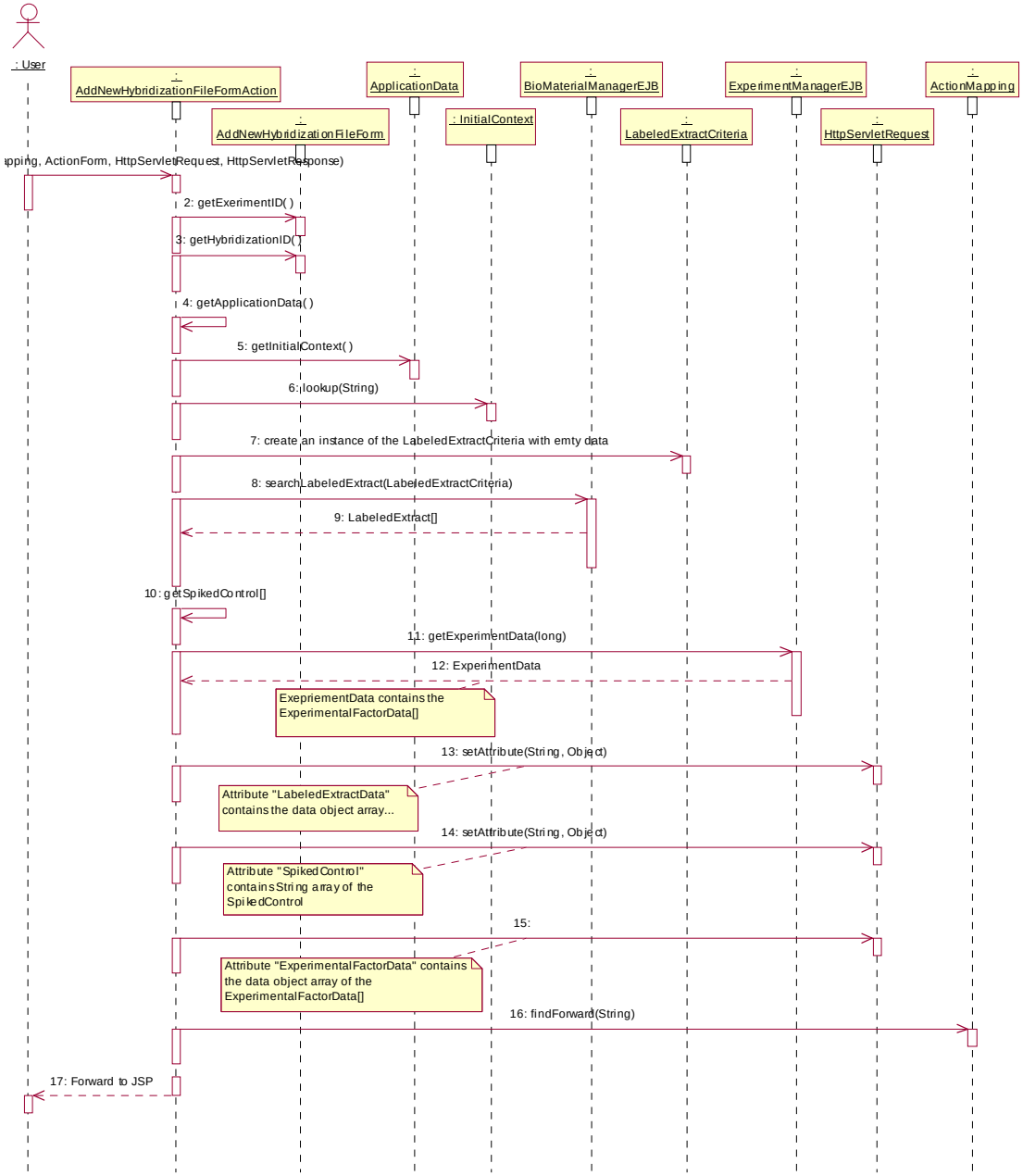
192. BuildNewHybridizationBatchUploadPageFormAction.performAction


: User
: NewHybridizationBatchUploadPageFormAction
: NewHybridizationBatchUploadPageForm
: ArrayDesignManagerEJB
: VocabManager
: ActionMapping
: ApplicationData
: InitialContext
: HttpServletRequest
: ProtocolManagerEJB
apping, ActionForm, HttpServletRequest, HttpServletResponse)
2: getExperimentID( )
3: getHybrizationID( )
4: getApplicationData( )
5: getInitialContext( )
6: lookup(String)
7: getArrayDesigns()
No such method in ArrayDesignManagerEJB, Need to add? or use empty searchCiteria to search for all arraydesign
8: setAttribute(String, Object)
Attribute "ArrayDesign" contains the object array ...
9: lookup(String)
10: getProtocolTypes( )
get Hybridization Protocol Type
11: lookup(String)
12: getProtocolsByType(ProtocolType[])
ProtocolType is for Hybridization that is selected from returned ProtocolType[]
13: setAttribute(String, Object)
Attribute "ProtocolTypeHybridization" contains the object array of the protocolType[], for the Hybridizaiton, later, we can use type IDs to get the protocolData
14: getProtocolsByType(ProtocolType[])
ProtocolType is for the image acquisition.
15: setAttribute(String, Object)
Attribute "ProtocolTypeImage" contains the object array of the ProcotolType for the Image acquisiation, later, we can use type IDs to get ProtocolData
16: getProtocolsByType(ProtocolType[])
ProtocolType is for the future extraction
17: setAttribute(String, Object)
Attribute "ProtocolTypeExtraction" contains the object array of the ProtocolType[] for the feature extraction. Later, we use the type IDs to get ProtocolData
18: setAttribute(String, Object)
attribute "HybirdizationID" contains the String of ...
19: findForward(String)
20: Forward to JSP

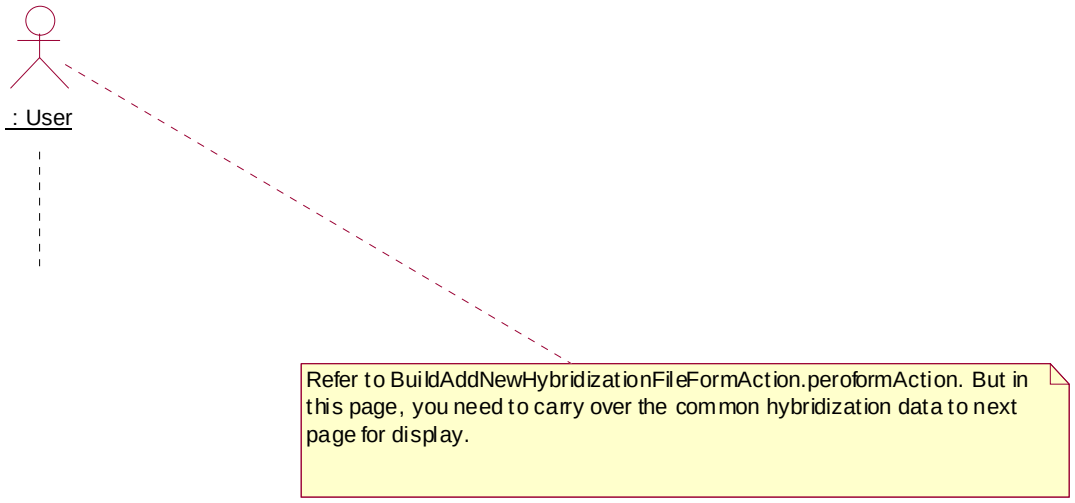
193. BuildSaveNewHybridizationBatchUploadFormAction.performAction



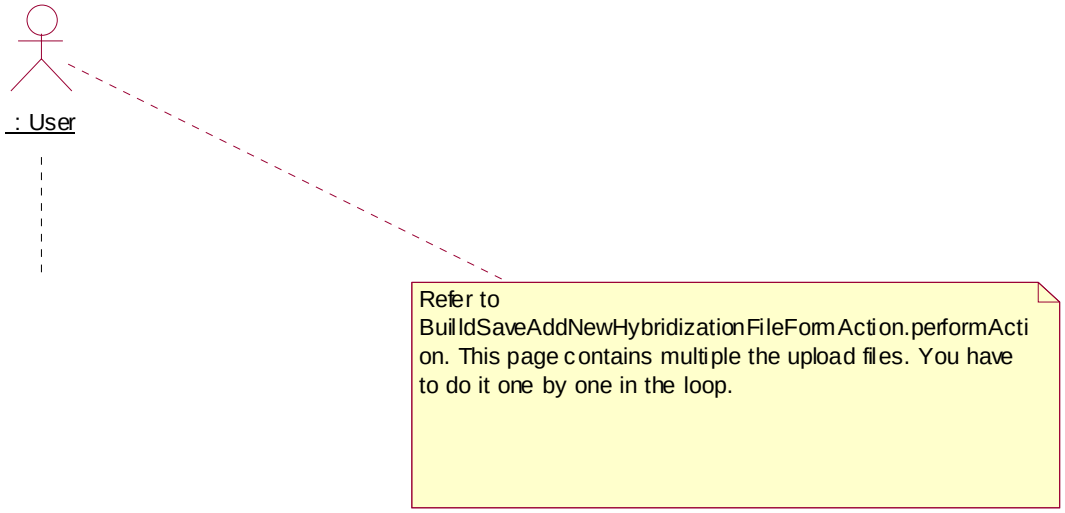
194. BuildAddNewHybridizationFileFormAction.performAction



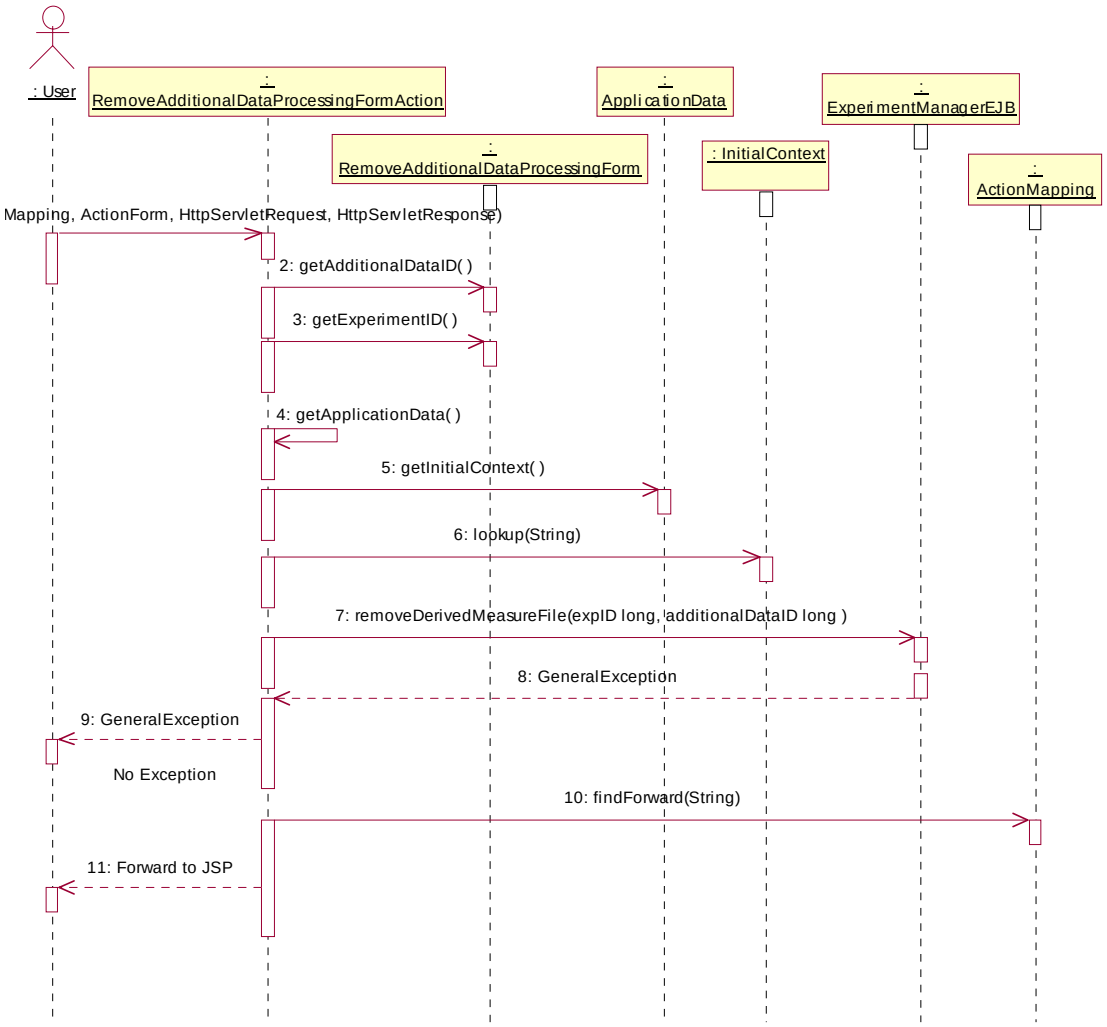
195. BuildDisplaySaveHybridizationBatchUploadFileFormAction .peroformAction



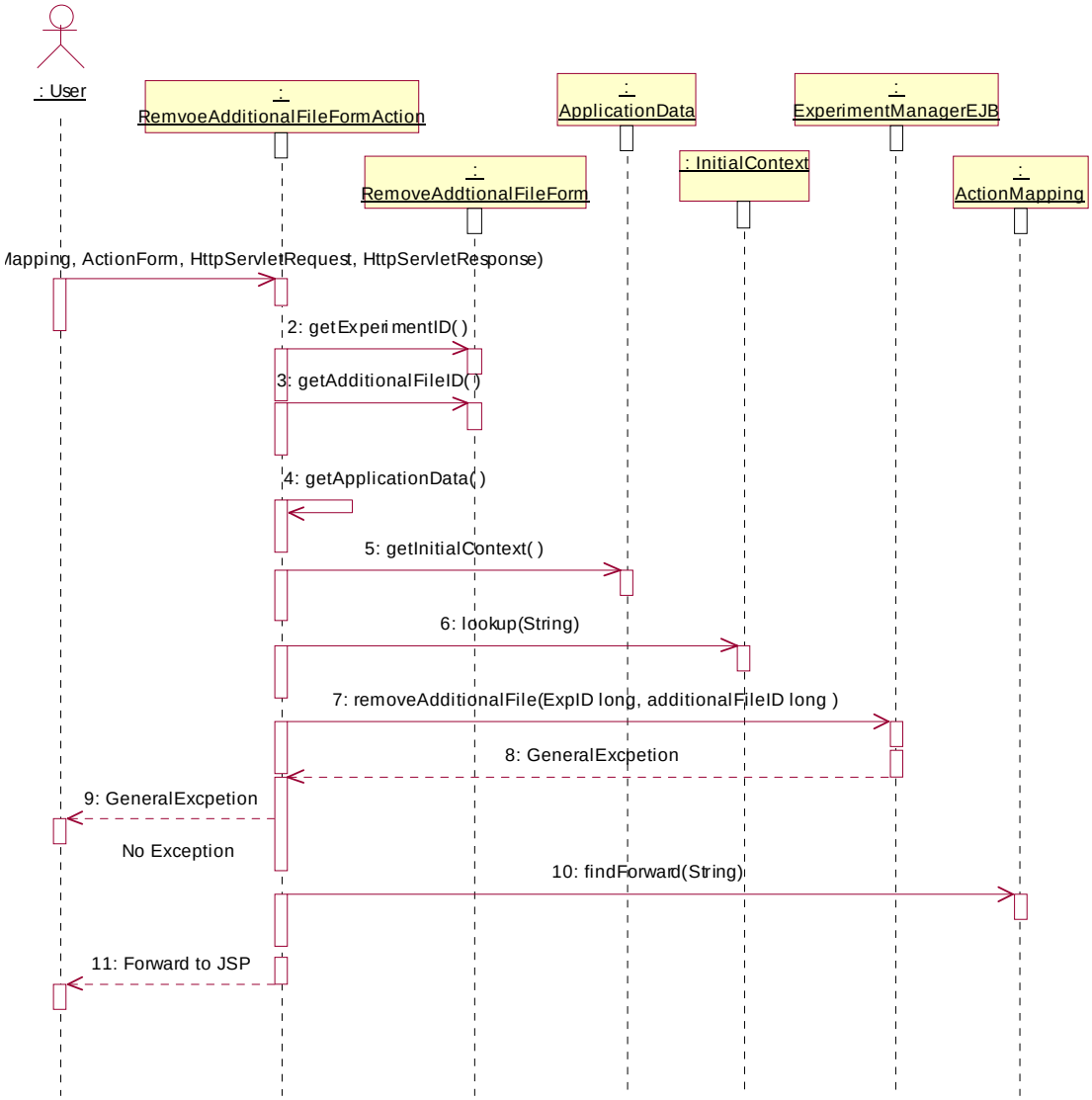
196. BuildSaveHybridizationBatchUploadFileFormAction.performAction



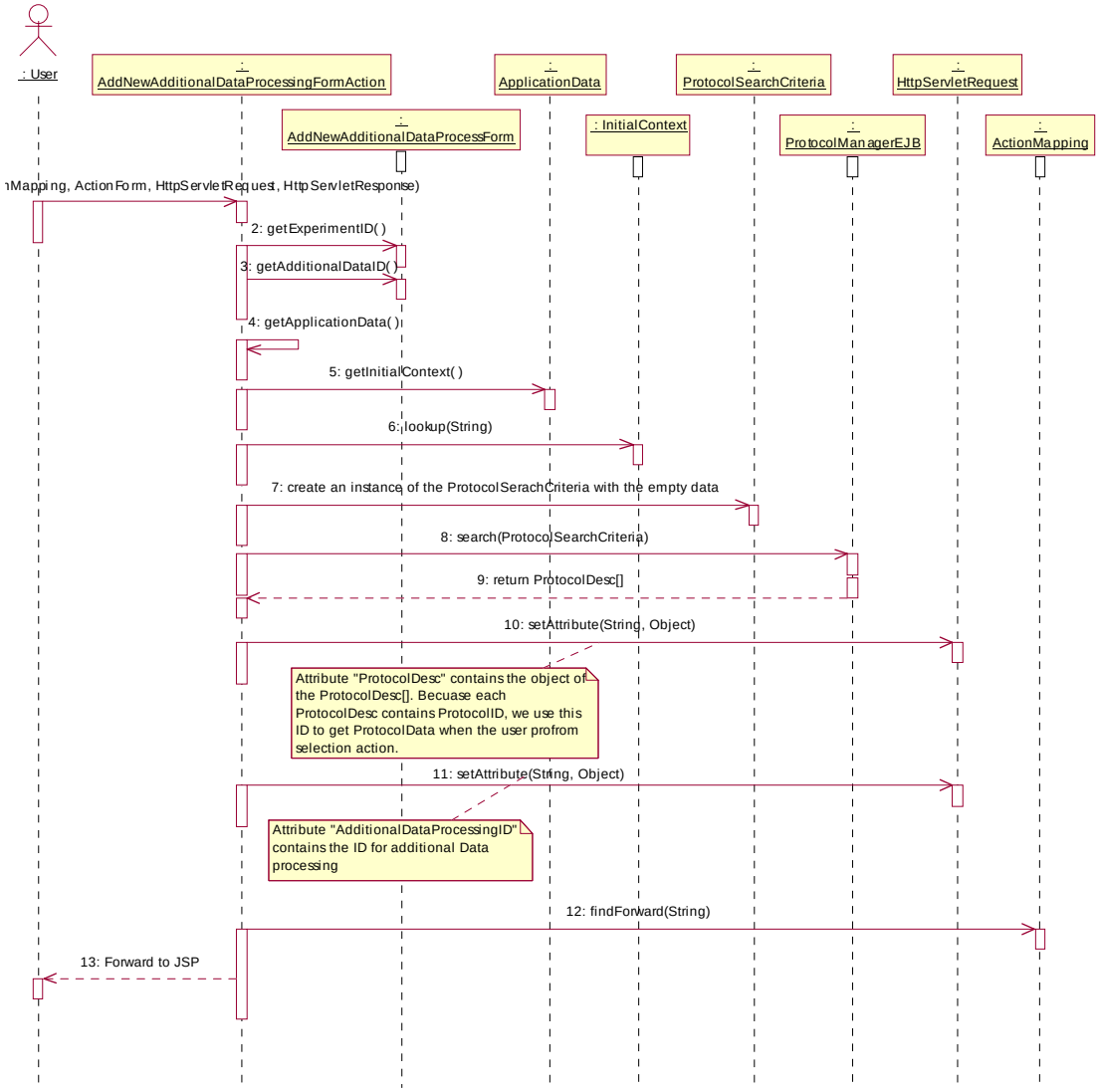
197. BuildRemoveAdditionalDataProcessingFormAction.performAction



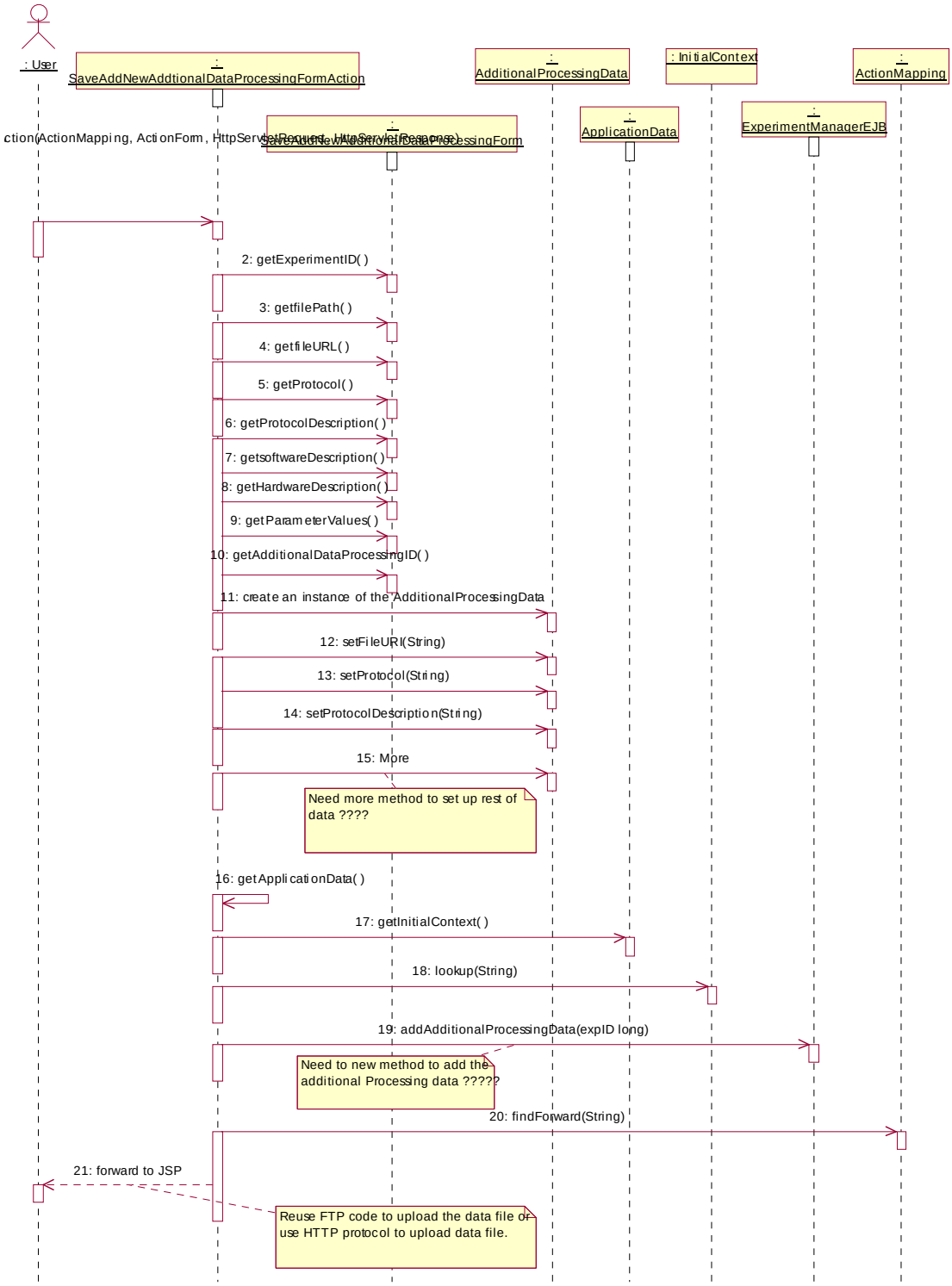
198. BuildRemoveAdditionalFileFormAction.performAction



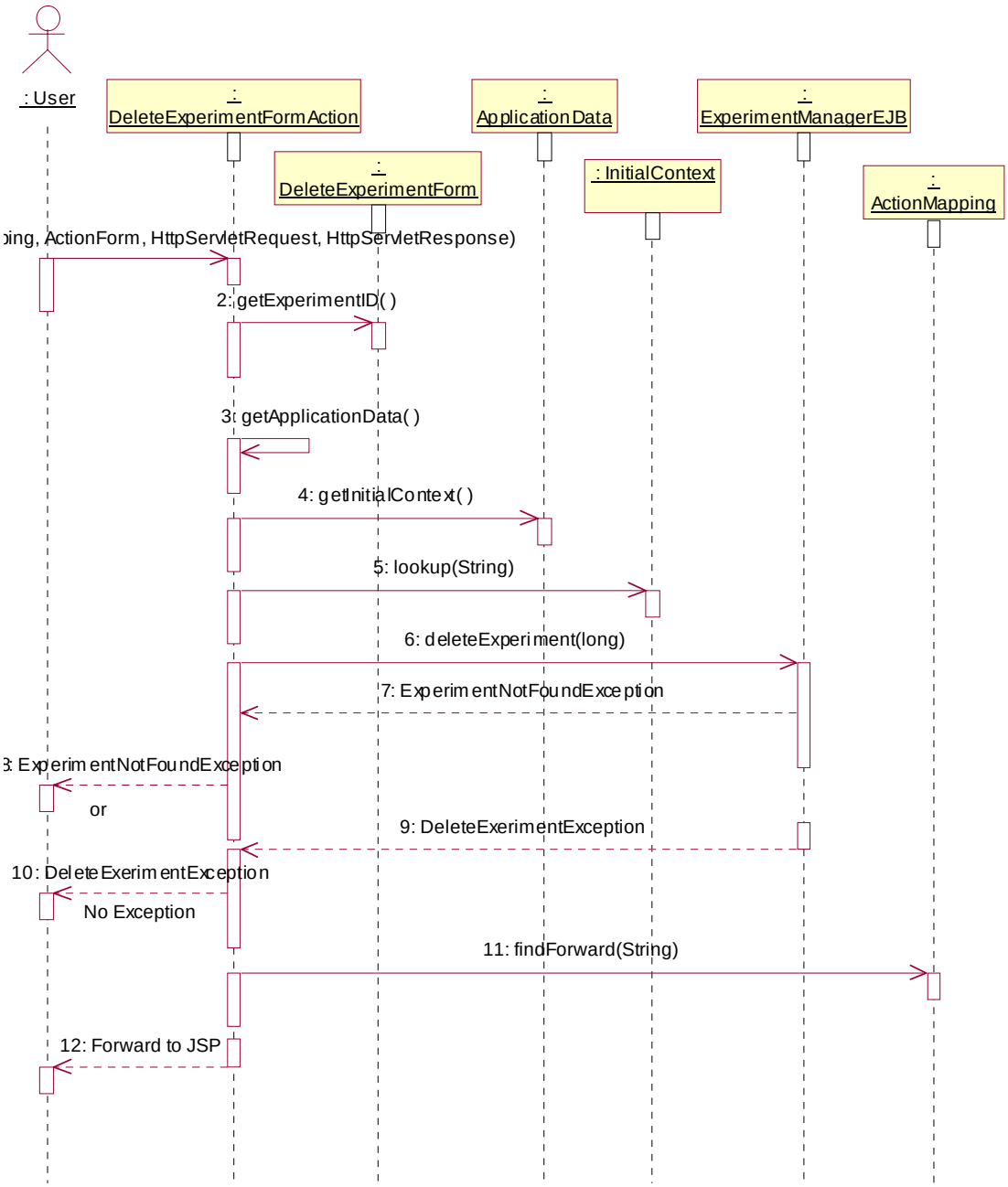
199. BuildAddNewAdditionalDataProcessingFormAction.performAction



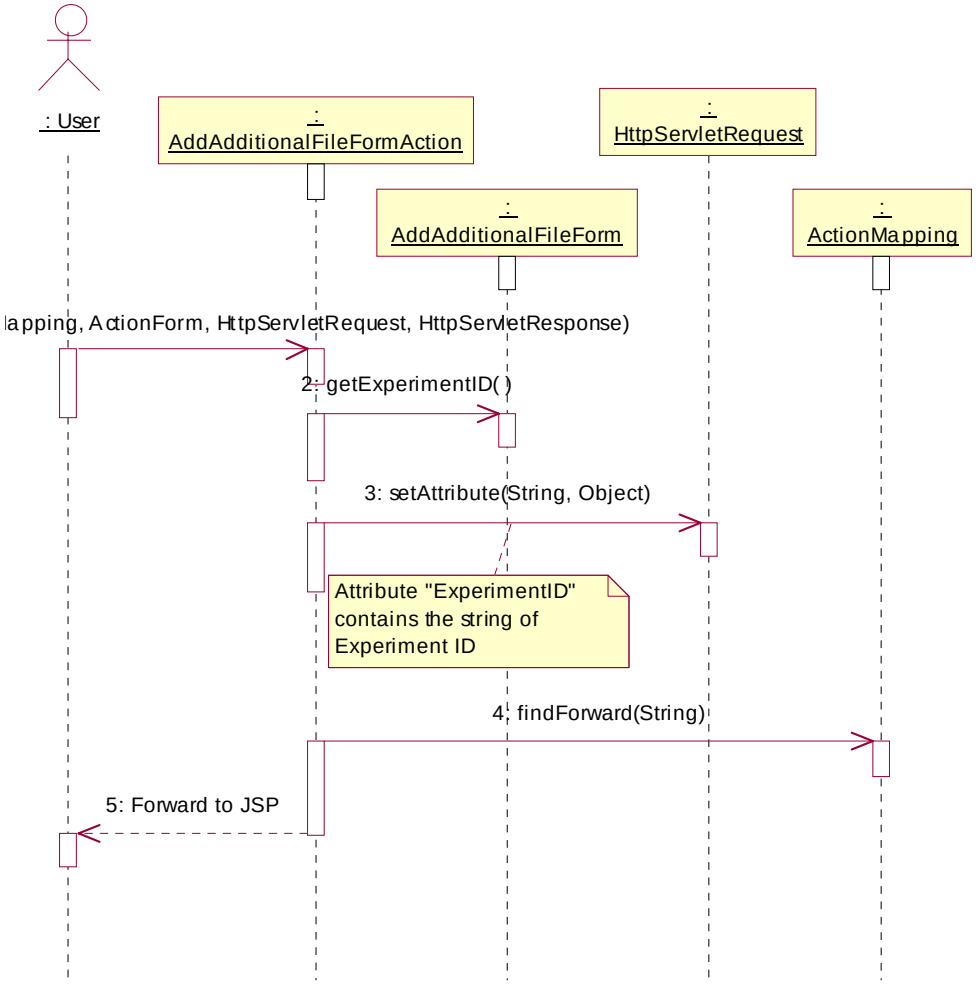
200. BuildSaveAddNewAdditionalDataProcessingFormAction.peroformAction


: User
: SaveAddNewAddtionalDataProcessingFormAction
: AdditionalProcessingData
: InitialContext
: ActionMapping
: ApplicationData
: ExperimentManagerEJB
ction(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse)
2: getExperimentID( )
3: getfilePath( )
4: getfileURL( )
5: getProtocol( )
6: getProtocolDescription( )
7: getsoftwareDescription( )
8: getHardwareDescription( )
9: getParameterValues( )
10: getAdditionalDataProcessingID( )
11: create an instance of the AdditionalProcessingData
12: setFileURI(String)
13: setProtocol(String)
14: setProtocolDescription(String)
15: More
Need more method to set up rest of data ????
16: getApplicationData( )
17: getInitialContext( )
18: lookup(String)
19: addAdditionalProcessingData(expID long)
Need to new method to add the additional Processing data ?????
20: findForward(String)
21: forward to JSP
Reuse FTP code to upload the data file or use HTTP protocol to upload data file.

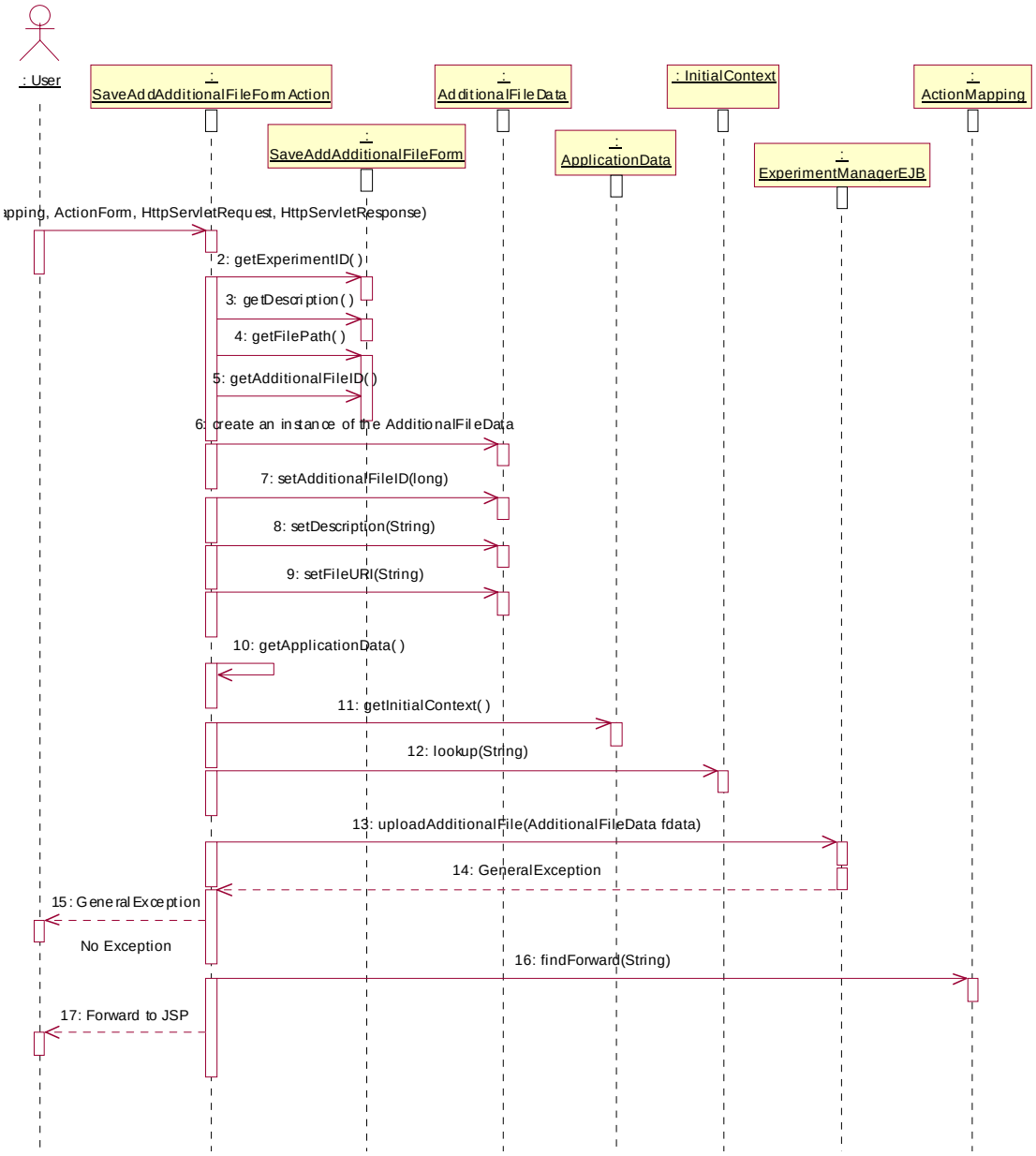
201. BuildDeleteExperimentFormAction.performAction



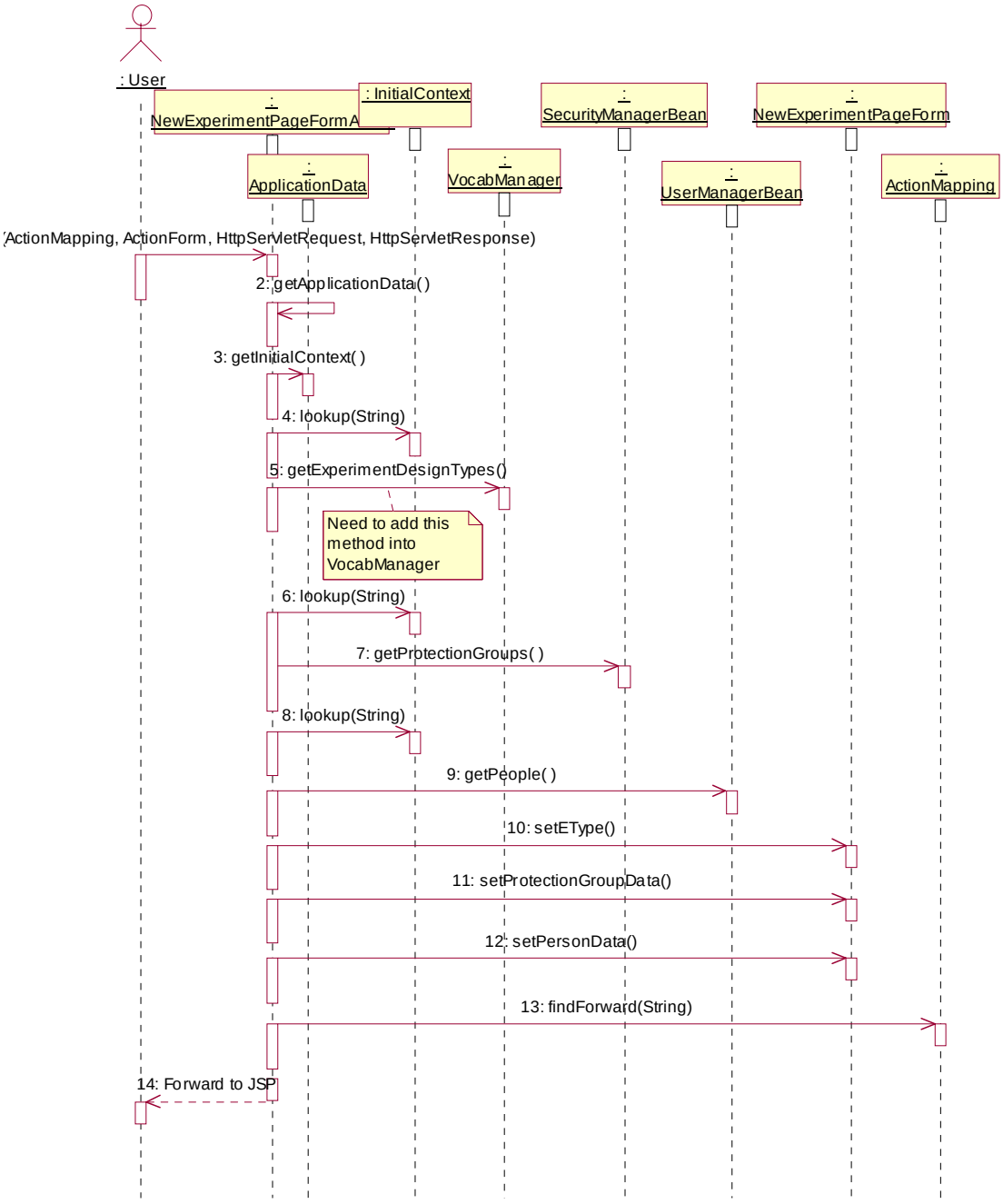
202. BuildAddAdditionalFileFormAction.performAction



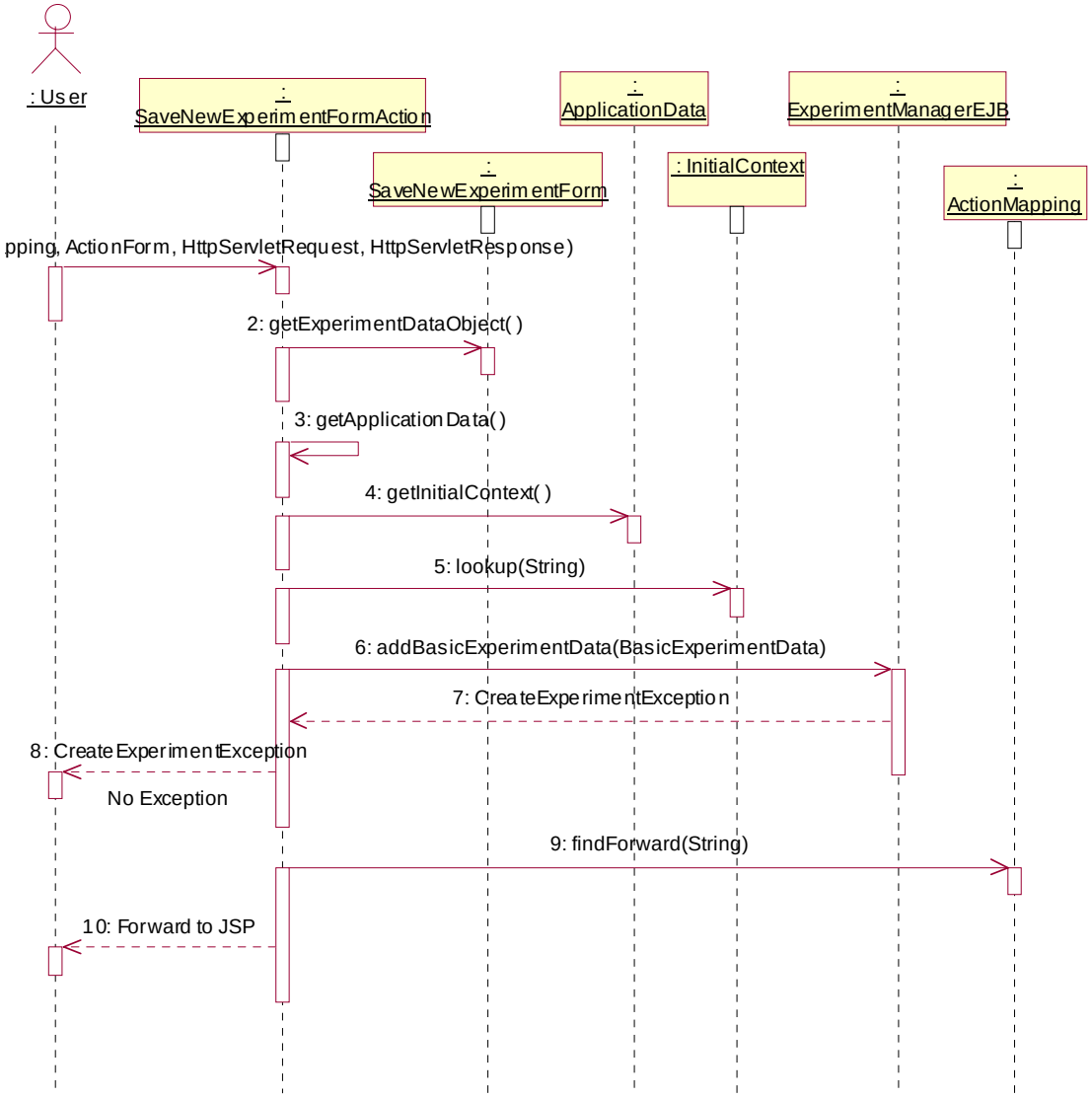
203. BuildSaveAddAdditionalFileFormAction.performAction



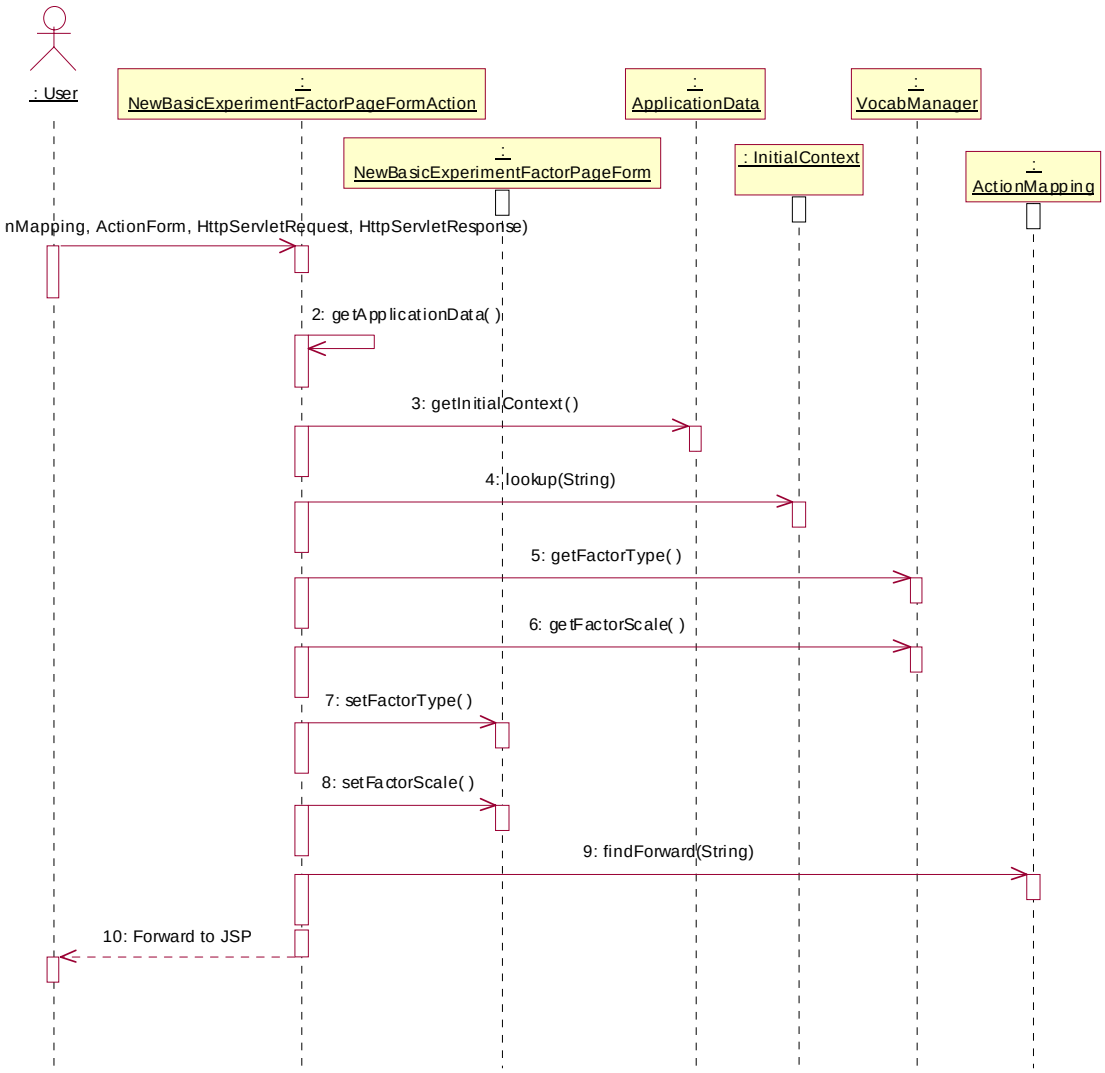
204. BuildNewExperimentPageFormAction.performAction



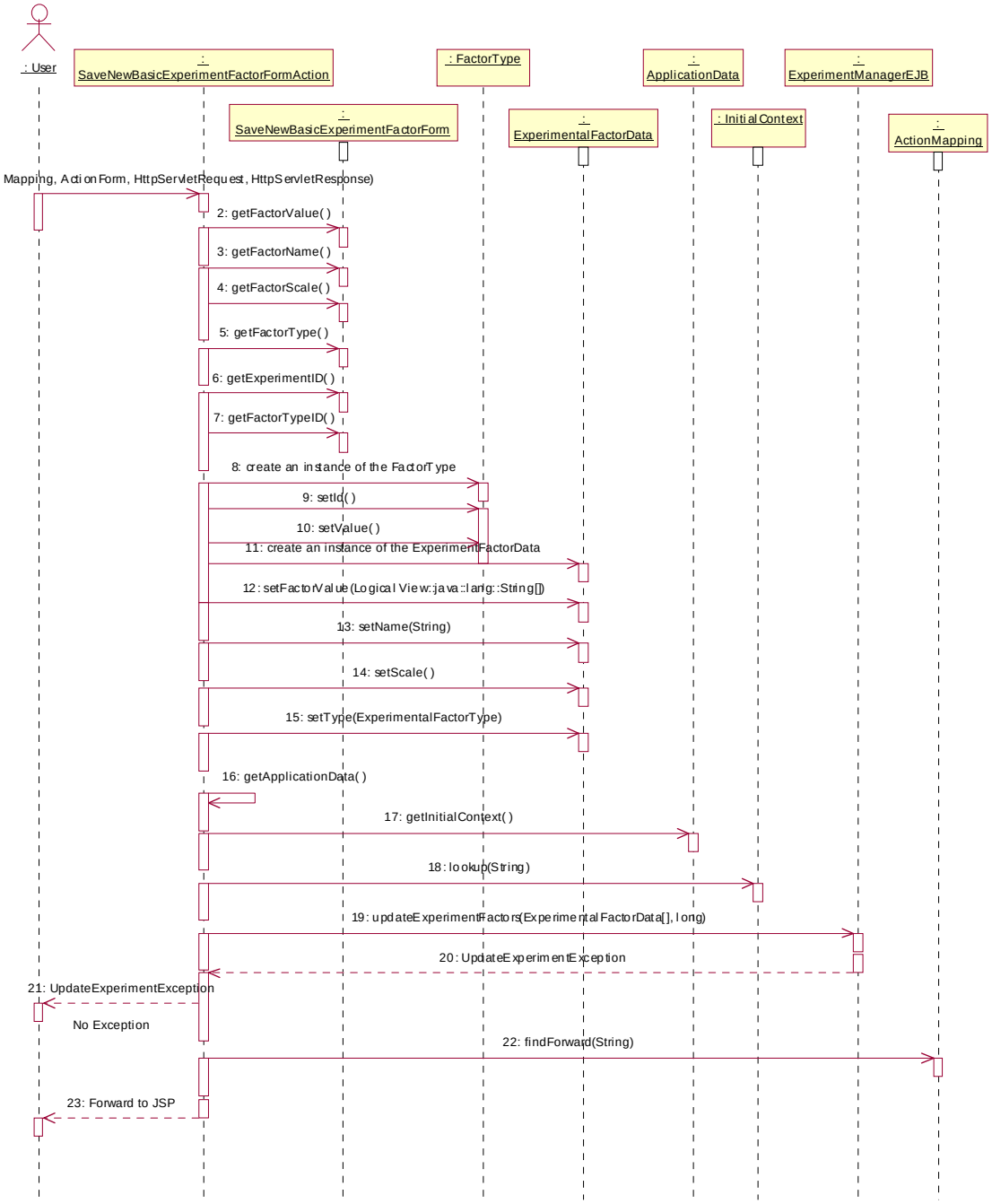
205. BuildSaveNewExperimentBasicInfoFormAction.perfomAction



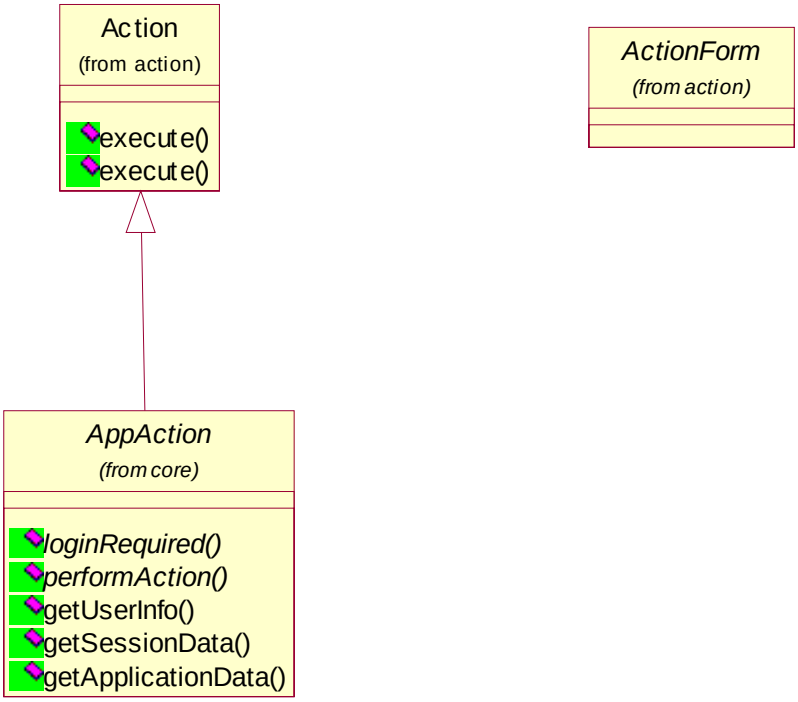
206. BuildNewBasicExperimentFactorPageFormAction.perform Action



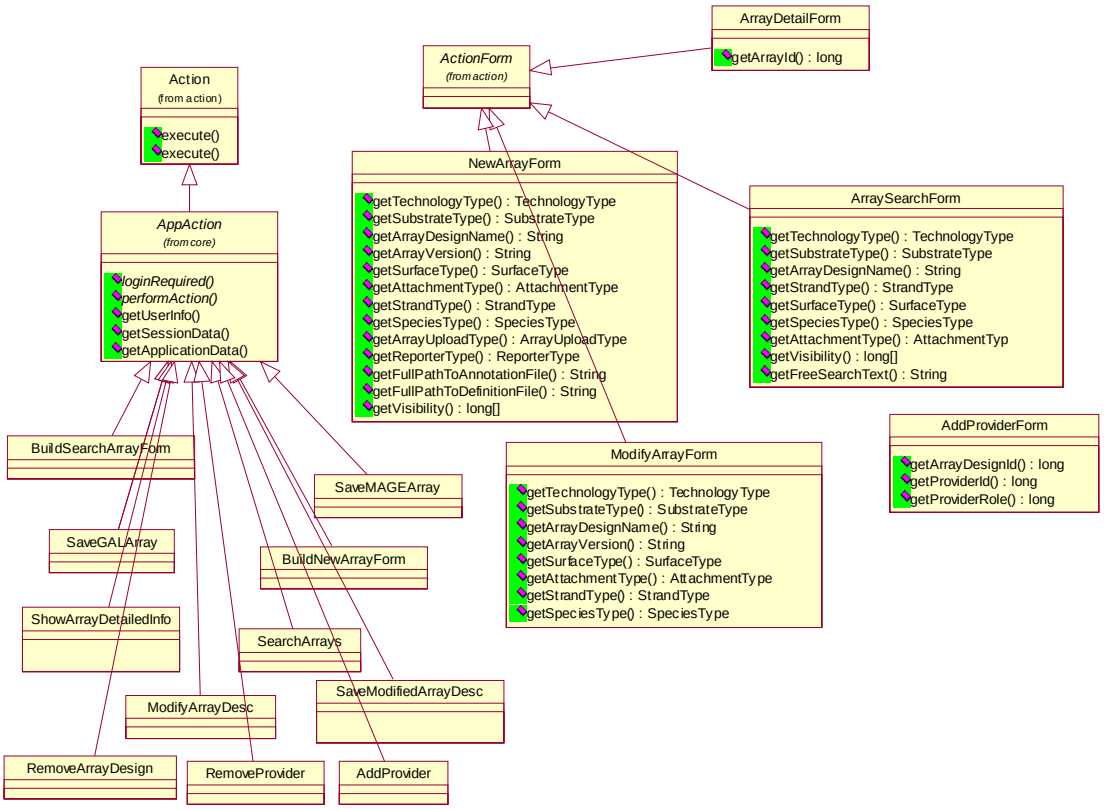
207. BuildSaveNewBasicExperimentFactorFormAction.peroformAction



208. Logical View::gov::nih::nci::caarray::ui::bioMaterial



209. Logical View::gov::nih::nci::caarray::ui::arraydesign


ArrayDetailForm
getArrayId() : long
ActionForm
(from action)
Action
(from action)
execute()
execute()
NewArrayForm
getTechnologyType() : TechnologyType
getSubstrateType() : SubstrateType
getArrayDesignName() : String
getArrayVersion() : String
getSurfaceType() : SurfaceType
getAttachmentType() : AttachmentType
getStrandType() : StrandType
getSpeciesType() : SpeciesType
getArrayUploadType() : ArrayUploadType
getReporterType() : ReporterType
getFullPathToAnnotationFile() : String
getFullPathToDefinitionFile() : String
getVisibility() : long[]
ArraySearchForm
getTechnologyType() : TechnologyType
getSubstrateType() : SubstrateType
getArrayDesignName() : String
getStrandType() : StrandType
getSurfaceType() : SurfaceType
getSpeciesType() : SpeciesType
getAttachmentType() : AttachmentTyp
getVisibility() : long[]
getFreeSearchText() : String
AppAction
(from core)
loginRequired()
performAction()
getUserInfo()
getSessionData()
getApplicationData()
AddProviderForm
getArrayDesignId() : long
getProviderId() : long
getProviderRole() : long
BuildSearchArrayForm
SaveMAGEArray
ModifyArrayForm
getTechnologyType() : TechnologyType
getSubstrateType() : SubstrateType
getArrayDesignName() : String
getArrayVersion() : String
getSurfaceType() : SurfaceType
getAttachmentType() : AttachmentType
getStrandType() : StrandType
getSpeciesType() : SpeciesType
SaveGALArray
BuildNewArrayForm
ShowArrayDetailedInfo
SearchArrays
SaveModifiedArrayDesc
ModifyArrayDesc
RemoveArrayDesign
RemoveProvider
AddProvider

209.1 BuildNewArrayForm

209.2 NewArrayForm

209.3 SaveGALArray

209.4 SaveMAGEArray

209.5 BuildSearchArrayForm

209.6 ArraySearchForm

209.7 SearchArrays

209.8 ShowArrayDetailedInfo

209.9 ArrayDetailForm

209.10 ModifyArrayDesc

209.11 ModifyArrayForm

209.12 SaveModifiedArrayDesc

209.13 RemoveArrayDesign

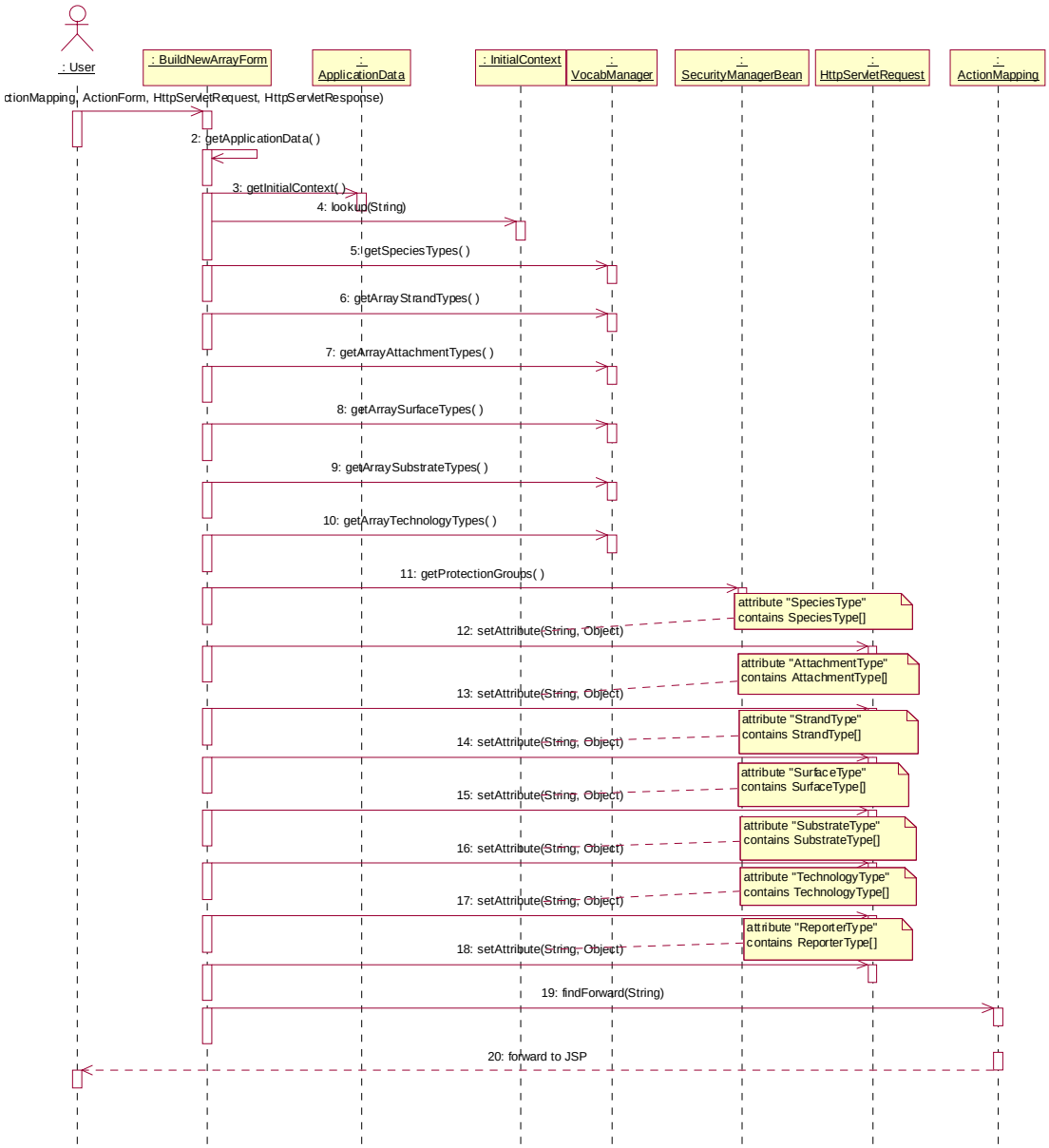
209.14 RemoveProvider

209.15 AddProvider

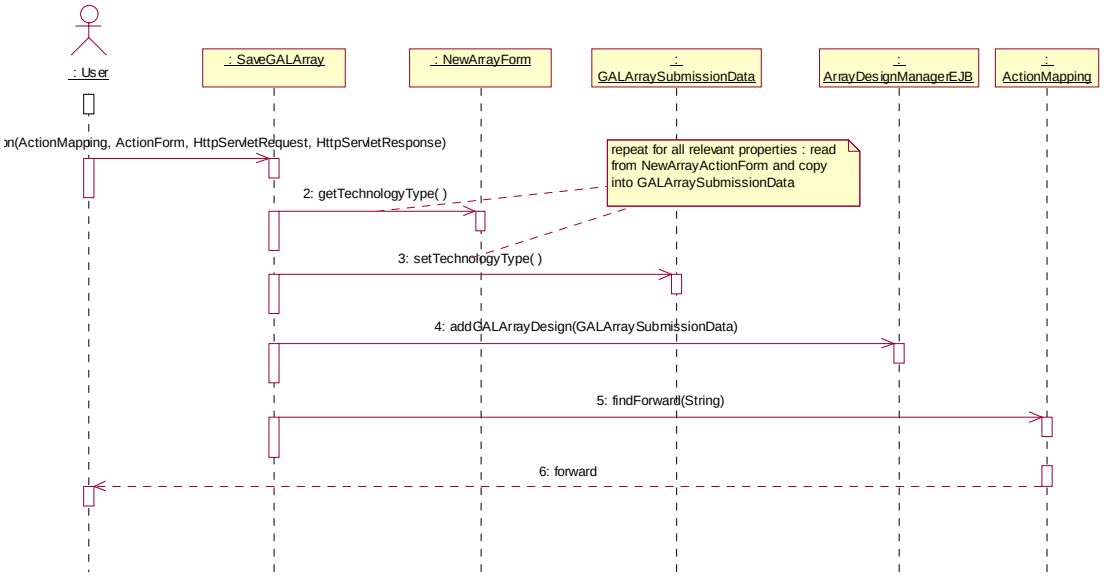
209.16 AddProviderForm

209.17

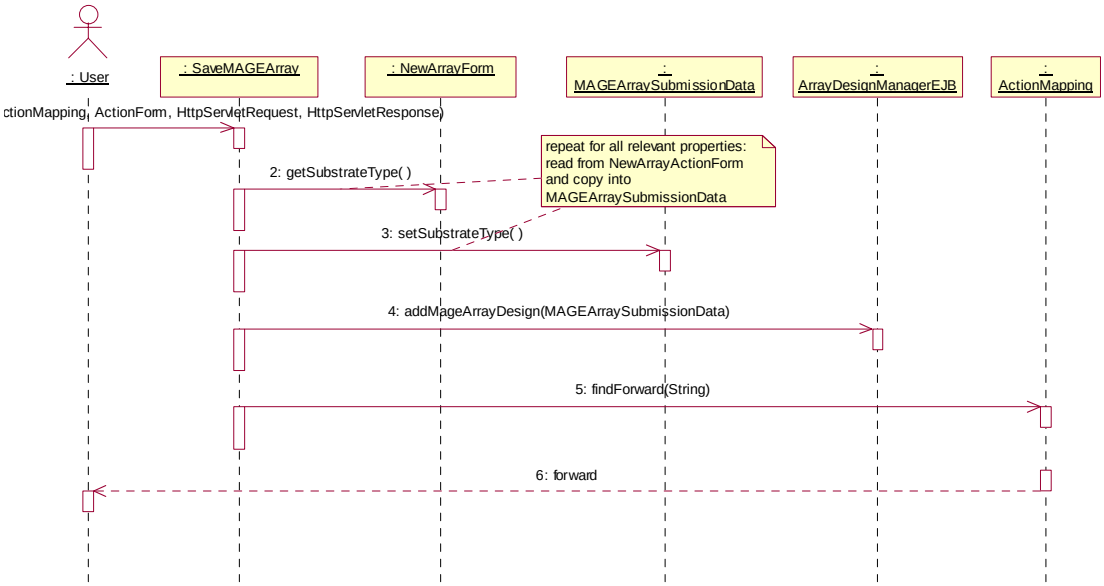
210. ArrayDesignUI.buildNewArrayForm



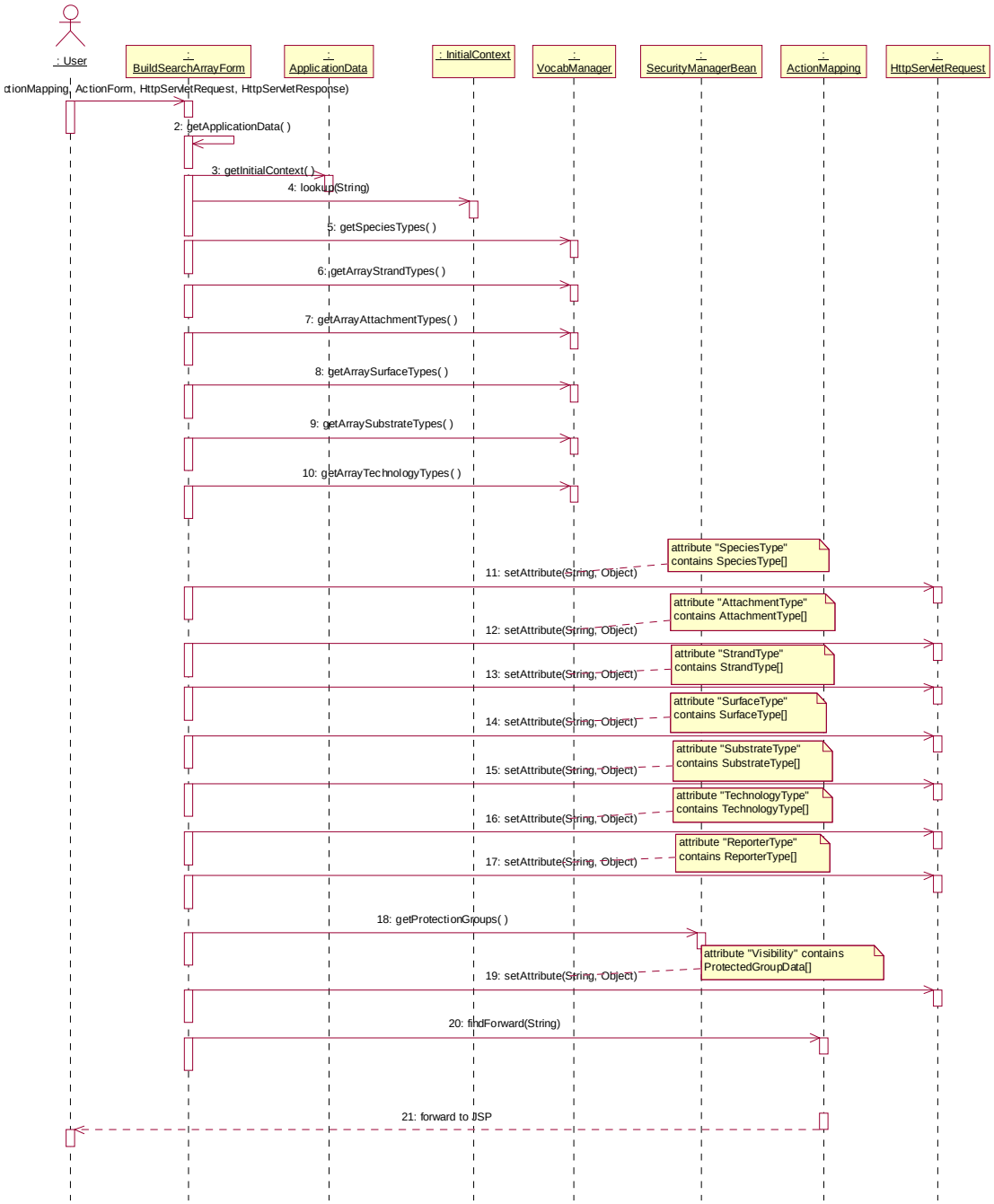
211. ArrayDesignUI.saveGALArray



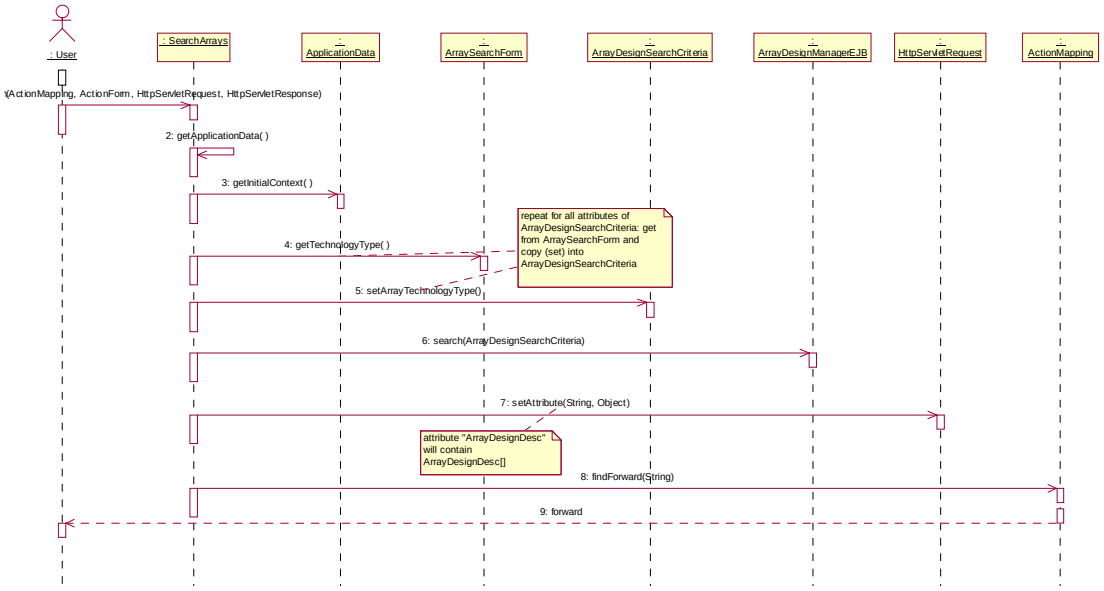
212. ArrayDesignUI.saveMAGEArray



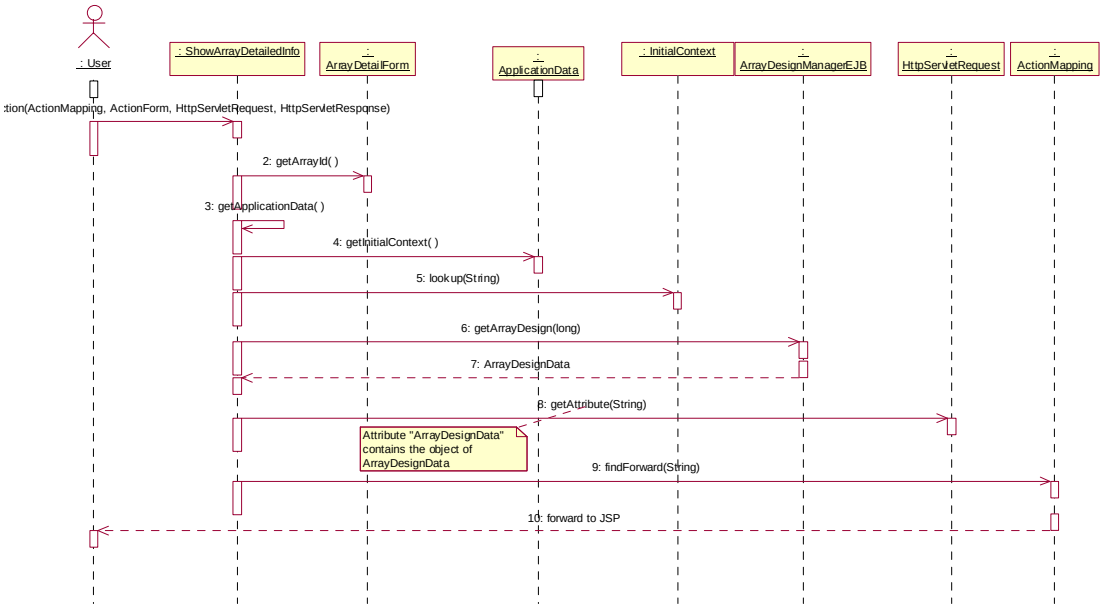
213. ArrayDesignUI.buildSearchArrayForm



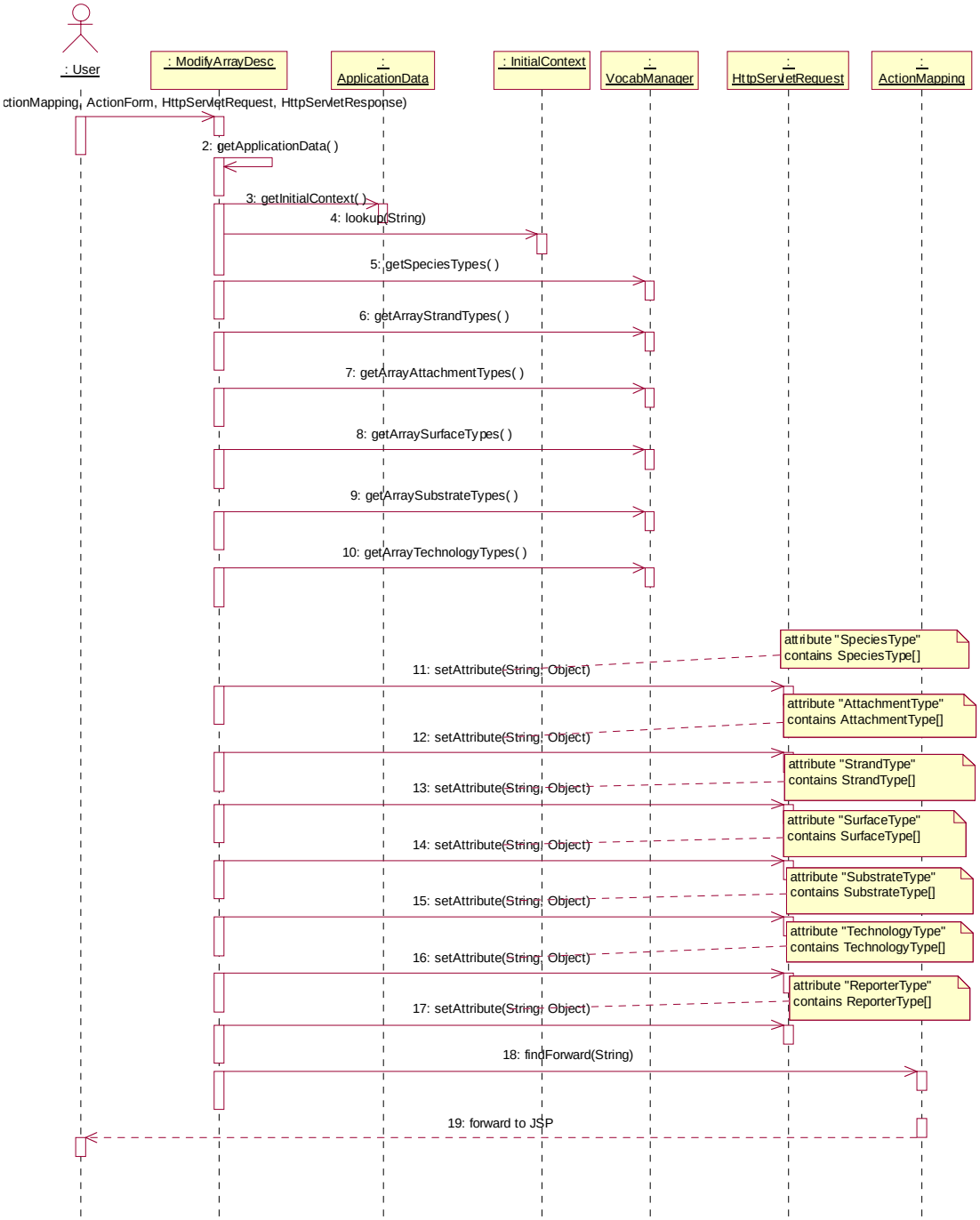
214. ArrayDesignUI.searchArrays



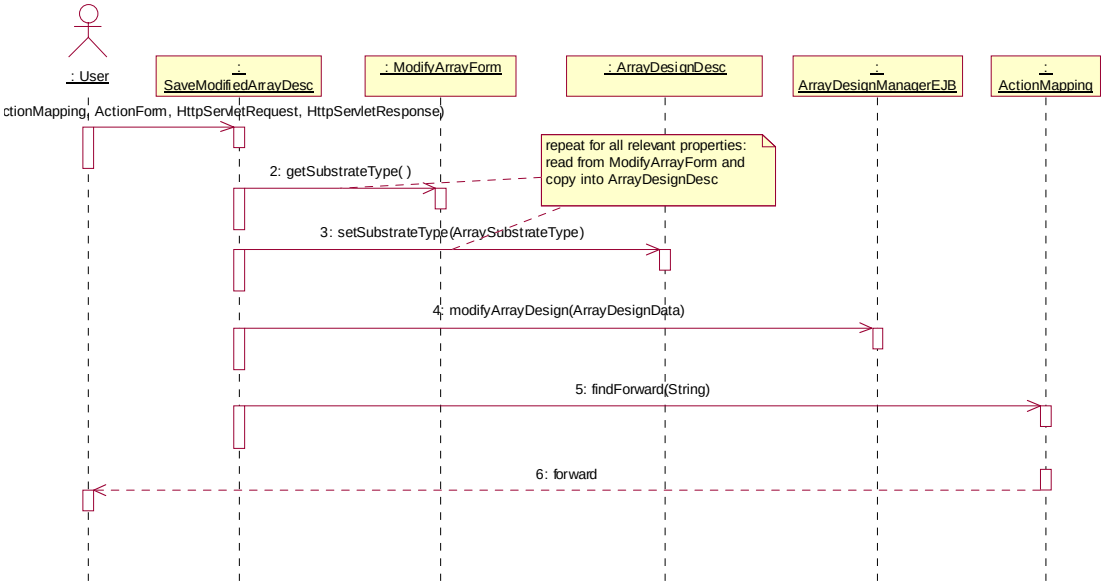
215. ArrayDesignUI.showArrayDetailedInfo



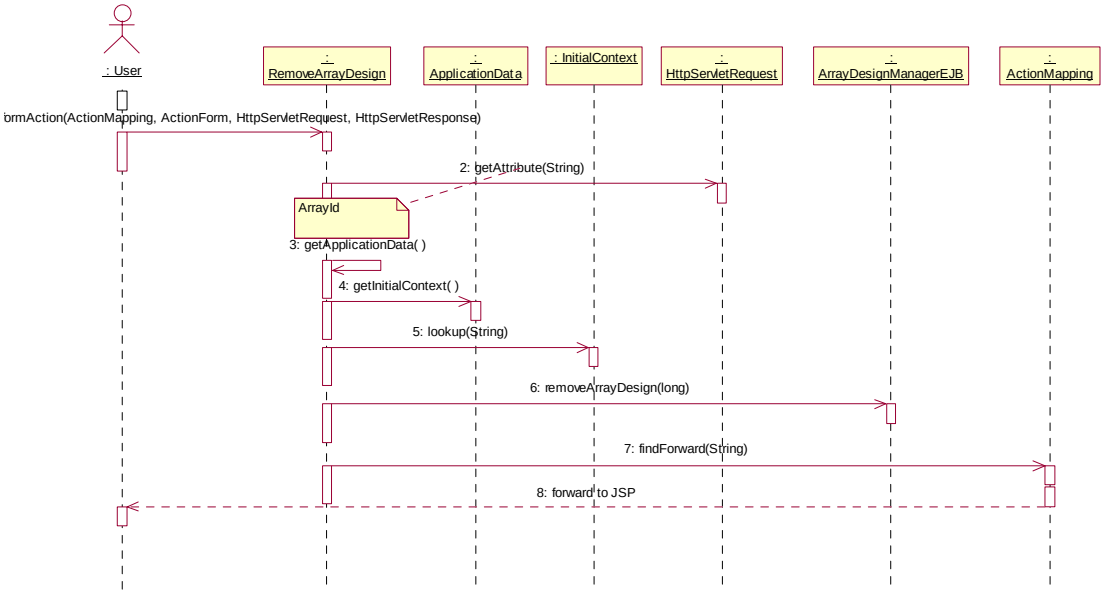
216. ArrayDesignUI.buildModifyArrayDescForm



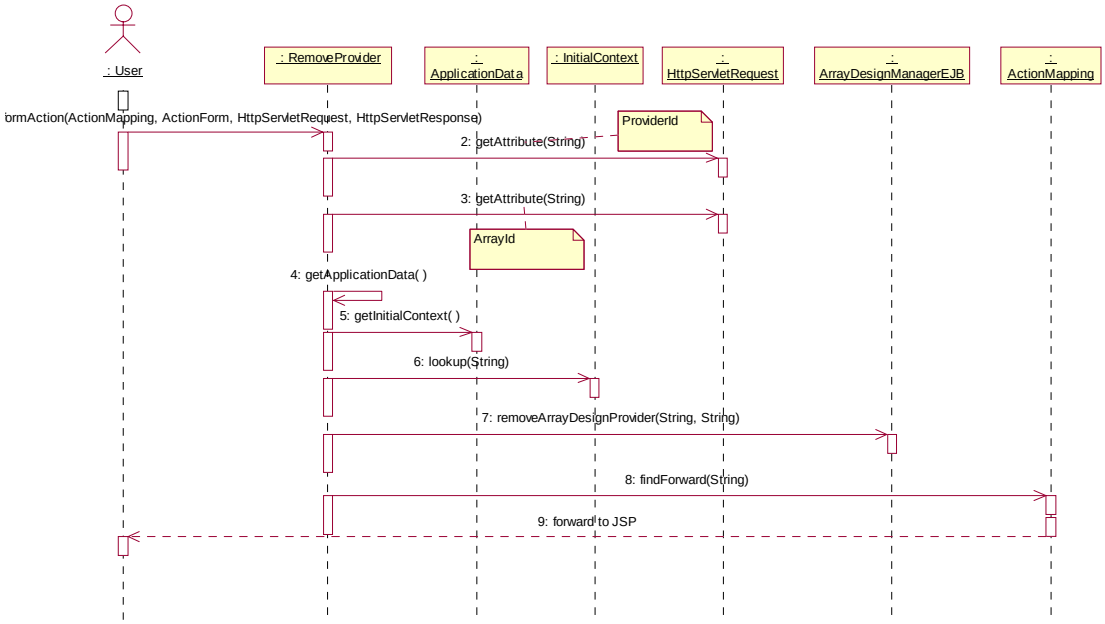
217. ArrayDesignUI.saveModifyArrayDesc



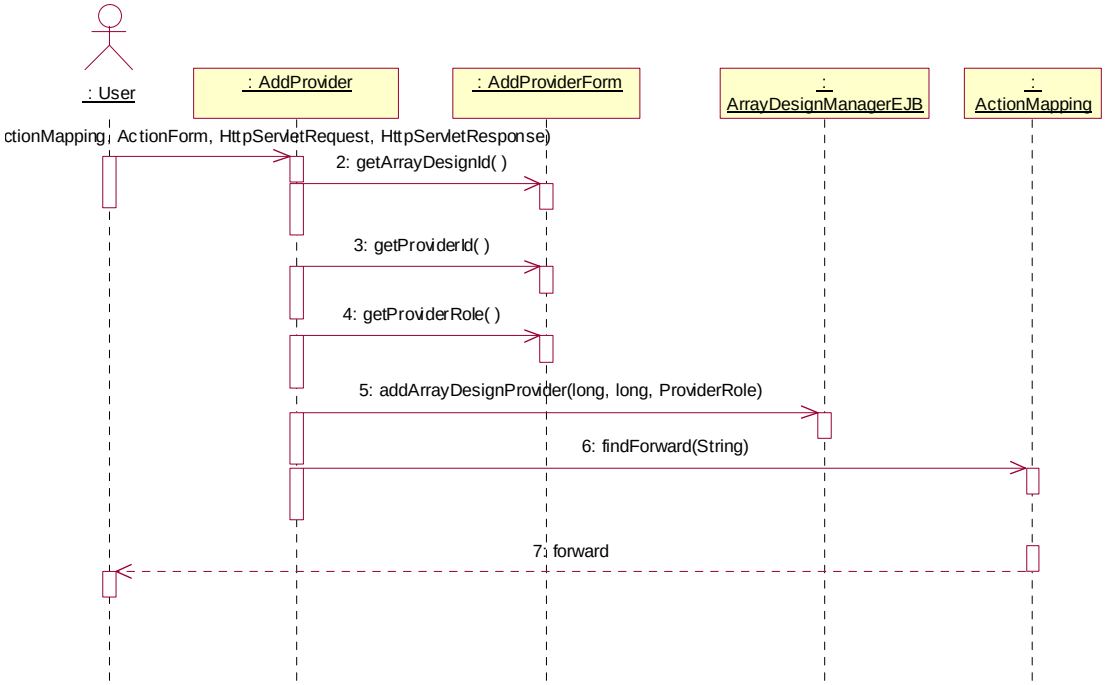
218. ArrayDesignUI.removeArrayDesign



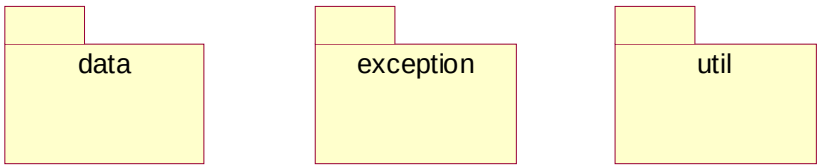
219. ArrayDesignUI.removeProvider



220. ArrayDesignUI.addProvider

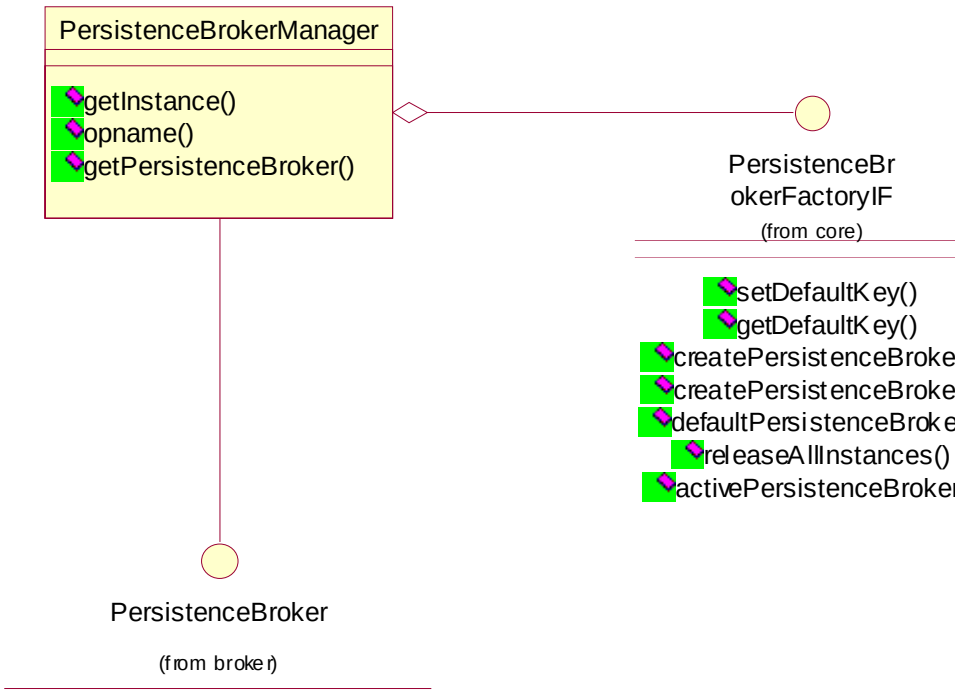


221. Logical View::gov::nih::nci::caarray::common



222. Logical View::gov::nih::nci::caarray::common::util

223. Logical View::gov::nih::nci::caarray::common::util::db



223.1 PersistenceBrokerManager

223.2

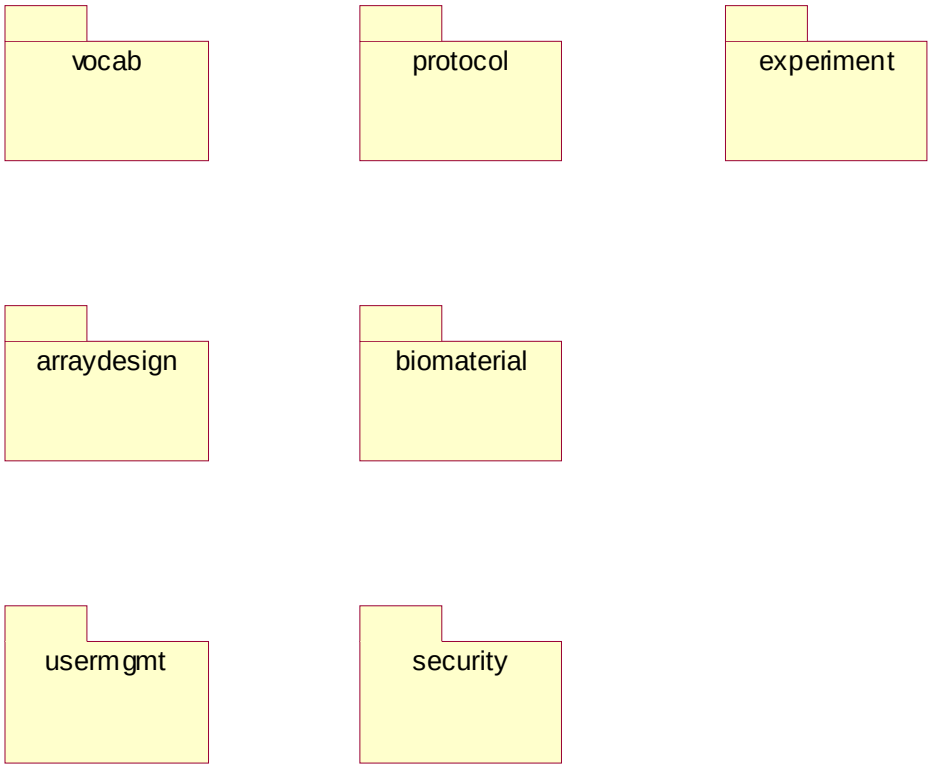
224. Logical View::gov::nih::nci::caarray::common::util::MageConversion

224.1 Mapper

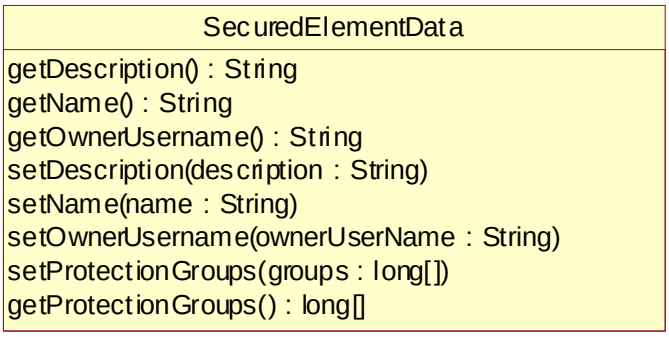
224.2 Mapinfo

224.3

225. Logical View::gov::nih::nci::caarray::common::data



226. Logical View::gov::nih::nci::caarray::common::data::security



226.1 SecuredElementThis class contains information about a secured element in the system. Each SecuredElement has an owner and a collection of protection groups.

226.2 UserSecurityDataThis class contains information about a user entry in the security database tables.

226.3 ProtectionGroupDataA ProtectionGroup is the mechanism used to determine which users can access which SecuredElements. A user must have been granted access to at least one protection group which is protecting a SecuredElement in order to gain access to the element.

226.4 ProtectionGroupTypeThis object specifies the type of a protection group.

226.5 SecuredElementDataThis class contains the minimum information necessary to create a new SecuredElement.

226.6 RoleDataA role is the mechanism used to determine what functionality in the system a user may access. Each role assignment may be limited by attaching protection groups to it. If there are no protection groups attached, then the role assignment is global and pertains to all data.

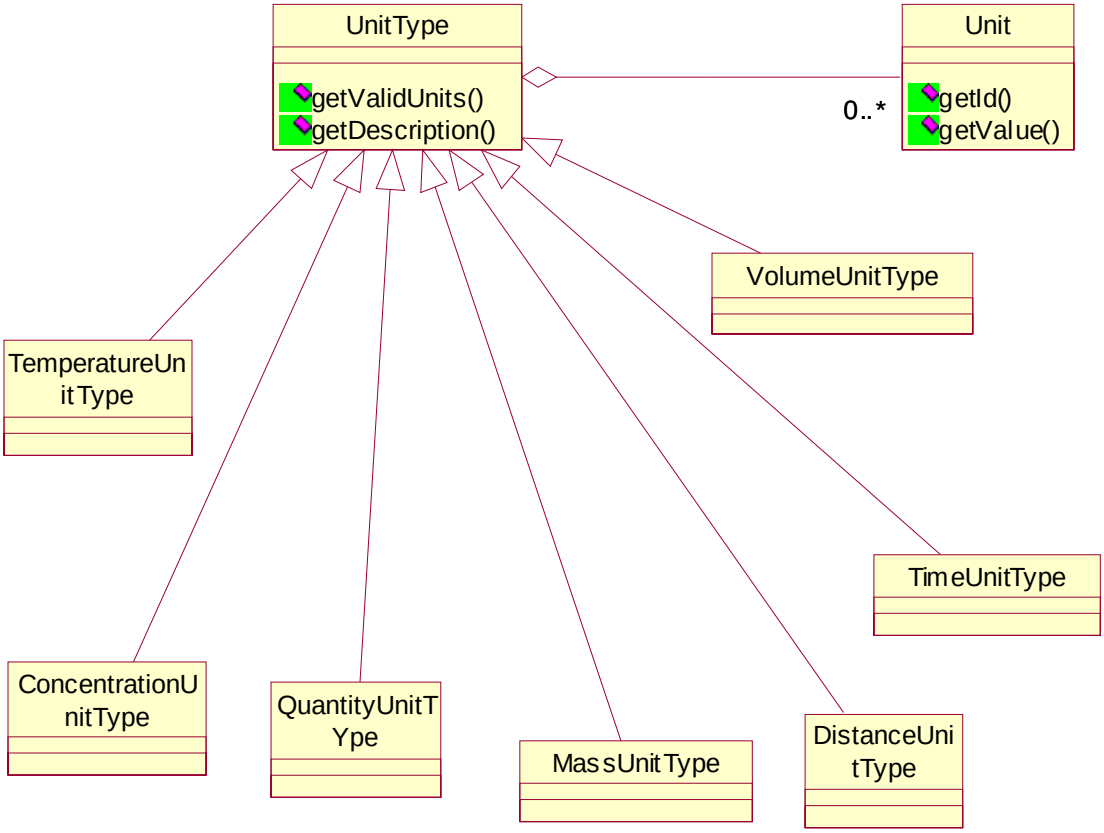
If there are protection groups attached, then the role assignment pertains only to secured elements protected by the specified protection groups.

226.7 UserRoleDataThis class is used to retrieve the collection of roles a user has been assigned from the database.

226.8

227. Logical View::gov::nih::nci::caarray::common::data::vocab

228. Logical View::gov::nih::nci::caarray::common::data::vocab::general



228.1 Species

228.2 VocabElement

228.3 UnitThis class is used to model the units used for a particular measurement. Examples are mm, cm, ml, etc.

228.4 DataType

228.5 VocabCategory

228.6 UnitType

228.7 ConcentrationUnitType

228.8 MassUnitType

228.9 TimeUnitType

228.10 DistanceUnitType

228.11 QuantityUnitTYpe

228.12 TemperatureUnitType

228.13 VolumeUnitType

228.14

229. Logical View::gov::nih::nci::caarray::common::data::vocab::arraydesign

229.1 ArrayAttachmentType

229.2 ArrayStrandType

229.3 ArraySubstrateType

229.4 ArrayTechnologyType

229.5 ArraySurfaceType

229.6 ArrayUploadType

229.7

230. Logical View::gov::nih::nci::caarray::common::data::vocab::protocol

230.1 ProtocolType

230.2 ParameterType

230.3 HardwareType

230.4 SoftwareType

230.5

231. Logical View::gov::nih::nci::caarray::common::data::vocab::biomaterial

231.1 BiosourceType

231.2 MaterialType

231.3

232. Logical View::gov::nih::nci::caarray::common::data::vocab::experiment

232.1 ExperimentType

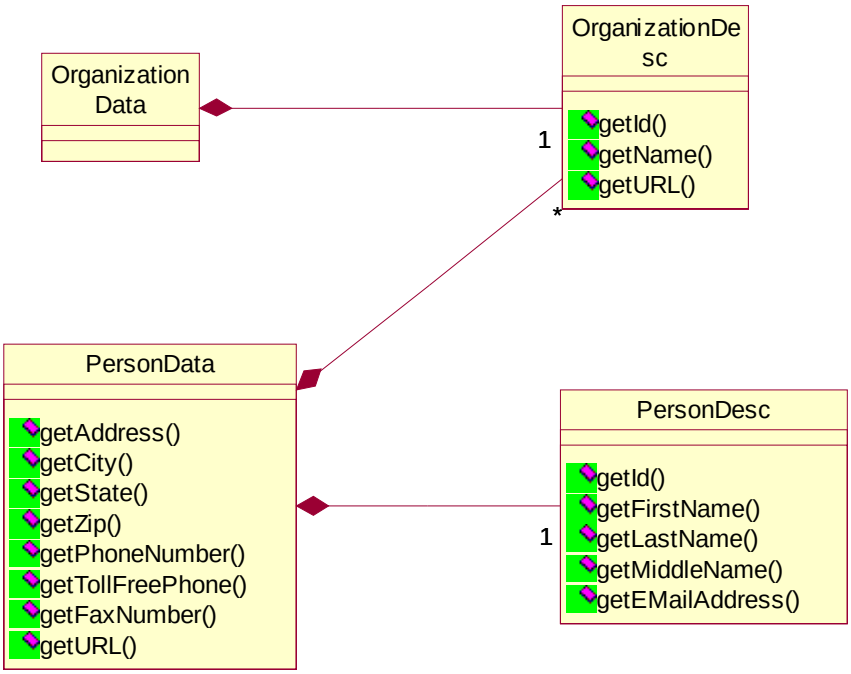
232.2 QualityControlType

232.3 FactorType

232.4 Scale

232.5

233. Logical View::gov::nih::nci::caarray::common::data::usermgmt



233.1 PersonDesc

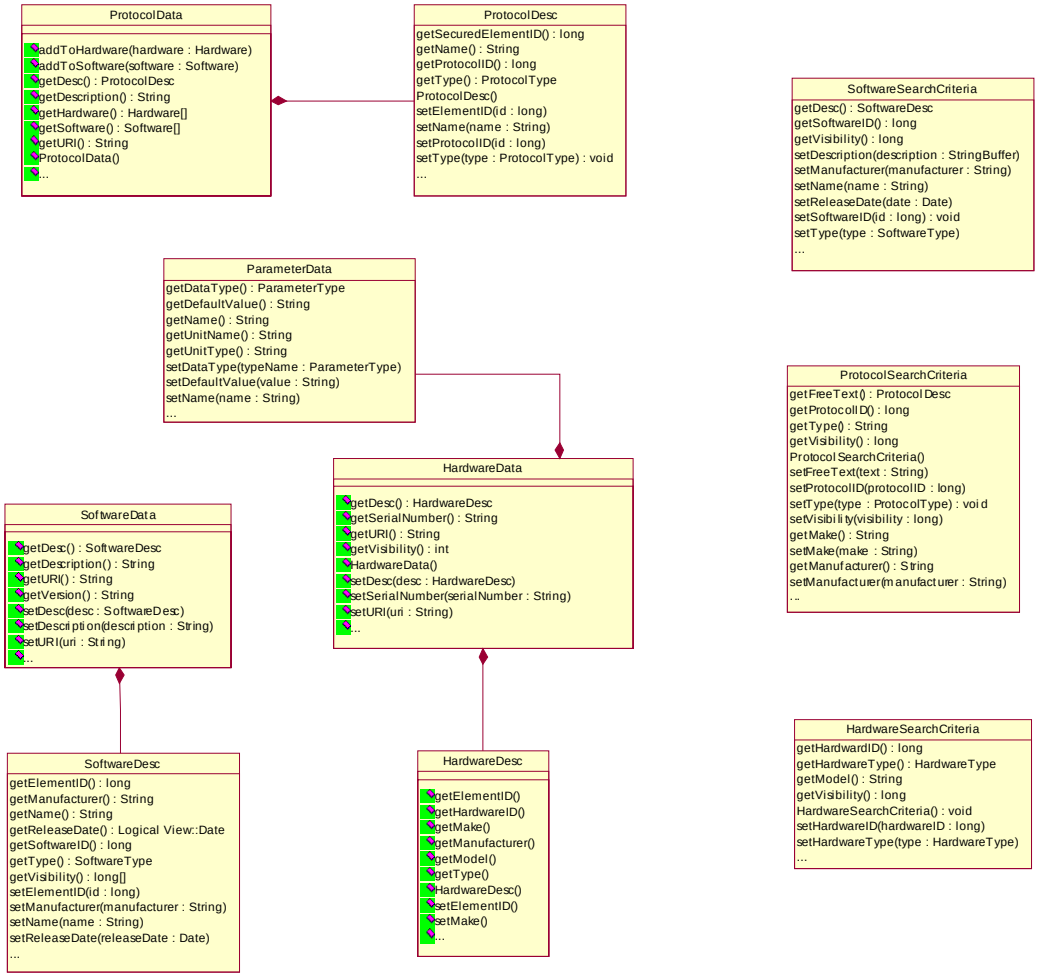
233.2 PersonData

233.3 OrganizationDesc

233.4 OrganizationData

233.5

234. Logical View::gov::nih::nci::caarray::common::data::protocol



234.1 ProtocolSearchCriteria

234.2 ProtocolDesc

234.3 SoftwareSearchCriteria

234.4 SoftwareDesc

234.5 HardwareDesc

234.6 HardwareSearchCriteria

234.7 SoftwareData

234.8 ProtocolData

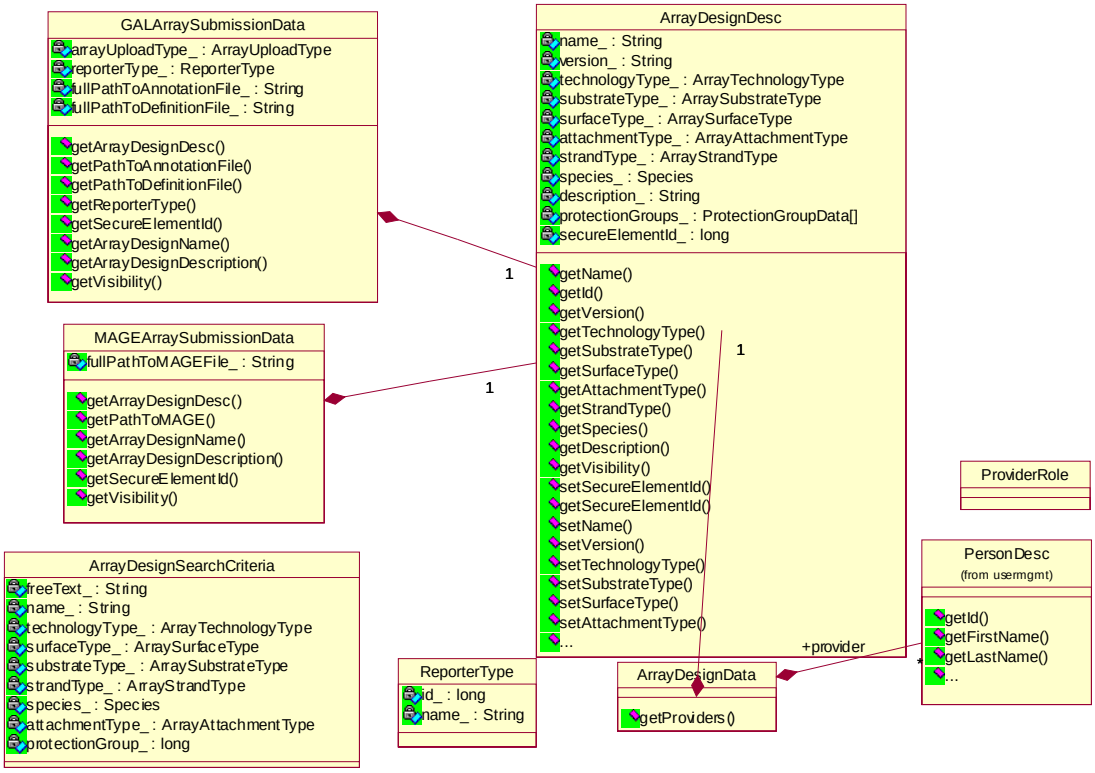
234.9 HardwareData

234.10 ParameterData

234.11 ProtocolElementData

234.12

235. Logical View::gov::nih::nci::caarray::common::data::arraydesign



235.1 ArrayDesignData

235.2 ArrayDesignDesc

235.3 ArrayDesignProvider

235.4 ArrayDesignSearchCriteria

235.5 ProviderRole

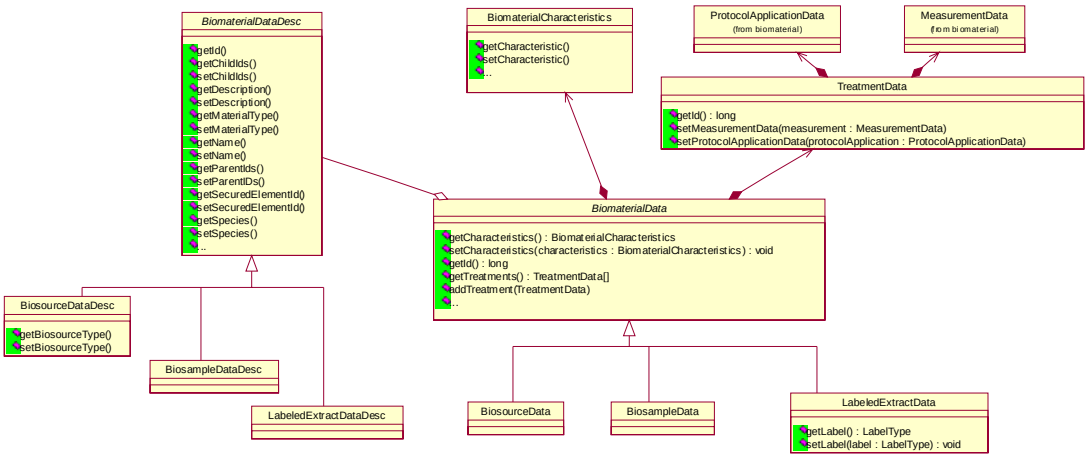
235.6 ReporterType

235.7 MAGEArraySubmissionData

235.8 GALArraySubmissionData

235.9

236. Logical View::gov::nih::nci::caarray::common::data::biomaterial



236.1 BiosourceData

236.2 BiomaterialData

236.3 Biosource

236.4 BiomaterialDataDesc

236.5 BiomaterialCharacteristics

236.6 BiosourceDataDesc

236.7 BiosampleDataDesc

236.8 LabeledExtractDataDesc

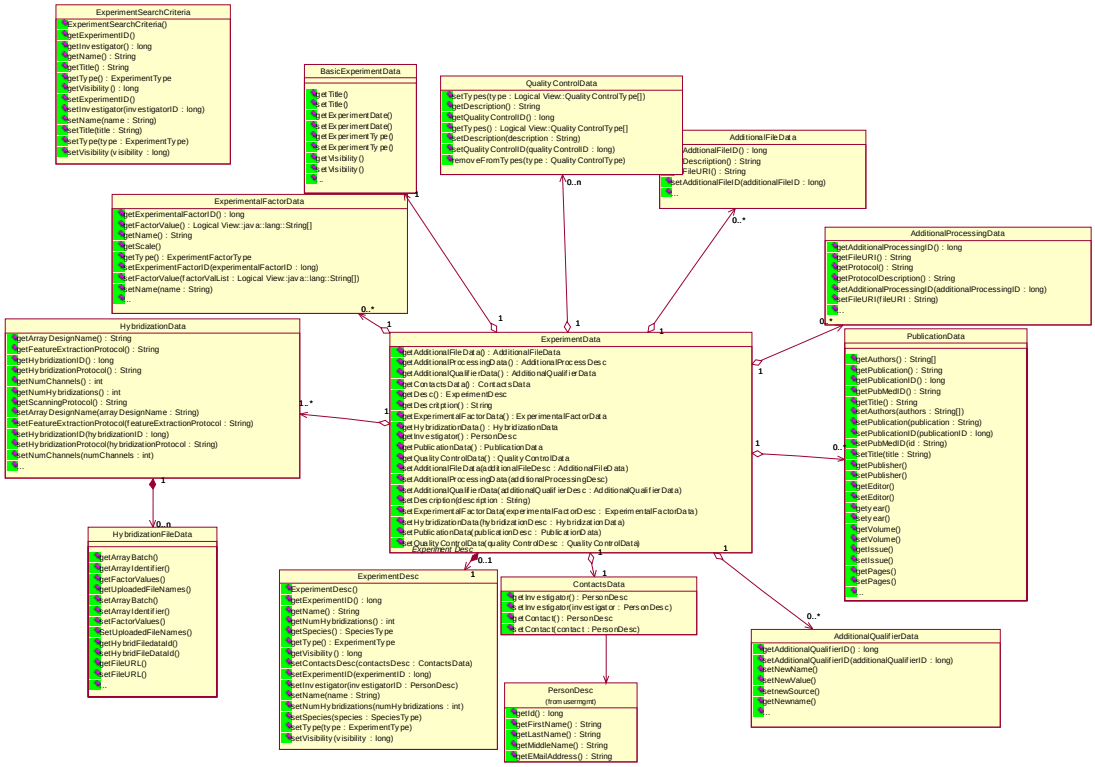
236.9 BiosampleData

236.10 LabeledExtractData

236.11 TreatmentData

236.12

237. Logical View::gov::nih::nci::caarray::common::data::experiment



237.1 ExperimentData

237.2 ExperimentSearchCriteria

237.3 ExperimentDesc

237.4 ExperimentalFactorData

237.5 PublicationData

237.6 QualityControlData

237.7 AdditionalQualifierData

237.8 HybridizationData

237.9 AdditionalProcessingData

237.10 AdditionalFileData

237.11 ContactsData

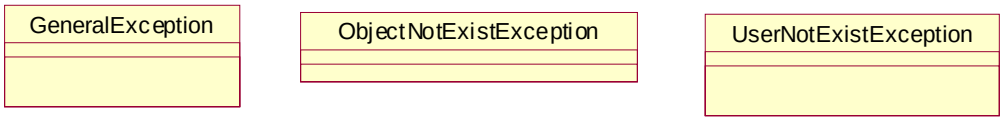
237.12 BasicExperimentData

237.13 FtpInfo

237.14 HybridizationFileData

237.15

238. Logical View::gov::nih::nci::caarray::common::exception



238.1 GeneralException

238.2 NewClass

238.3 ObjectNotExistException

238.4 UserNotExistException

238.5 AccessDeniedException

238.6

239. Logical View::sun

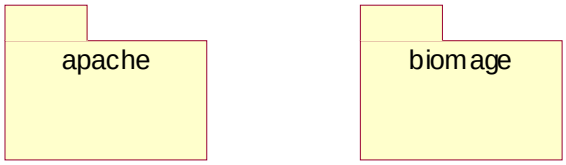
240. Logical View::sun::security

241. Logical View::sun::security::util

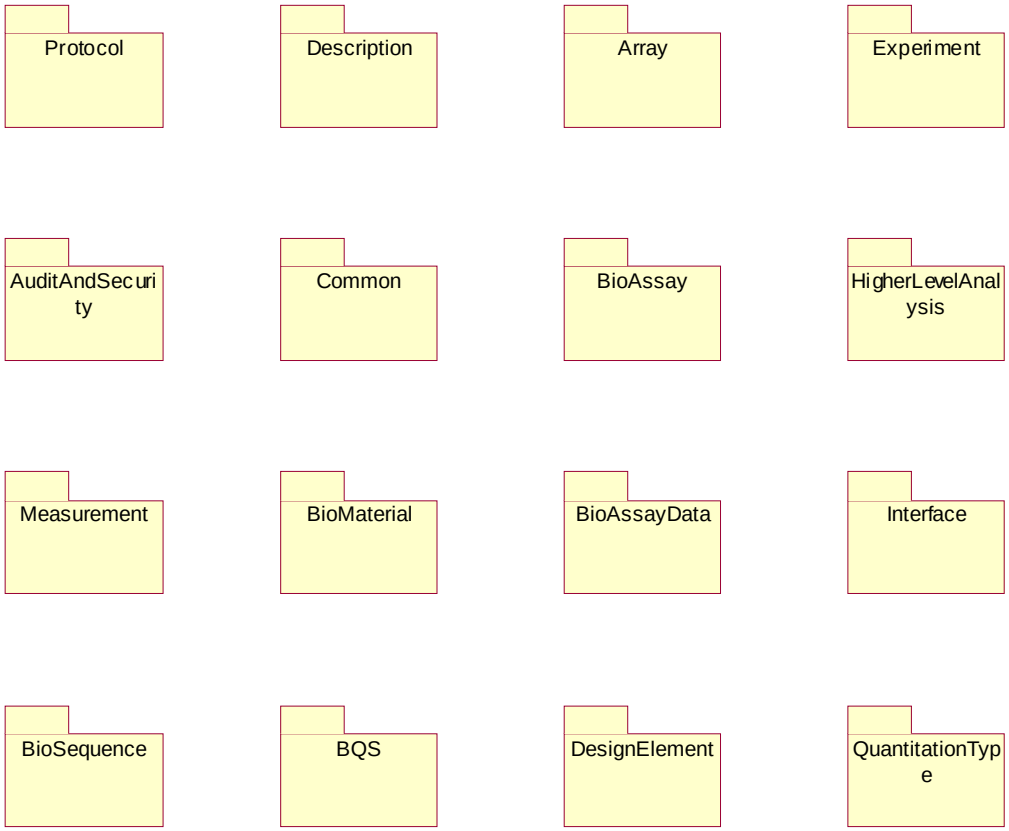
241.1 Debug

241.2

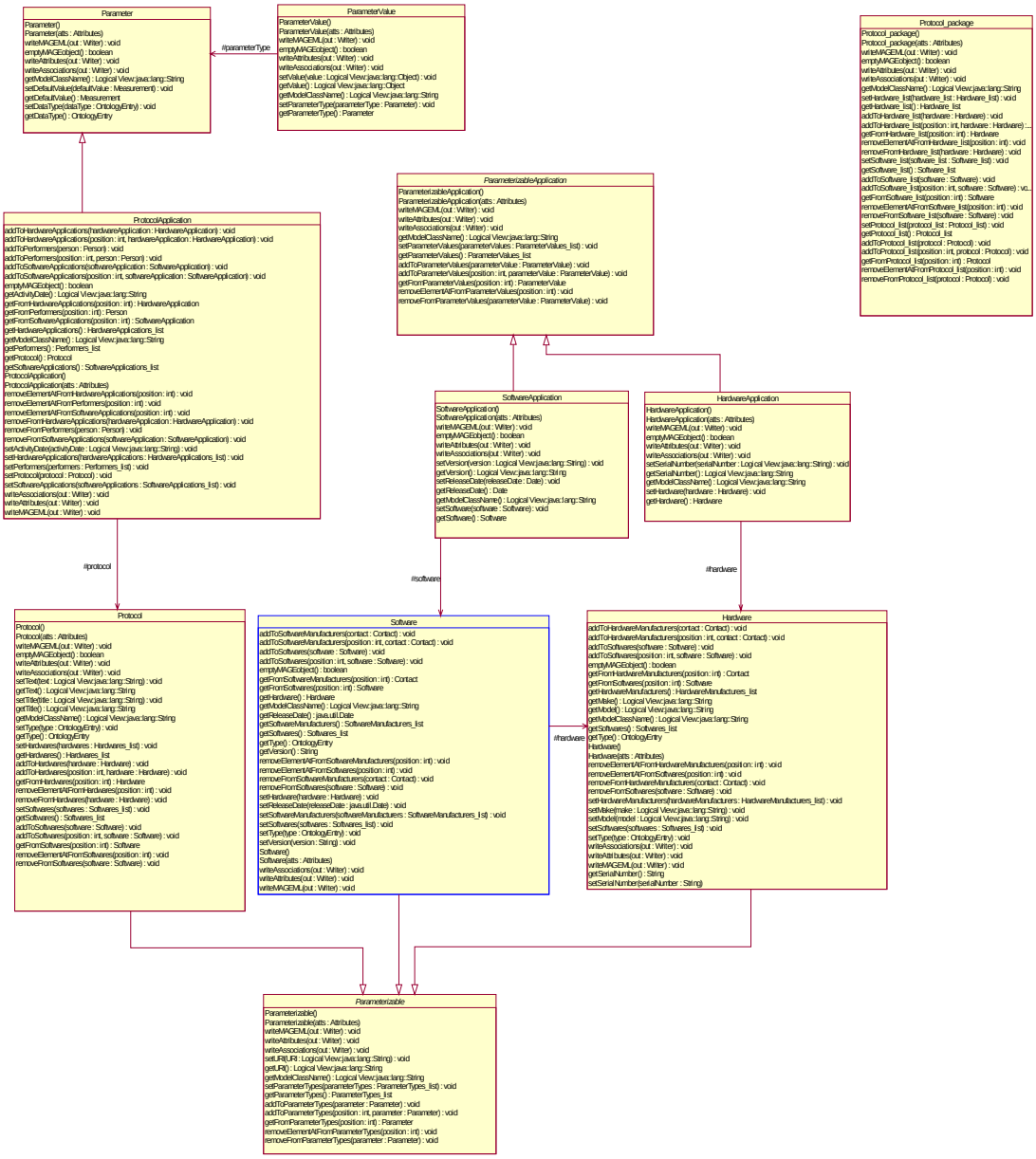
242. Logical View::org



243. Logical View::org::biomage



244. Logical View::org::biomage::Protocol



244.1 ProtocolApplicationThe use of a protocol with the requisite Parameters and

244.2 ParameterValues.

244.3 Protocol_packageProvides a relatively immutable class, Protocol, that can describe a

244.4 generic laboratory procedure or analysis algorithm, for example, and an

244.5 instance class, ProtocolApplication, which can describe the actual

244.6 application of a protocol. The ProtocolApplication provides values for

244.7 the replaceable parameters of the Protocol and, through the Description

244.8 association of Describable, any variation from the Protocol.

244.9 HardwareHardware represents the hardware used. Examples of Hardware include:

244.10 computers, scanners, wash stations etc...

244.11 HardwareApplicationThe use of a piece of hardware with the requisite Parameters and

244.12 ParameterValues.

244.13 ParameterA Parameter is a replaceable value in a Parameterizable class.

244.14 Examples of Parameters include: scanning wavelength, laser power,

244.15 centrifuge speed, multiplicative errors, the number of input nodes to a

244.16 SOM, and PCR temperatures.

244.17 ParameterValueThe value of a Parameter.

244.18 ProtocolA Protocol is a parameterizable description of a method.

244.19 ProtocolApplication is used to specify the ParameterValues of it's

244.20 Protocol's Parameters.

244.21 SoftwareSoftware represents the software used. Examples of Software include:

244.22 feature extraction software, clustering software, etc...

244.23 SoftwareApplicationThe use of a piece of software with the requisite Parameters and

244.24 ParameterValues.

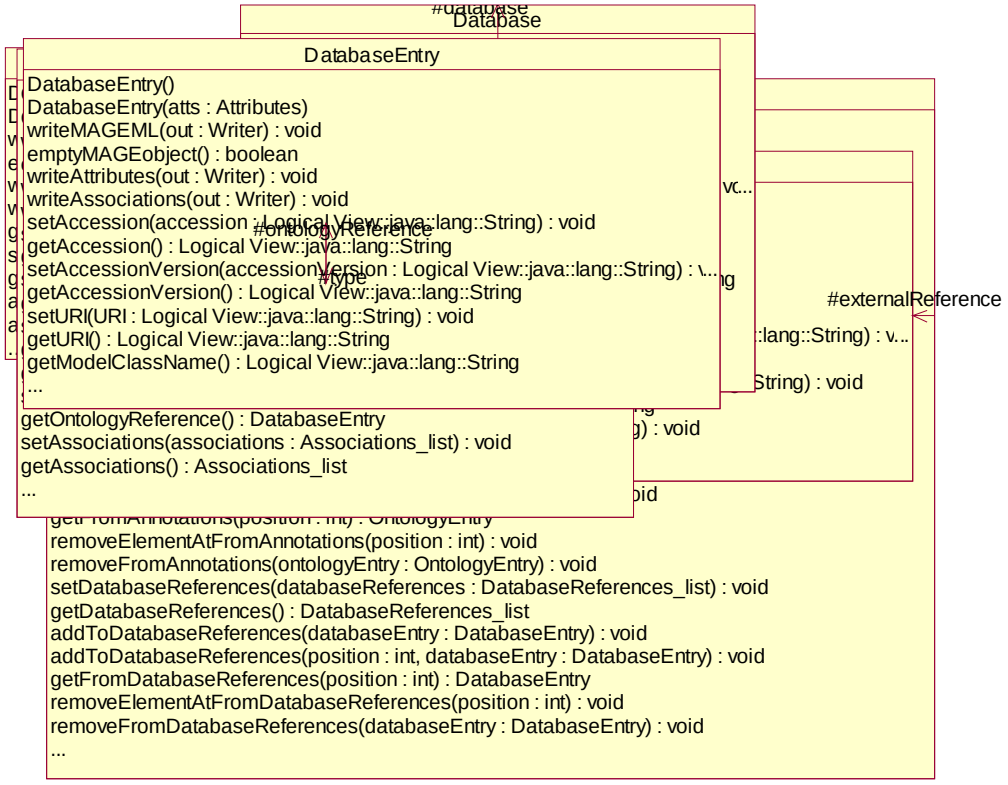
244.25 ParameterizableThe Parameterizable interface encapsulates the association of

244.26 Parameters with ParameterValues.

244.27 ParameterizableApplicationThe interface that is the use of a Parameterizable class.

244.28

245. Logical View::org::biomage::Description


#database
Database
DatabaseEntry
DatabaseEntry()
DatabaseEntry(atts : Attributes)
writeMAGEML(out : Writer) : void
emptyMAGEobject() : boolean
writeAttributes(out : Writer) : void
writeAssociations(out : Writer) : void
setAccession(accession : Logical View::java::lang::String) : void
getAccession() : Logical View::java::lang::String
setAccessionVersion(accessionVersion : Logical View::java::lang::String) : void
getAccessionVersion() : Logical View::java::lang::String
setURI(URI : Logical View::java::lang::String) : void
getURI() : Logical View::java::lang::String
getModelClassName() : Logical View::java::lang::String
...
#ontologyReference
#type
#externalReference
getOntologyReference() : DatabaseEntry
setAssociations(associations : Associations_list) : void
getAssociations() : Associations_list
...
removeElementAtFromAnnotations(position : int) : void
removeFromAnnotations(ontologyEntry : OntologyEntry) : void
setDatabaseReferences(databaseReferences : DatabaseReferences_list) : void
getDatabaseReferences() : DatabaseReferences_list
addToDatabaseReferences(databaseEntry : DatabaseEntry) : void
addToDatabaseReferences(position : int, databaseEntry : DatabaseEntry) : void
getFromDatabaseReferences(position : int) : DatabaseEntry
removeElementAtFromDatabaseReferences(position : int) : void
removeFromDatabaseReferences(databaseEntry : DatabaseEntry) : void
...

245.1 OntologyEntryA single entry from an ontology or a controlled vocabulary. For

245.2 instance, category could be 'species name', value could be 'homo

245.3 sapiens' and ontology would be taxonomy database, NCBI.

245.4 DatabaseEntryA reference to a record in a database.

245.5 DescriptionA free text description of an object.

245.6 Description_packageThe classes in this package allow a variety of references to third

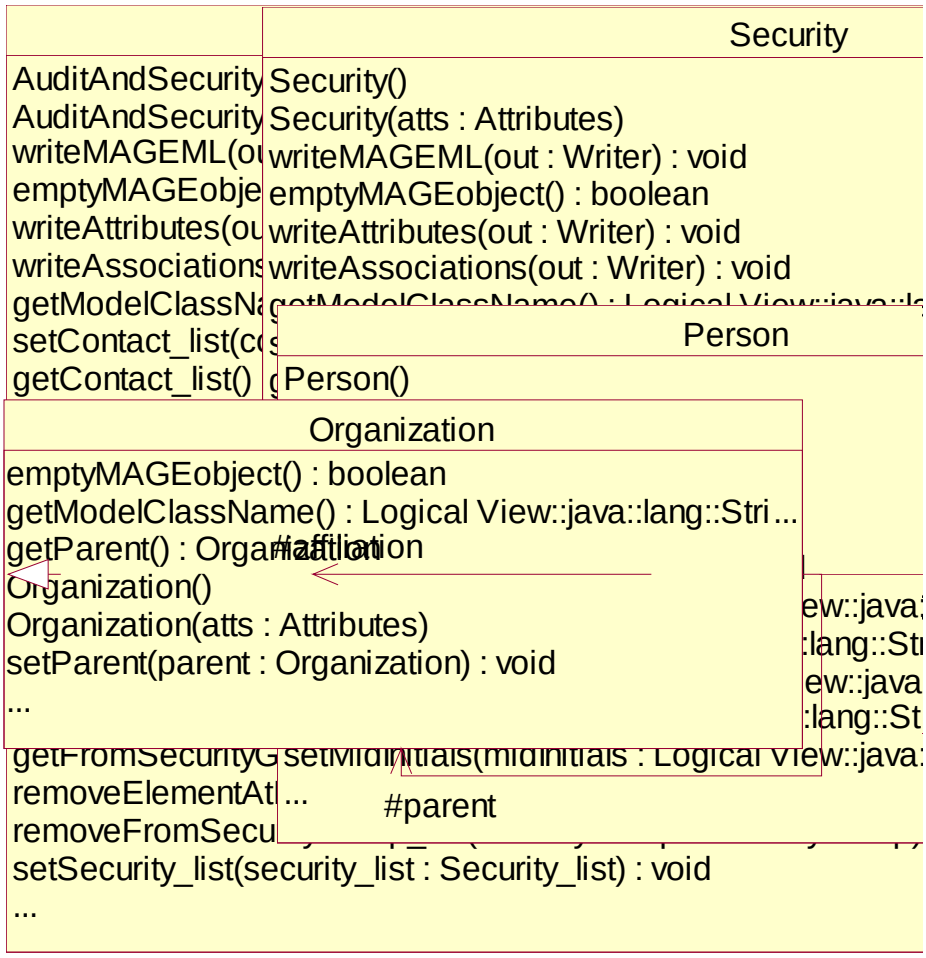
245.7 party annotation and direct annotation by the experimenter.

245.8 DatabaseAn address to a repository.

245.9 ExternalReferenceA reference to the originating source for the object.

245.10

246. Logical View::org::biomage::AuditAndSecurity


Security
Security()
Security(atts : Attributes)
writeMAGEML(out : Writer) : void
emptyMAGEobject() : boolean
writeAttributes(out : Writer) : void
writeAssociations(out : Writer) : void
Person
Person()
Organization
emptyMAGEobject() : boolean
getModelClassName() : Logical View::java::lang::Stri...
Organization()
Organization(atts : Attributes)
setParent(parent : Organization) : void
...
#affiliation
#parent
AuditAndSecurity
AuditAndSecurity
writeMAGEML(ou
emptyMAGEobje
writeAttributes(ou
writeAssociations
getModelClassNa
setContact_list(co
getContact_list()
getParent() : Orga
ew::java
:lang::Str
ew::java
:lang::St
getFromSecurityG
removeElementAt
removeFromSecu
setMidInitials(midInitials : Logical View::java
setSecurity_list(security_list : Security_list) : void
...

246.1 ContactA contact is either a person or an organization.

246.2 AuditTracks information on the contact that creates or modifies an object.

246.3 AuditAndSecurity_packageSpecifies classes that allow tracking of changes and information on

246.4 user permissions to view the data and annotation.

246.5 SecurityGroupGroups contacts together based on their security privileges.

246.6 SecurityPermission information for an object as to ownership, write and read

246.7 permissions.

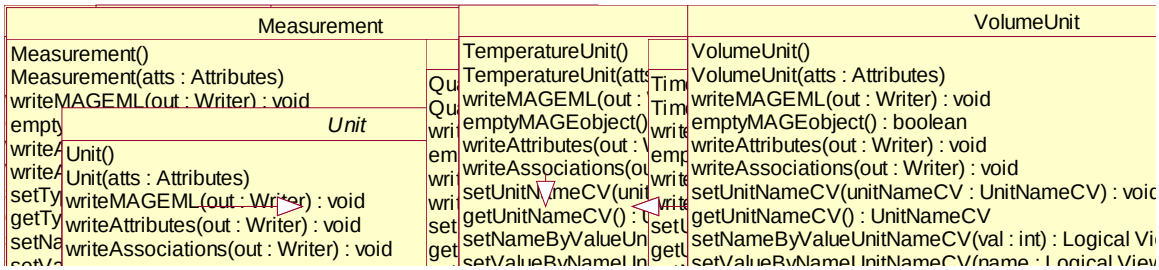
246.8 OrganizationOrganizations are entities like companies, universities, government

246.9 agencies for which the attributes are self describing.

246.10 PersonA person for which the attributes are self describing.

246.11

247. Logical View::org::biomage::Measurement



247.1 DistanceUnitDistance

247.2 MeasurementA Measurement is a quantity with a unit.

247.3 Measurement_packageThe classes of this package provide utility information on the

247.4 quantities of other classes to each other.

247.5 UnitThe unit is a strict enumeration of types.

247.6 ConcentrationUnitConcentration

247.7 MassUnitMass

247.8 QuantityUnitQuantity

247.9 TemperatureUnitTemperature

247.10 TimeUnitTime

247.11 VolumeUnitVolume

247.12

248. Logical View::org::biomage::Common

```
MA
MAGEJava()
MAGEJava(atts : Attributes)
writeMAGEML(out : Writer) : void
writeAttributes(out : Writer) : void
writeAssociations(out : Writer) : void
getClassWithModelName(modelName : Logica
getModelClassName() : Logical View::java::lang
setAuditAndSecurity_package(auditAndSecurit
getAuditAndSecurity_package() : AuditAndSec
setDescription_package(description_package
getDescription_package() : Description_packa
setMeasurement_package(measurement_pack
getMeasurement_package() : Measurement_pa
setBQS_package(bQS_package : BQS_packa
getBQS_package() : BQS_package
setBioEvent_package(bioEvent_package : Bio
getBioEvent_package() : BioEvent_package
setProtocol_package(protocol_package : Proto
getProtocol_package() : Protocol_package
setBioMaterial_package(bioMaterial_package
getBioMaterial_package() : BioMaterial_packa
setBioSequence_package(bioSequence_pack
getBioSequence_package() : BioSequence_pa
setDesignElement_package(designElement_pa
getDesignElement_package() : DesignElement
setArrayDesign_package(arrayDesign_packag
getArrayDesign_package() : ArrayDesign_pack
setArray_package(array_package : Array_pack
getArray_package() : Array_package
setBioAssay_package(bioAssay_package : Bio
getBioAssay_package() : BioAssay_package
...
```

248.1 IdentifiableAn Identifiable class is one that has an unambiguous reference within

248.2 the scope. It also has a potentially ambiguous name.

248.3 NameValueTypeA tuple designed to store data, keyed by a name and type.

248.4 ExtendableAbstract class that specifies for subclasses an association to

248.5 NameValueTypes. These can be used, for instance, to specify proprietary

248.6 properties and in-house processing hints.

248.7 DescribableAbstract class that allows subclasses to inherit the association to

248.8 Description, for detailed annotations such as Ontology entries and

248.9 Database references, the association to Audit, for tracking changes, and

248.10 the association to Security for indicating permissions.

248.11 MAGEException

248.12 MAGEJavaTop-level object that represents the model. Contains the packages.

248.13

249. Logical View::org::biomage::ArrayDesign

ZoneLayout
ZoneLayout()
ZoneLayout(atts : Attributes)
writeMAGEML(out : Writer) : void
emptyMAGEobject() : boolean
writeAttributes(out : Writer) : void
writeAssociations(out : Writer) : void
setNumFeaturesPerRow(numFeaturesPerRow : Logical V
getNumFeaturesPerRow() : Logical View::java::lang::Integ
setNumFeaturesPerCol(numFeaturesPerCol : Logical Vie
getNumFeaturesPerCol() : Logical View::java::lang::Intege
setSpacingBetweenRows(spacingBetweenRows : Logica
getSpacingBetweenRows() : Logical View::java::lang::Floa
setSpacingBetweenCols(spacingBetweenCols : Logical V
...
ArrayDesign_package()
ArrayDesign_package(atts : At
writeMAGEML(out : Writer) : vo
emptyMAGEobject() : boolean
writeAttributes(out : Writer) : vo
writeAssociations(out : Writer)
getModelClassName() : Logica
setReporterGroup_list(reporter
getReporterGroup_list() : Repo
addToReporterGroup_list(repo
addToReporterGroup_list(posi
getFromReporterGroup_list(po
removeElementAtFromReporte
removeFromReporterGroup_lis
setCompositeGroup_list(compositeGroup_lis
getCompositeGroup_list() : CompositeGroup
addToCompositeGroup_list(compositeGroup
addToCompositeGroup_list(position : int, con
getFromCompositeGroup_list(position : int) :
removeElementAtFromCompositeGroup_list(
removeFromCompositeGroup_list(composite
setArrayDesign_list(arrayDesign_list : ArrayD
...
setDistanceUnit(distanceUnit : DistanceUnit) : void
getDistanceUnit() : DistanceUnit
setZoneLayout(zoneLayout : ZoneLayout) : void
getZoneLayout() : ZoneLayout
setZoneLocations(zoneLocations : ZoneLocations_list) : void
getZoneLocations() : ZoneLocations_list
...
removeFromFeatureGroups(featureGroup : FeatureGroup) : void
setReporterGroups(reporterGroups : ReporterGroups_list) : void
getReporterGroups() : ReporterGroups_list
addToReporterGroups(reporterGroup : ReporterGroup) : void
addToReporterGroups(position : int, reporterGroup : ReporterGroup) : void
getFromReporterGroups(position : int) : ReporterGroup
removeElementAtFromReporterGroups(position : int) : void
removeFromReporterGroups(reporterGroup : ReporterGroup) : void
setCompositeGroups(compositeGroups : CompositeGroups_list) : void
getCompositeGroups() : CompositeGroups_list
addToCompositeGroups(compositeGroup : CompositeGroup) : void
addToCompositeGroups(position : int, compositeGroup : CompositeGroup) : void
getFromCompositeGroups(position : int) : CompositeGroup
removeElementAtFromCompositeGroups(position : int) : void
removeFromCompositeGroups(compositeGroup : CompositeGroup) : void
...

249.1 ArrayDesignDescribes the design of an gene expression layout. In some cases

249.2 this might be virtual and, for instance, represent the output from

249.3 analysis software at the composite level without reporters or features.

249.4 ZoneSpecifies the location of a zone on an array.

249.5 FeatureGroupA collection of like features.

249.6 ReporterGroupAllows specification of the type of Reporter Design Element.

249.7 CompositeGroupAllows specification of the type of Composite Design Element.

249.8 ArrayDesign_packageDescribes a microarray design that can be printed and then, in the

249.9 case of gene expression, hybridized. An array design consists of

249.10 several features (also called spots) in which reporter sequences are

249.11 placed. Many features may have the same reporter replicated and a

249.12 reporter may be specified in one or more array designs.

249.13 The nature of the reporter's biosequence placed on a spot will depend on

249.14 the technology. Two well-known technologies differ

249.15 significantly-spotter arrays draw material from a well and place a spot

249.16 on the array whereas in situ oligo arrays are created through the

249.17 synthesis of many, short (~20-100mer) nucleotide sequences onto the

249.18 features.

249.19 Reporters can be grouped together into CompositeSequences, typically

249.20 representing a gene or one or more splice variants in gene expression

249.21 experiments.

249.22 There are then two distinct ways that DesignElements are grouped. The

249.23 one described in the ArrayDesign package by FeatureGroup, ReporterGroup

249.24 and CompositeGroup is by technology type, that is, one might want to

249.25 segregate the controls to a Group and all the non-controls to another.

249.26 Or if PCR Product and Oligos are both used on an array they would likely

249.27 be in different groups. The grouping described in the DesignElement

249.28 package by the mappings relates the Features to the Reporter, the

249.29 Reporters to CompositeSequence, and at higher levels, CompositeSequences

249.30 to CompositeSequence.

249.31 DesignElementGroupThe DesignElementGroup holds information on either features,

249.32 reporters, or compositeSequences, particularly that information that is

249.33 common between all of the DesignElements contained.

249.34 PhysicalArrayDesignA design that is expected to be used to manufacture physical arrays.

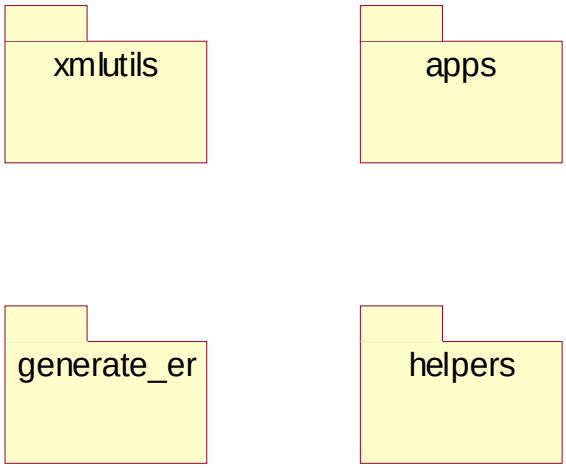
249.35 ZoneGroupSpecifies a repeating area on an array. This is useful for printing

249.36 when the same pattern is repeated in a regular fashion.

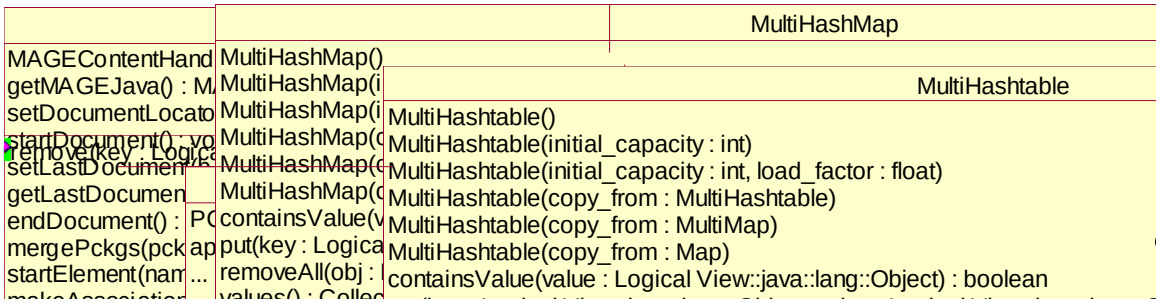
249.37 ZoneLayoutSpecifies the layout of features in a rectangular grid.

249.38

250. Logical View::org::biomage::tools



251. Logical View::org::biomage::tools::xmlutils



251.1 PCData@author kjellp

251.2 ImportExportTestDescription:

251.3 Demonstration class that imports a MAGE-ML document into

251.4 a MAGE-OM model and then writes it back out again.

251.5 MAGEContentHandlerDescription:

251.6 Content handler for MAGE.

251.7 PCDataImpl@author kjellp

251.8 MultiMapUtility interface for associating multiple values with a key.

251.9 MAGEReader@author Kjell Petersen

251.10 MultiHashMapUtility class for associating multiple values with a key in a Synchronized

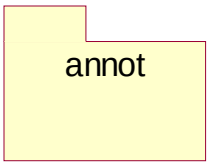
251.11 Map.

251.12 MultiHashtableUtility class for associating multiple values with a key in a Synchronized

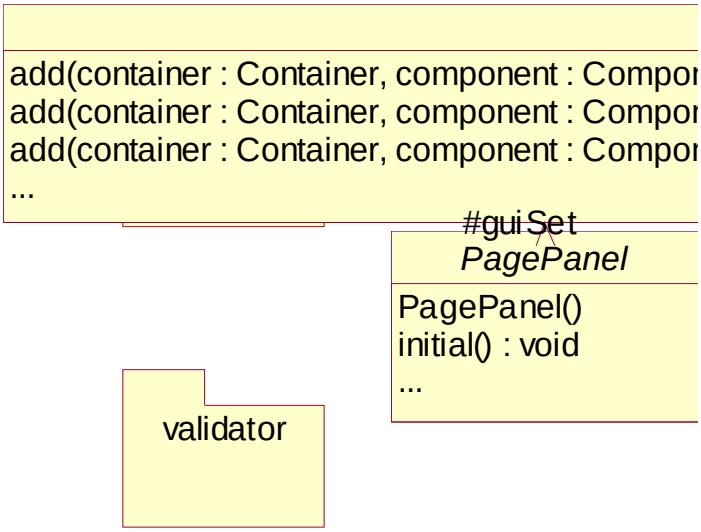
251.13 Map.

251.14

252. Logical View::org::biomage::tools::apps



253. Logical View::org::biomage::tools::apps::annot

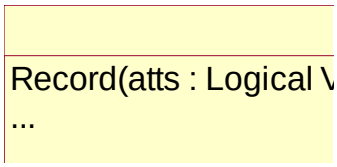


253.1 GuiSet

253.2 PagePanelimport org.biomage.tools.apps.annot.data.Record;

253.3

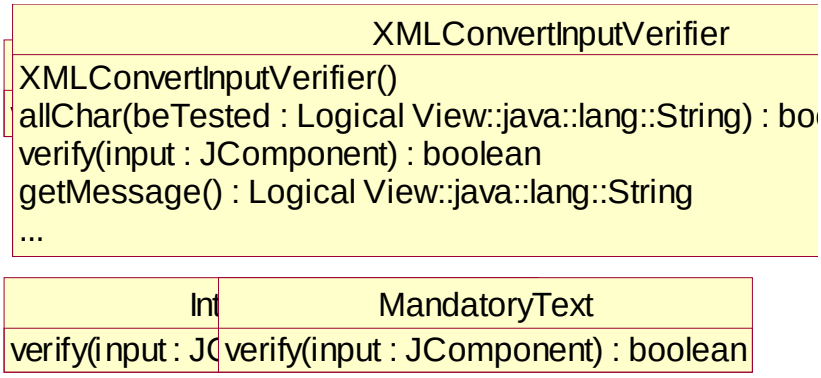
254. Logical View::org::biomage::tools::apps::annot::data



254.1 PageRecord

254.2

255. Logical View::org::biomage::tools::apps::annot::validator



255.1 FloatValidator

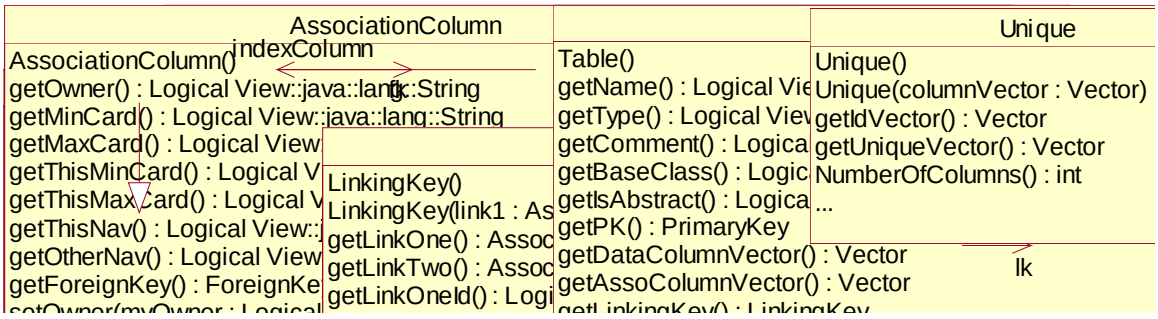
255.2 IntValidator

255.3 MandatoryText

255.4 XMLConvertInputVerifier

255.5

256. Logical View::org::biomage::tools::generate_er



256.1 AssociationColumn----------

256.2 <code>AssociationColumn</code> class represents an

256.3 association for an object. It is a column in the table

256.4 for that object.

256.5 ----------

256.6 DataColumn----------

256.7 <code>DataColumn</code> class represents a simple type data

256.8 for an object. It is a column in the table for

256.9 that object.

256.10 ----------

256.11 ForeignKey----------

256.12 <code>ForeignKey</code> class represents a foreign key

256.13 for a table. It is a column in the table for

256.14 that object, but it is included in the AssociationColumn.

256.15 Because in our XML file, each foreign key is a column

256.16 element and there is a foreign key element to specify

256.17 that this element is a foreign key.

256.18 ----------

256.19 Index----------

256.20 <code>PrimaryKey</code> class represents the primary

256.21 key for a table. It is a column in the table for

256.22 that object.

256.23 ----------

256.24 LinkingKey----------

256.25 <code>LinkingKey</code> class represents the linking

256.26 keys for a linking table. It is a column in the table for

256.27 that object.

256.28 ----------

256.29 PrimaryKey----------

256.30 <code>PrimaryKey</code> class represents the primary

256.31 key for a table. It is a column in the table for

256.32 that object.

256.33 ----------

256.34 Table----------

256.35 <code>DataColumn</code> class represents a simple type data

256.36 for an object. It is a column in the table for

256.37 that object.

256.38 ----------

256.39 Unique----------

256.40 <code>PrimaryKey</code> class represents the unique

256.41 element for a table. It is a column in the table for

256.42 that object.

256.43 ----------

256.44

257. Logical View::org::biomage::tools::helpers

...
setVerbose(level : int) : void writeOutput(mess : Logical View::java:: initialCap(string : Logical View::java::la initialLower(string : Logical View::java:: writeJavaComment(writer : FileWriter, writeCPPComment(writer : FileWriter,

257.1 ExternalDataCubeIOHelpers@author kjellp

257.2 Description:

257.3 Class that provides static helper metheds for writing and reading

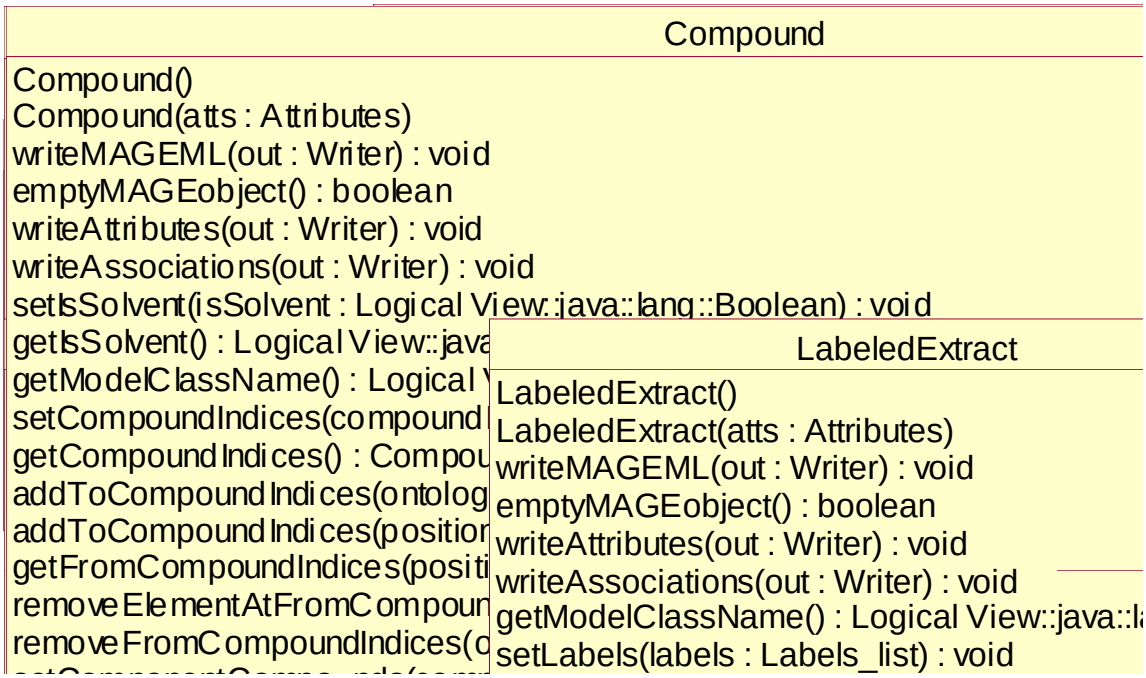
257.4 external cube data to/from a file.

257.5 StringOutputHelpersDescription:

257.6 Class that provides static helper metheds for outputting strings.

257.7

258. Logical View::org::biomage::BioMaterial


Compound
Compound()
Compound(atts : Attributes)
writeMAGEML(out : Writer) : void
emptyMAGEobject() : boolean
writeAttributes(out : Writer) : void
writeAssociations(out : Writer) : void
setIsSolvent(isSolvent : Logical View::java::lang::Boolean) : void
getIsSolvent() : Logical View::java
getModelClassName() : Logical
setCompoundIndices(compound
getCompoundIndices() : Compou
addToCompoundIndices(ontolog
addToCompoundIndices(position
getFromCompoundIndices(positi
removeElementAtFromCompoun
removeFromCompoundIndices(c
LabeledExtract
LabeledExtract()
LabeledExtract(atts : Attributes)
writeMAGEML(out : Writer) : void
emptyMAGEobject() : boolean
writeAttributes(out : Writer) : void
writeAssociations(out : Writer) : void
getModelClassName() : Logical View::java::l
setLabels(labels : Labels_list) : void

258.1 BioMaterialBioMaterial is an abstract class that represents the important

258.2 substances such as cells, tissues, DNA, proteins, etc... Biomaterials

258.3 can be related to other biomaterial through a directed acyclic graph

258.4 (represented by treatment(s)).

258.5 BioMaterialMeasurementA BioMaterialMeasurement is a pairing of a source BioMaterial and an

258.6 amount (Measurement) of that BioMaterial.

258.7 CompoundA Compound can be a simple compound such as SDS (sodium dodecyl

258.8 sulfate). It may also be made of other Compounds in proportions using

258.9 CompoundMeasurements to enumerate the Compounds and their amounts such

258.10 as LB (Luria Broth) Media.

258.11 TreatmentThe process by which a biomaterial is created (from source

258.12 biomaterials). Treatments have an order and an action.

258.13 BioMaterial_packageThe classes in this package describe how a BioSource is treated to

258.14 obtain the BioMaterial (typically a LabeledExtract) that is used by a

258.15 BioAssayCreation in combination with an Array to produce a

258.16 PhysicalBioAssay. A set of treatments are typically linear in time but

258.17 can form a Directed Acyclic Graph.

258.18 BioSampleBioSamples are products of treatments that are of interest.

258.19 BioSamples are often used as the sources for other biosamples. The Type

258.20 attribute describes the role the BioSample holds in the treatment

258.21 hierarchy. This type can be an extract.

258.22 BioSourceThe BioSource is the original source material before any treatment

258.23 events. It is also a top node of the directed acyclic graph generated

258.24 by treatments. The association to OntologyEntry allows enumeration of

258.25 a BioSource's inherent properties.

258.26 CompoundMeasurementA CompoundMeasurement is a pairing of a source Compound and an amount

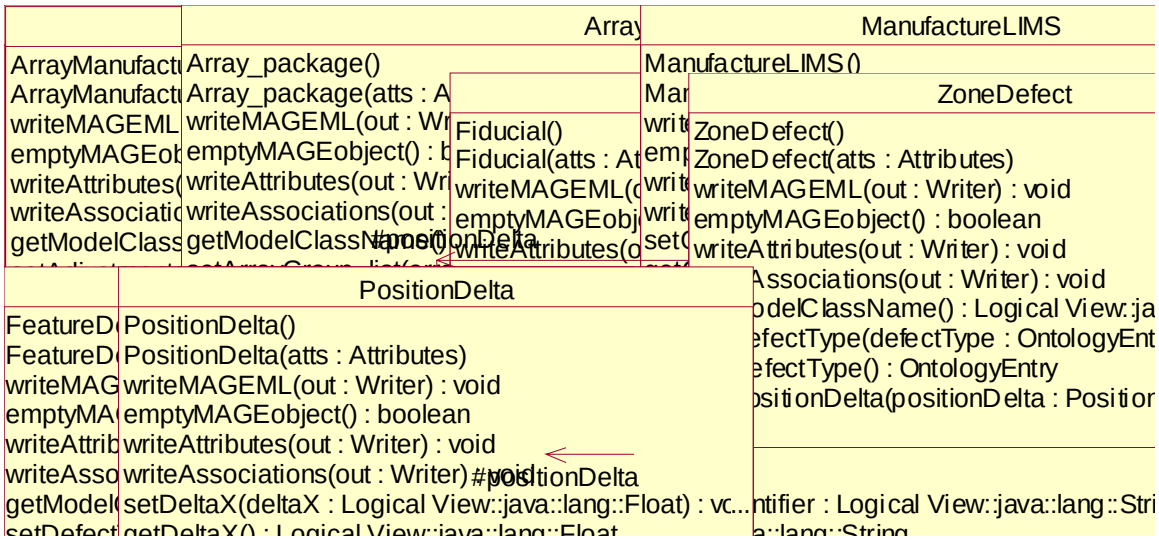
258.27 (Measurement) of that Compound.

258.28 LabeledExtractLabeledExtracts are special BioSamples that have Compounds which are

258.29 detectable (these are often fluorescent or reactive moieties).

258.30

259. Logical View::org::biomage::Array


Array
ArrayManufactu
ArrayManufactu
writeMAGEML
emptyMAGEob
writeAttributes(
writeAssociatio
getModelClass
Array_package()
Array_package(atts : A
writeMAGEML(out : Wr
emptyMAGEobject() : b
writeAttributes(out : Wri
writeAssociations(out :
getModelClassName()
ManufactureLIMS
ManufactureLIMS()
Fiducial()
Fiducial(atts : At
writeMAGEML(
emptyMAGEobj
writeAttributes(o
ZoneDefect
ZoneDefect()
ZoneDefect(atts : Attributes)
writeMAGEML(out : Writer) : void
emptyMAGEobject() : boolean
writeAttributes(out : Writer) : void
Associations(out : Writer) : void
odelClassName() : Logical View::ja
efectType(defectType : OntologyEnt
efectType() : OntologyEntry
ositionDelta(positionDelta : Positior
PositionDelta
FeatureD
FeatureD
writeMAG
emptyMA
writeAttrib
writeAsso
getModel
setDefect
PositionDelta()
PositionDelta(atts : Attributes)
writeMAGEML(out : Writer) : void
emptyMAGEobject() : boolean
writeAttributes(out : Writer) : void
writeAssociations(out : Writer) : void
setDeltaX(deltaX : Logical View::java::lang::Float) : vc...
getDeltaX() : Logical View::java::lang::Float
#positionDelta
ntifier : Logical View::java::lang::Stri
a::lang::String

259.1 ArrayThe physical substrate along with its features and their annotation

259.2 ArrayManufactureDescribes the process by which arrays are produced.

259.3 ArrayGroupAn array package is a physical platform that contains one or more

259.4 arrays that are separately addressable (e.g. several arrays that can be

259.5 hybridized on a single microscope slide) or a virtual grouping together

259.6 of arrays.

259.7 The array package that has been manufactured has information about where

259.8 certain artifacts about the array are located for scanning and feature

259.9 extraction purposes.

259.10 ArrayManufactureDeviationStores information of the potential difference between an array

259.11 design and arrays that have been manufactured using that design (e.g. a

259.12 tip failed to print several spots).

259.13 Array_packageDescribes the process of creating arrays from array designs.

259.14 Includes information on how arrays are grouped together, if relevant,

259.15 how an array deviates from its array design both in layout and per

259.16 feature and potentially contains references to LIMS data that might

259.17 contain more detail on the BioMaterial used to create the reporters.

259.18 FiducialA marking on the surface of the array that can be used to identify

259.19 the array's origin, the coordinates of which are the fiducial's

259.20 centroid.

259.21 ManufactureLIMSInformation on the physical production of arrays within the

259.22 laboratory.

259.23 ZoneDefectStores the defect information for a zone.

259.24 FeatureDefectStores the defect information for a feature.

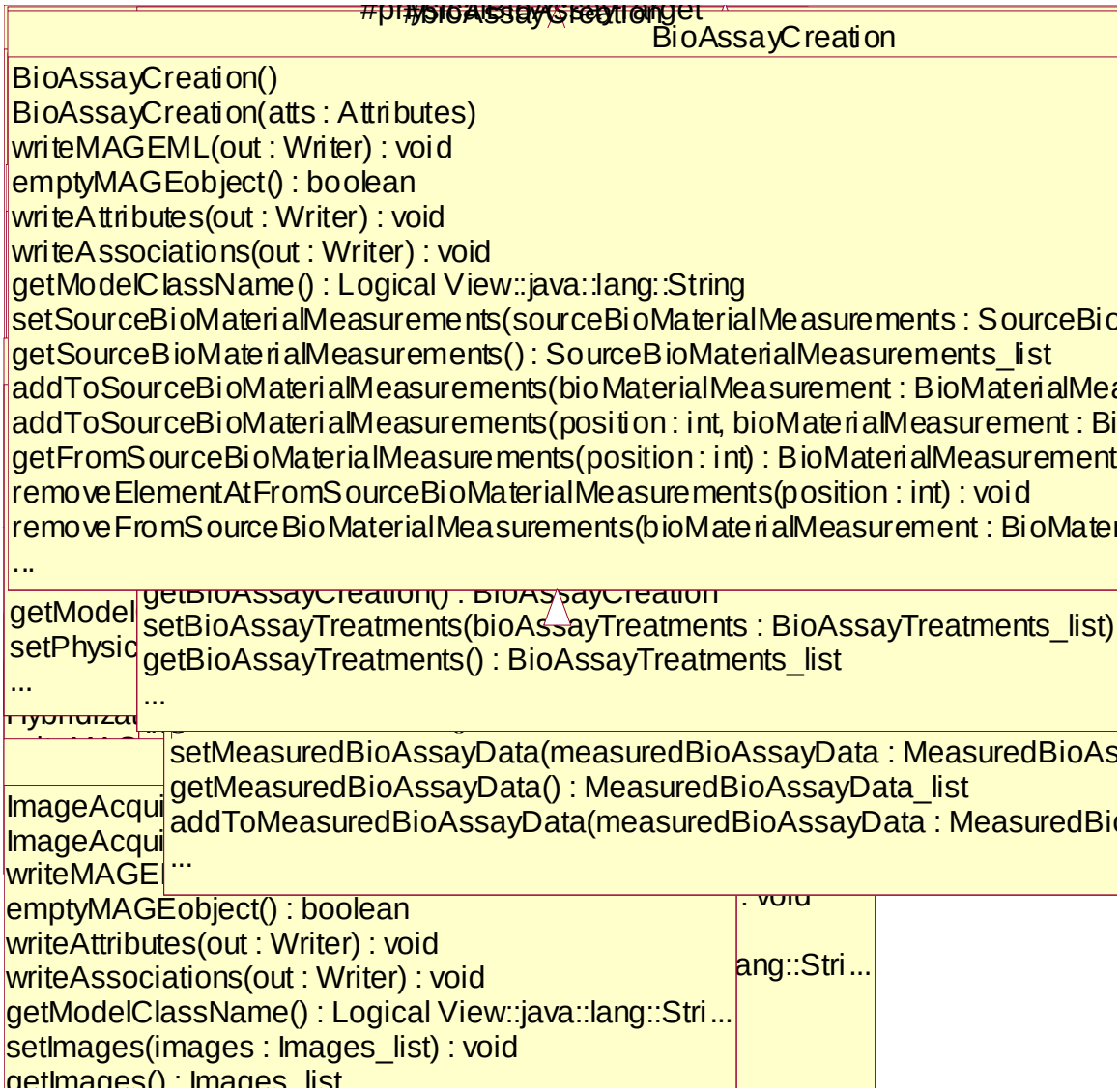
259.25 PositionDeltaThe delta the feature was actually printed on the array from the

259.26 position specified for the feature in the array design.

259.27 ManufactureLIMSBiomaterialStores the location from which a biomaterial was obtained.

259.28

260. Logical View::org::biomage::BioAssay


BioAssayCreation
BioAssayCreation()
BioAssayCreation(atts : Attributes)
writeMAGEML(out : Writer) : void
emptyMAGEobject() : boolean
writeAttributes(out : Writer) : void
writeAssociations(out : Writer) : void
getModelClassName() : Logical View::java::lang::String
setSourceBioMaterialMeasurements(sourceBioMaterialMeasurements : SourceBio
getSourceBioMaterialMeasurements() : SourceBioMaterialMeasurements_list
addToSourceBioMaterialMeasurements(bioMaterialMeasurement : BioMaterialMea
addToSourceBioMaterialMeasurements(position : int, bioMaterialMeasurement : Bi
getFromSourceBioMaterialMeasurements(position : int) : BioMaterialMeasurement
removeElementAtFromSourceBioMaterialMeasurements(position : int) : void
removeFromSourceBioMaterialMeasurements(bioMaterialMeasurement : BioMater
...
setBioAssayTreatments(bioAssayTreatments : BioAssayTreatments_list)
getBioAssayTreatments() : BioAssayTreatments_list
...
setMeasuredBioAssayData(measuredBioAssayData : MeasuredBioAs
getMeasuredBioAssayData() : MeasuredBioAssayData_list
addToMeasuredBioAssayData(measuredBioAssayData : MeasuredBi
...
emptyMAGEobject() : boolean
writeAttributes(out : Writer) : void
writeAssociations(out : Writer) : void
getModelClassName() : Logical View::java::lang::Stri...
setImages(images : Images_list) : void

260.1 BioAssayAn abstract class which represents both physical and computational

260.2 groupings of arrays and biomaterials.

260.3 ChannelA channel represents an independent acquisition scheme for the

260.4 ImageAcquisition event, typically a wavelength.

260.5 BioAssay_packageProvides classes that contain information and annotation on the event

260.6 of joining an Array with a BioMaterial preparation, the acquisition of

260.7 images and the extraction of data on a per feature basis from those

260.8 images. The derived classes of BioAssay represent the base

260.9 PhysicalBioAssays which lead to the production of Images, the

260.10 MeasuredBioAssay which is associated with the set of quantitations

260.11 produced by FeatureExtraction, and DerivedBioAssay (see BioAssayData

260.12 package) which groups together BioAssays that have been analyzed

260.13 together to produce further refinement of the quantitations.

260.14 The design of this package and the related BioAssayData package was

260.15 driven by the following query considerations and the desire to return as

260.16 little data as necessary to satisfy a query. Often, the first set of

260.17 queries for experiments below the Experiment level will want to discover

260.18 the why of an experiment and this is captured in the BioAssay class

260.19 through its FactorValue, BioEvent and Description associations. This

260.20 separates it from the data but allows an overview of the experiment

260.21 hierarchy. The BioAssayData class association to BioDataValues is

260.22 optional only to allow queries on them to discover the how of the

260.23 experiment through the association to the transformation and mappings of

260.24 the three BioAssayData dimensions and the protocols used. Once a

260.25 researcher, for instance, has narrowed down the experiments of interest

260.26 then the actual data, represented by the BioDataValues, can be

260.27 downloaded. Because these data can be in the hundreds of megabytes to

260.28 gigabytes range, it was considered desirable to be able to return

260.29 information and annotation on the experiment without the data.

260.30 BioAssayCreationThe process by which an array and one or more biomaterials are

260.31 combined to create a bioAssayCreation.

260.32 PhysicalBioAssayA bioAssay created by the bioAssayCreation event (e.g. in gene

260.33 expression analysis this event is represented by the hybridization

260.34 event).

260.35 BioAssayTreatmentThe event which records the process by which PhysicalBioAssays are

260.36 processed (typically washing, blocking, etc...).

260.37 DerivedBioAssayA BioAssay that is created by the Transformation BioEvent from one or

260.38 more MeasuredBioAssays or DerivedBioAssays.

260.39 FeatureExtractionThe process by which data is extracted from an image producing a

260.40 measuredBioAssayData and a measuredBioAssay.

260.41 MeasuredBioAssayA measured bioAssay is the direct processing of information in a

260.42 physical bioAssay by the featureExtraction event. Often uses images

260.43 which are referenced through the physical bioAssay.

260.44 HybridizationThe archetypal bioAssayCreation event, whereby biomaterials are

260.45 hybridized to an array.

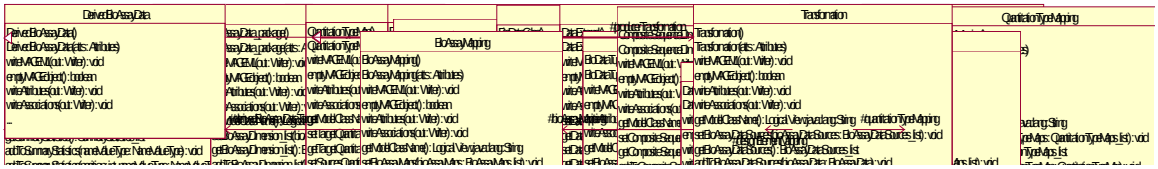
260.46 ImageAn image is created by an imageAcquisition event, typically by

260.47 scanning the hybridized array (the PhysicalBioAssay).

260.48 ImageAcquisitionThe process by which an image is generated (typically scanning).

260.49

261. Logical View::org::biomage::BioAssayData



261.1 DerivedBioAssayDataThe output of a transformation event.

261.2 BioAssayMapThe BioAssayMap is the description of how source MeasuredBioAssays

261.3 and/or DerivedBioAssays are manipulated (mathematically) to produce

261.4 DerivedBioAssays.

261.5 MeasuredBioAssayDataThe data associated with the MeasuredBioAssay produced by

261.6 FeatureExtraction.

261.7 BioAssayDataRepresents the dataset created when the BioAssays are created.

261.8 BioAssayData is the entry point to the values. Because the actual

261.9 values are represented by a different object, BioDataValues, which can

261.10 be memory intensive, the annotation of the transformation can be gotten

261.11 separate from the data.

261.12 BioAssayDimensionAn ordered list of bioAssays.

261.13 DesignElementDimensionAn ordered list of designElements. It will be realized as one of its

261.14 three subclasses.

261.15 QuantitationTypeDimensionAn ordered list of quantitationTypes.

261.16 BioDataValuesThe actual values for the BioAssayCube.

261.17 BioAssayData_packageThe classes defined here provide data and the information and

261.18 annotation on the derivation of that data. Some of the scenarios that

261.19 might occur are the following.

261.20 FeatureExtraction of a single PhysicalBioAssay produces

261.21 MeasuredBioAssayData that has a single BioAssay on the

261.22 BioAssayDimension, typically the Features described in the ArrayDesign

261.23 on the DesignElementDimension, and the Quantitations associated with the

261.24 application of a FeatureExtraction protocol on the

261.25 QuantitationDimension.

261.26 An error model transformation might be applied that doesn't change the

261.27 BioAssayDimension or the DesignElementDimension but likely changes the

261.28 QuantitationDimension. A transformation on replicate Reporters or

261.29 CompositeSequences might be applied on the single BioAssay that would

261.30 change the DesignElementDimension and the QuantitationDimension both.

261.31 Replicate and Control BioAssays might be added to the BioAssayDimension

261.32 and a transformation could change the BioAssayDimension and the

261.33 QuantitationDimension but not change the DesignElementDimension to

261.34 produce a new DerivedBioAssayData. Or some combination of the above

261.35 transformations could be performed at once to change all three

261.36 dimensions.

261.37 Because the classes derive from Describable, the Experimenter can

261.38 provide as much detail at each level of the class hierarchy as desired.

261.39 QuantitationTypeMapA QuantitationTypeMap is the description of how source

261.40 QuantitationTypes are mathematically transformed into a target

261.41 QuantitationType.

261.42 BioAssayMappingContainer of the mappings of the input BioAssay dimensions to the

261.43 output BioAssay dimension.

261.44 BioAssayTupleTransformed container to specify a BioAssay and the Design Elements

261.45 and their data for that BioAssay.

261.46 DesignElementTupleTransformed container to specify a DesignElement and

261.47 QuantitationTypes for that Element.

261.48 BioDataCubeA three-dimensional cube representation of the data.

261.49 DataInternalTransformed class to associate whitespaced delimited data to the

261.50 BioAssayDataCube

261.51 DataExternalTransformed class to associate external data to the BioAssayDataCube

261.52 BioDataTuplesA relational, tuple representation of the data.

261.53 CompositeSequenceDimensionSpecialized DesignElementDimension to hold CompositeSequences.

261.54 DatumTransformed container to hold a value. QuantitationType will

261.55 determine the type of this value.

261.56 TransformationThe process by which derivedBioAssays are created from

261.57 measuredBioAssays and/or derivedBioAssays. It uses mappings to indicate

261.58 the input and output dimensions.

261.59 DesignElementMapA DesignElementMap is the description of how source DesignElements

261.60 are transformed into a target DesignElement.

261.61 DesignElementMappingContainer of the mappings of the input DesignElement dimensions to

261.62 the output DesignElement dimension.

261.63 QuantitationTypeTupleTransformed container to specify a Quantitation Type and the value

261.64 for that Type.

261.65 FeatureDimensionSpecialized DesignElementDimension to hold Features.

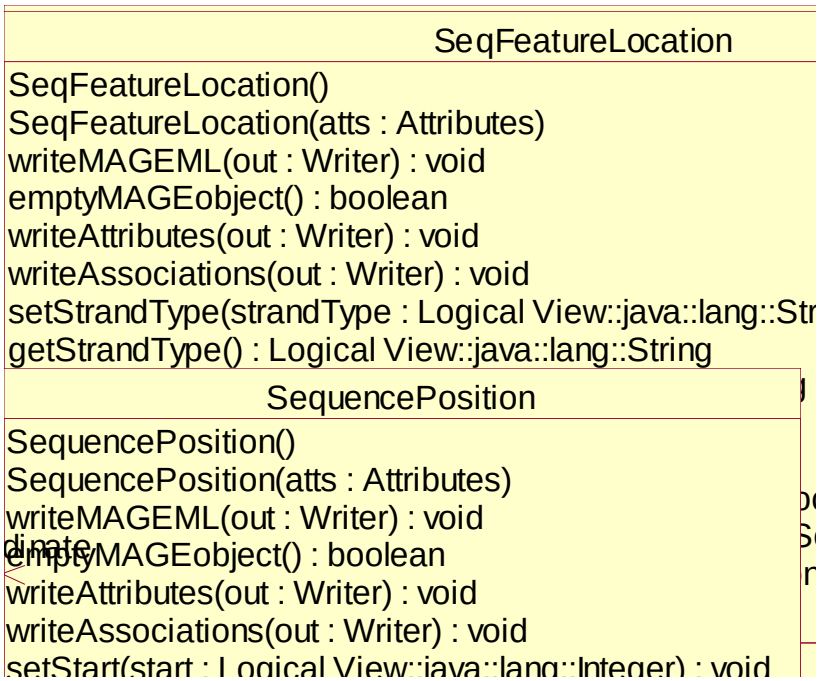
261.66 QuantitationTypeMappingContainer of the mappings of the input QuantitationType dimensions to

261.67 the output QuantitationType dimension.

261.68 ReporterDimensionSpecialized DesignElementDimension to hold Reporters.

261.69

262. Logical View::org::biomage::BioSequence


SeqFeatureLocation
SeqFeatureLocation()
SeqFeatureLocation(atts : Attributes)
writeMAGEML(out : Writer) : void
emptyMAGEobject() : boolean
writeAttributes(out : Writer) : void
writeAssociations(out : Writer) : void
setStrandType(strandType : Logical View::java::lang::Str
getStrandType() : Logical View::java::lang::String
SequencePosition
SequencePosition()
SequencePosition(atts : Attributes)
writeMAGEML(out : Writer) : void
emptyMAGEobject() : boolean
writeAttributes(out : Writer) : void
writeAssociations(out : Writer) : void
setStart(start : Logical View::java::lang::Integer) : void

262.1 BioSequenceA BioSequence is a representation of a DNA, RNA, or protein sequence.

262.2 It can be represented by a Clone, Gene, or the sequence.

262.3 SeqFeatureRepresents, in general, what would be a GenBank Feature Table

262.4 annotation for a sequence.

262.5 BioSequence_packageDescribes a known gene or sequence. BioAssays typically seek to

262.6 identify what BioSequences are expressed in a BioMaterial after

262.7 treatments, the expression level measured from the association between

262.8 the BioMaterial and the Array. The Array's Features typically provide

262.9 known locations for this association to occur. Most often, the Reporter

262.10 and CompositeSequence are known and the presence or absence of a

262.11 particular BioSequence in the BioMaterial is based on whether there as

262.12 been an association to the DesignElement targeted for it. Some other

262.13 experiments may not know the DesignElement's target but can discover it

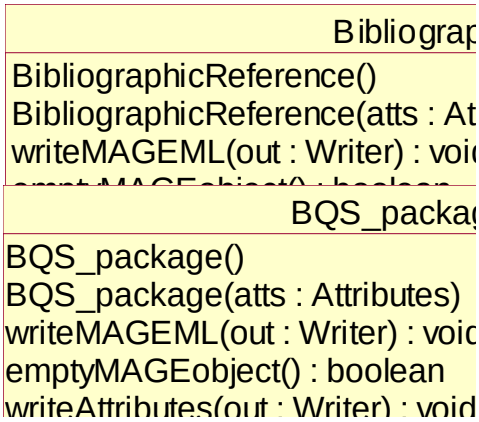
262.14 with known properties of the BioSequences in the BioMaterial.

262.15 SeqFeatureLocationThe location of the SeqFeature annotation.

262.16 SequencePositionDesignates the position of the Feature in its BioSequence.

262.17

263. Logical View::org::biomage::BQS



263.1 BibliographicReferenceAttributes for the most common criteria and association with

263.2 OntologyEntry allows criteria to be specified for searching for a

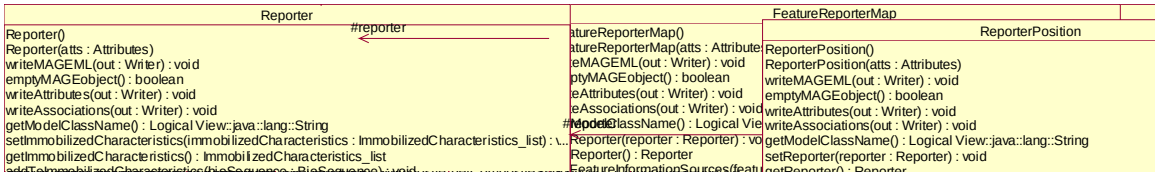
263.3 Bibliographic reference.

263.4 BQS_packageAllows a reference to an article, book or other publication to be

263.5 specified for searching repositories.

263.6

264. Logical View::org::biomage::DesignElement



264.1 FeatureAn intended position on an array.

264.2 PositionSpecifies a position on an array.

264.3 CompositeSequenceA collection of Reporter or CompositeSequence Design Elements,

264.4 annotated through the association to BioSequence.

264.5 ReporterA Design Element that represents some biological material (clone,

264.6 oligo, etc.) on an array which will report on some biosequence or

264.7 biosequences. The derived data from the measured data of its Features

264.8 represents the presence or absence of the biosequence or biosequences it

264.9 is reporting on in the BioAssay.

264.10 Reporters are Identifiable and several Features on the same array can be

264.11 mapped to the same reporter as can Features from a different

264.12 ArrayDesign. The granularity of the Reporters independence is dependent

264.13 on the technology and the intent of the ArrayDesign. Oligos using

264.14 mature technologies can in general be assumed to be safely replicated on

264.15 many features where as with PCR Products there might be the desire for

264.16 quality assurence to make reporters one to one with features and use the

264.17 mappings to CompositeSequences for replication purposes.

264.18 DesignElementAn element of an array. This is generally of type feature but can be

264.19 specified as reporters or compositeSequence for arrays that are

264.20 abstracted from a physical array.

264.21 DesignElement_packageThe classes of this package are the contained classes of the

264.22 ArrayDesign and describe through the DesignElements what is intended to

264.23 be at each location of the Array. The Feature describes an intended

264.24 location on the Array, the Reporter the Oligo, Clone, PCR Product that

264.25 is on a Feature and the CompositeSequence which combines Reporters or

264.26 CompositeSequences into what the child DesignElements are meant to

264.27 represent biologically, e.g. a Gene, Exon, SpliceVariant, etc.

264.28 CompositeCompositeMapA CompositeCompositeMap is the description of how source

264.29 CompositeSequences are transformed into a target CompositeSequence.

264.30 For instance, several CompositeSequences could represent different

264.31 sequence regions for a Gene and could be mapped to different

264.32 CompositeSequences, each representing a different splice variant for

264.33 that Gene.

264.34 CompositePositionThe location in the compositeSequence target's sequence to which a

264.35 source compositeSequence maps. The association to MismatchInformation

264.36 allows the specification, usually for control purposes, of deviations

264.37 from the CompositeSequence's BioMaterial.

264.38 MismatchInformationDescribes how a reporter varies from its ReporterCharacteristics

264.39 sequence(s) or how a Feature varies from its Reporter sequence.

264.40 ReporterCompositeMapA ReporterCompositeMap is the description of how source Reporters are

264.41 transformed into a target CompositeSequences. For instance, several

264.42 reporters that tile across a section of a chromosome could be mapped to

264.43 a CompositeSequence.

264.44 FeatureReporterMapA FeatureReporterMap is the description of how source features are

264.45 transformed into a target reporter. These would map replicate features

264.46 for a reporter to the reporter.

264.47 FeatureLocationSpecifies where a feature is located relative to a grid.

264.48 FeatureInformationAs part of the map information, allows the association of one or more

264.49 differences in the BioMaterial on a feature from the BioMaterial of the

264.50 Reporter. Useful for control purposes such as in Affymetrix probe

264.51 pairs.

264.52 ReporterPositionThe location in the composite target's sequence to which a source

264.53 reporter maps. The association to MismatchInformation allows the

264.54 specification, usually for control purposes, of deviations from the

264.55 CompositeSequence's BioMaterial.

264.56

265. Logical View::org::biomage::Experiment

ExperimentalFactor

ExperimentalFactor()
ExperimentalFactor(atts : Attributes)
writeMAGEML(out : Writer) : void
emptyMAGEobject() : boolean
writeAttributes(out : Writer) : void
writeAssociations(out : Writer) : void
getModelClassName() : Logical View::java::lang::S
setCategory(category : OntologyEntry) : void
getCategory() : OntologyEntry
setFactorValues(factorValues : FactorValues_list)
getFactorValues() : FactorValues_list
addToFactorValues(factorValue : FactorValue) : vo
addToFactorValues(position : int, factorValue : Fac
getFromFactorValues(position : int) : FactorValue
removeElementAtFromFactorValues(position : int)
removeFromFactorValues(factorValue : FactorValu
setAnnotations(annotations : Annotations_list) : voi
getAnnotations() : Annotations_list
...

FactorValue()
FactorValue(atts : Attrib
writeMAGEML(out : Wri
emptyMAGEobject() : b
writeAttributes(out : Writ
writeAssociations(out : \
getModelClassName() :
setExperimentalFactor(
getExperimentalFactor(
setMeasurement(measu
...

getFromTypes(position :
removeElementAtFromT
removeFromTypes(onto
setTopLevelBioAssays(
getTopLevelBioAssays(
addToTopLevelBioAssa
addToTopLevelBioAssa
getFromTopLevelBioAs
removeElementAtFromT
removeFromTopLevelBioAssays(bioAssay : BioAssay) : void
setExperimentalFactors(experimentalFactors : ExperimentalFactors_list) : v
getExperimentalFactors() : ExperimentalFactors_list
addToExperimentalFactors(experimentalFactor : ExperimentalFactor) : voic
addToExperimentalFactors(position : int, experimentalFactor : Experimenta
getFromExperimentalFactors(position : int) : ExperimentalFactor
removeElementAtFromExperimentalFactors(position : int) : void
...

getBioAssays() : BioAssays_list
addToBioAssays(bioAssay : BioAssay) : void
addToBioAssays(position : int, bioAssay : BioAssay) : void
getFromBioAssays(position : int) : BioAssay
removeElementAtFromBioAssays(position : int) : void
...

265.1 FactorValueThe value for a ExperimentalFactor

265.2 Experiment_packageRepresents the container for a hierarchical grouping of BioAssays.

265.3 Can have the end results of Clustering Analysis specified and, through

265.4 the ExperimentDesign, a description and annotation of the overall design

265.5 of the experiment and what it was to show.

265.6 MAGEExperimentThe Experiment is the collection of all the BioAssays that are

265.7 related by the ExperimentDesign.

265.8 ExperimentDesignThe ExperimentDesign is the description and collection of

265.9 ExperimentalFactors and the hierarchy of BioAssays to which they

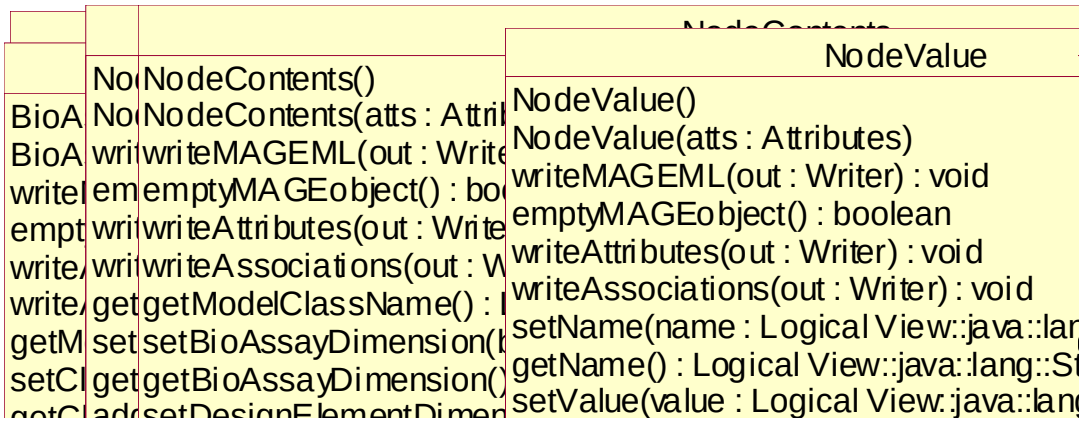
265.10 pertain.

265.11 ExperimentalFactorExperimentFactors are the dependent variables of an experiment (e.g.

265.12 time, glucose concentration, ...).

265.13

266. Logical View::org::biomage::HigherLevelAnalysis



266.1 HigherLevelAnalysis_packageDescribes the results of performing analysis on the result of the

266.2 BioAssayData from an Experiment.

266.3 BioAssayDataClusterA mathematical method of higher level analysis whereby BioAssayData

266.4 are grouped together into nodes.

266.5 NodeAn individual component of a clustering. May contain other nodes.

266.6 NodeContentsThe contents of a node for any or all of the three Dimensions. If a

266.7 node only contained genes just the DesignElementDimension would be

266.8 defined.

266.9 NodeValueA value associated with the Node that can rank it in relation to the

266.10 other nodes produced by the clustering algorithm.

266.11

267. Logical View::org::biomage::Interface

267.1 HasArrayDesign

267.2 HasArrayGroup

267.3 HasInformation

267.4 HasArrayManufactureDeviations

267.5 HasArrays

267.6 HasFiducials

267.7 HasDistanceUnit

267.8 HasSubstrateType

267.9 HasProtocolApplications

267.10 HasFeatureLIMSs

267.11 HasArrayManufacturers

267.12 HasQualityControlStatistics

267.13 HasFeatureDefects

267.14 HasAdjustments

267.15 HasPositionDelta

267.16 HasFeature

267.17 HasDefectType

267.18 HasFiducialType

267.19 HasPosition

267.20 HasIdentifierLIMS

267.21 HasBioMaterial

267.22 HasZone

267.23 HasCompositeGroups

267.24 HasDesignProviders

267.25 HasReporterGroups

267.26 HasFeatureGroups

267.27 HasCompositeSequences

267.28 HasSpecies

267.29 HasTypes

267.30 HasFeatures

267.31 HasTechnologyType

267.32 HasFeatureShape

267.33 HasZoneGroups

267.34 HasSurfaceType

267.35 HasReporters

267.36 HasZoneLayout

267.37 HasZoneLocations

267.38 HasPerformer

267.39 HasRoles

267.40 HasParent

267.41 HasAffiliation

267.42 HasSecurityGroups

267.43 HasOwner

267.44 HasMembers

267.45 HasChannels

267.46 HasBioAssayFactorValues

267.47 HasArray

267.48 HasSourceBioMaterialMeasurements

267.49 HasPhysicalBioAssayTarget

267.50 HasPhysicalBioAssay

267.51 HasTarget

267.52 HasLabels

267.53 HasDerivedBioAssayData

267.54 HasType

267.55 HasDerivedBioAssayMap

267.56 HasPhysicalBioAssaySource

267.57 HasMeasuredBioAssayTarget

267.58 HasFormat

267.59 HasImages

267.60 HasFeatureExtraction

267.61 HasMeasuredBioAssayData

267.62 HasPhysicalBioAssayData

267.63 HasBioAssayCreation

267.64 HasBioAssayTreatments

267.65 HasBioAssayDimension

267.66 HasDesignElementDimension

267.67 HasQuantitationTypeDimension

267.68 HasSummaryStatistics

267.69 HasBioDataValues

267.70 HasBioAssays

267.71 HasBioAssayMapTarget

267.72 HasSourceBioAssays

267.73 HasBioAssayMaps

267.74 HasDesignElementTuples

267.75 HasBioAssayTupleData

267.76 HasProducerTransformation

267.77 HasDesignElementMaps

267.78 HasQuantitationTypeTuples

267.79 HasContainedFeatures

267.80 HasQuantitationTypes

267.81 HasTargetQuantitationType

267.82 HasSourcesQuantitationType

267.83 HasQuantitationTypeMaps

267.84 HasBioAssayDataSources

267.85 HasBioAssayMapping

267.86 HasDerivedBioAssayDataTarget

267.87 HasQuantitationTypeMapping

267.88 HasDesignElementMapping

267.89 HasTreatments

267.90 HasMaterialType

267.91 HasCharacteristics

267.92 HasMeasurement

267.93 HasSourceContact

267.94 HasComponentCompounds

267.95 HasCompoundIndices

267.96 HasExternalLIMS

267.97 HasCompound

267.98 HasCompoundMeasurements

267.99 HasActionMeasurement

267.100 HasAction

267.101 HasSequenceDatabases

267.102 HasOntologyEntries

267.103 HasPolymerType

267.104 HasSeqFeatures

267.105 HasRegions

267.106 HasSubregions

267.107 HasCoordinate

267.108 HasAccessions

267.109 HasParameters

267.110 HasSecurity

267.111 HasAuditTrail

267.112 HasDescriptions

267.113 HasPropertySets

267.114 HasContacts

267.115 HasDatabase

267.116 HasDatabaseReferences

267.117 HasBibliographicReferences

267.118 HasExternalReference

267.119 HasAnnotations

267.120 HasOntologyReference

267.121 HasAssociations

267.122 HasCompositePositionSources

267.123 HasCompositeSequence

267.124 HasComposite

267.125 HasMismatchInformation

267.126 HasBiologicalCharacteristics

267.127 HasReporterCompositeMaps

267.128 HasCompositeCompositeMaps

267.129 HasControlType

267.130 HasFeatureGroup

267.131 HasControlledFeatures

267.132 HasControlFeatures

267.133 HasFeatureLocation

267.134 HasReporter

267.135 HasFeatureInformationSources

267.136 HasFailTypes

267.137 HasWarningType

267.138 HasImmobilizedCharacteristics

267.139 HasFeatureReporterMaps

267.140 HasReporterPositionSources

267.141 HasAnalysisResults

267.142 HasProviders

267.143 HasBioAssayData

267.144 HasExperimentDesigns

267.145 HasFactorValues

267.146 HasCategory

267.147 HasTopLevelBioAssays

267.148 HasExperimentalFactors

267.149 HasReplicateDescription

267.150 HasQualityControlDescription

267.151 HasNormalizationDescription

267.152 HasExperimentalFactor

267.153 HasValue

267.154 HasNodes

267.155 HasClusterBioAssayData

267.156 HasNodeContents

267.157 HasNodeValue

267.158 HasQuantitationDimension

267.159 HasScale

267.160 HasDataType

267.161 HasBioAssay

267.162 HasChannel

267.163 HasConfidenceIndicators

267.164 HasDefaultValue

267.165 HasDesignElement

267.166 HasHardware

267.167 HasHardwareApplications

267.168 HasHardwareManufacturers

267.169 HasHardwares

267.170 HasParameterType

267.171 HasParameterTypes

267.172 HasParameterValues

267.173 HasPerformers

267.174 HasProtocol

267.175 HasQuantitationType

267.176 HasSoftware

267.177 HasSoftwareApplications

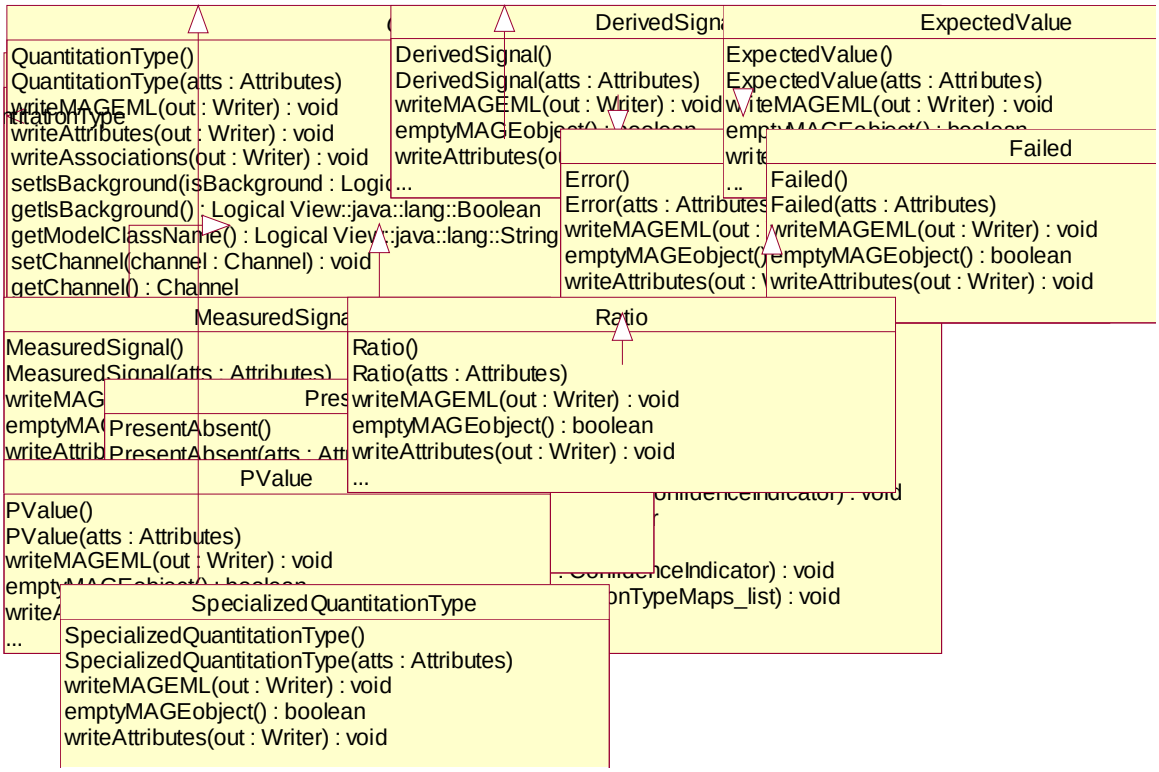
267.178 HasSoftwareManufacturers

267.179 HasSoftwares

267.180 HasUnit

267.181

268. Logical View::org::biomage::QuantitationType



268.1 QuantitationTypeA method for calculating a single datum of the matrix (e.g. raw

268.2 intensity, background, error).

268.3 QuantitationType_packageThis Package defines the classes for quantitations, such as measured

268.4 and derived signal, error, and pvalue. The subclasses of

268.5 StandardQuantitationType will be the best fit from FeatureExtraction or

268.6 Transformation Protocol for the values obtained. Other values can be

268.7 specified using SpecializedQuantitationType.

268.8 ConfidenceIndicatorIndication of some measure of confidence for a standard quantitation

268.9 type.

268.10 StandardQuantitationTypeSuperclass for the named quantitation type. Useful for mapping to

268.11 those languages that can use a fly-weight for processing the subclasses.

268.12 DerivedSignalA calculated measurement of the intensity of a signal, for example,

268.13 after a transformation involving normalization and/or replicate

268.14 DesignElements. Of type float.

268.15 ErrorError measurement of a quantitation. Of type float.

268.16 ExpectedValueIndication of what value is expected of the associated standard

268.17 quantitation type.

268.18 FailedValues associated with this QuantitationType indicate a failure of

268.19 some kind for a particular DesignElement for a BioAssay. Of type

268.20 boolean.

268.21 MeasuredSignalBest measure from feature extraction as to the presence and intensity

268.22 of the signal. Of type float.

268.23 PresentAbsentIndicates relative presence or absence. From the enumeration

268.24 AbsoluteCallTypeEnum {Present | Absent | Marginal | No call} or

268.25 ComparisonCallTypeEnum {Increase I Marginal Increase | Decrease |

268.26 Marginal Decrease | No change | No Call | Unknown }, as specified by

268.27 the dataType.

268.28 PValueMeasurement of the accuracy of a quantitation. Of type float.

268.29 RatioThe ratio of two or more signals, typically between two channels. Of

268.30 type float.

268.31 SpecializedQuantitationTypeUser defined quantitation type.

268.32