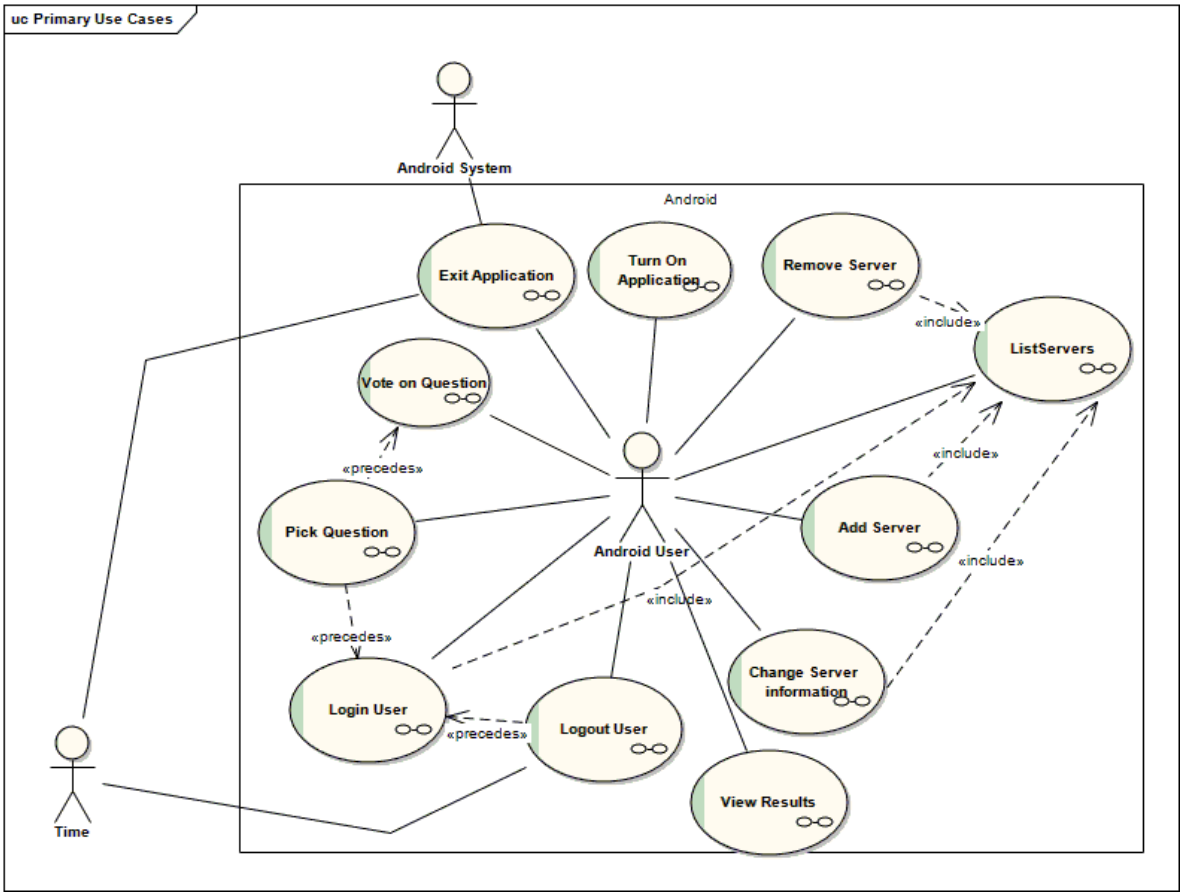
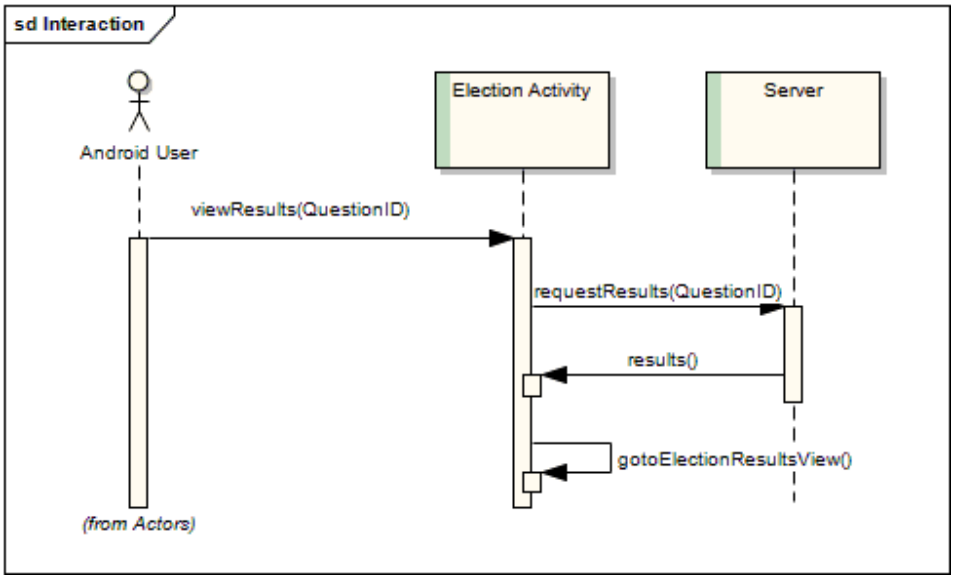
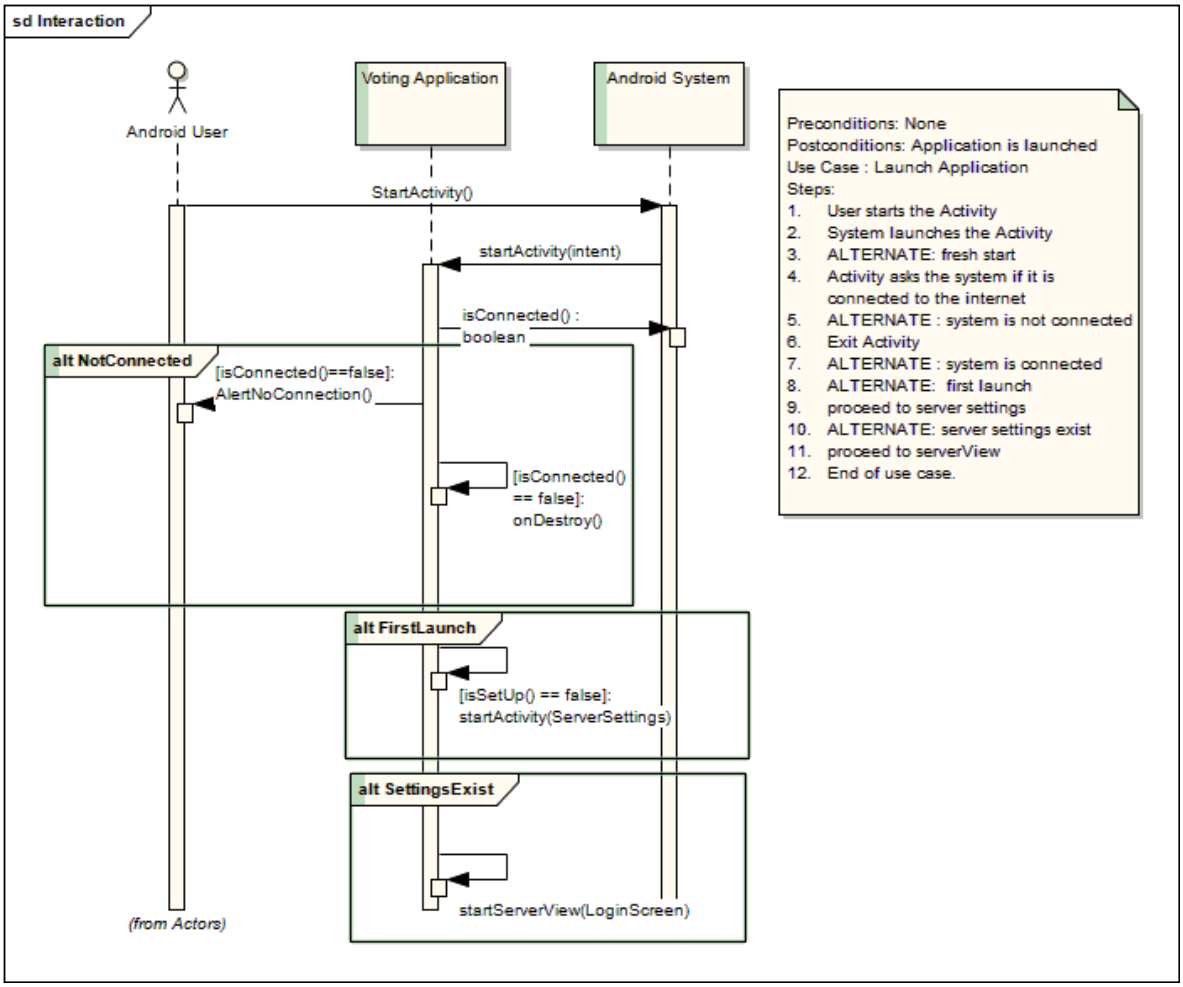
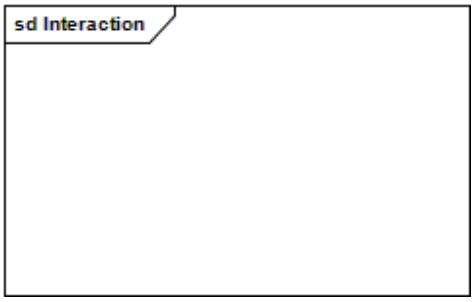
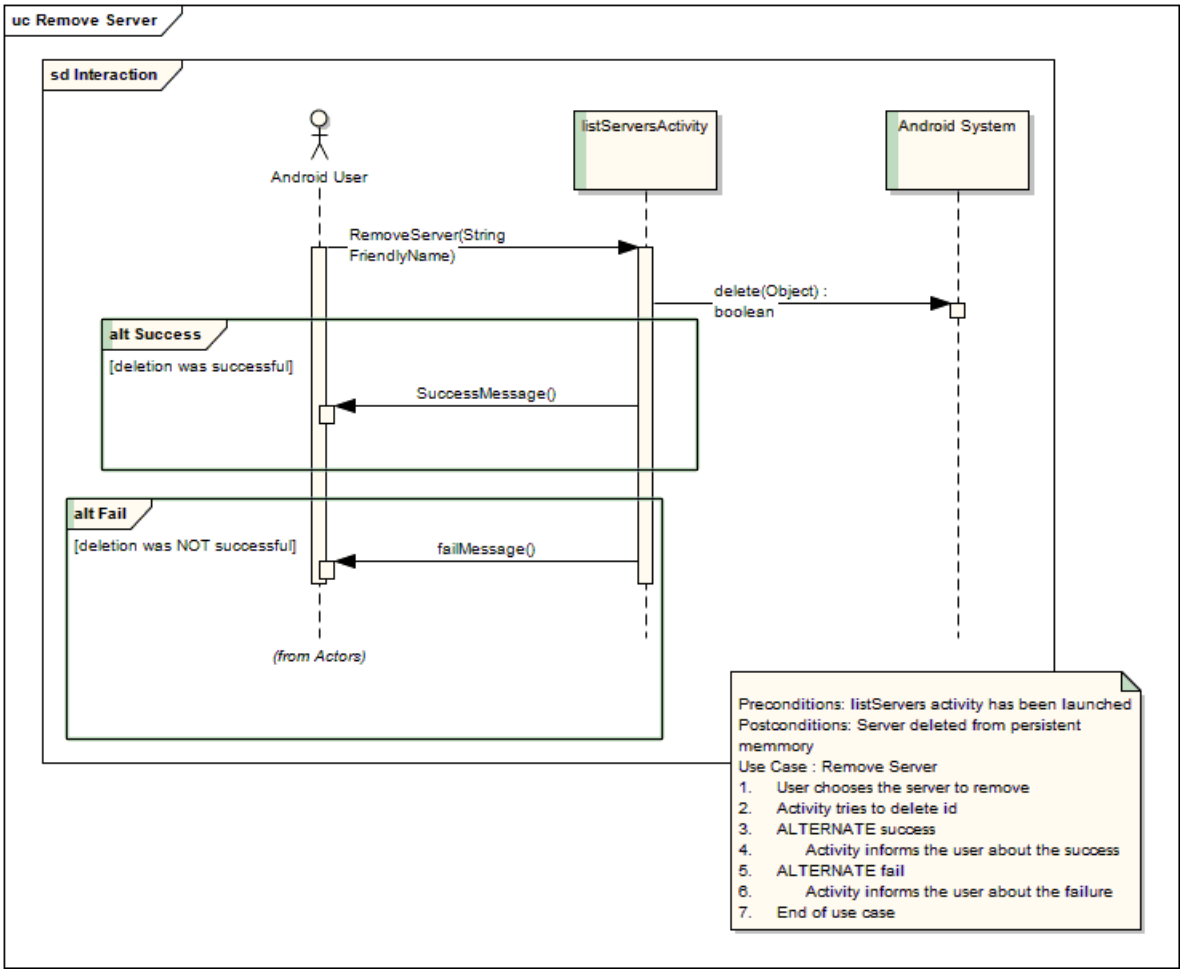
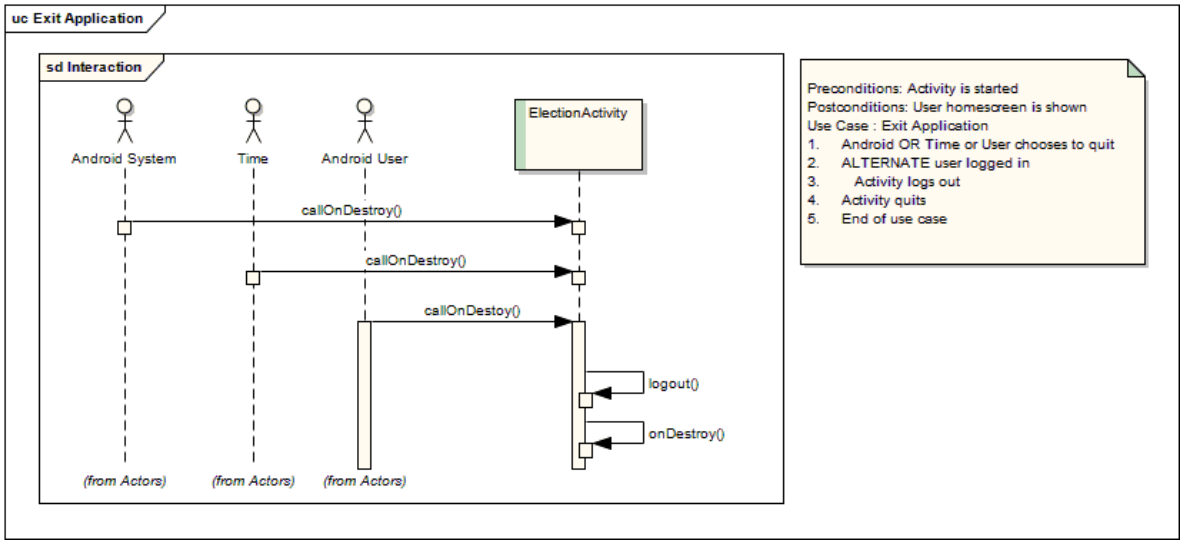

uc Primary Use Cases
Android System
Android
Exit Application
Turn On Application
Remove Server
«include»
ListServers
Vote on Question
«precedes»
Pick Question
Android User
«include»
Add Server
«include»
«include»
«precedes»
Login User
«precedes»
Logout User
Change Server information
View Results
Time

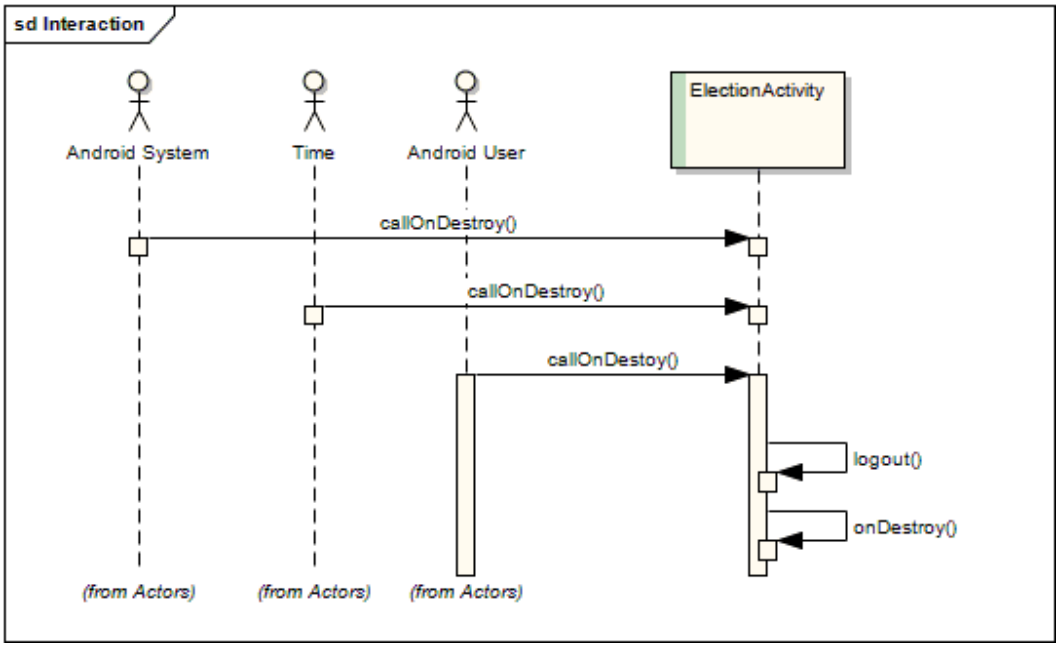

sd Interaction
Android User
Election Activity
Server
viewResults(QuestionID)
requestResults(QuestionID)
results()
gotoElectionResultsView()
(from Actors)

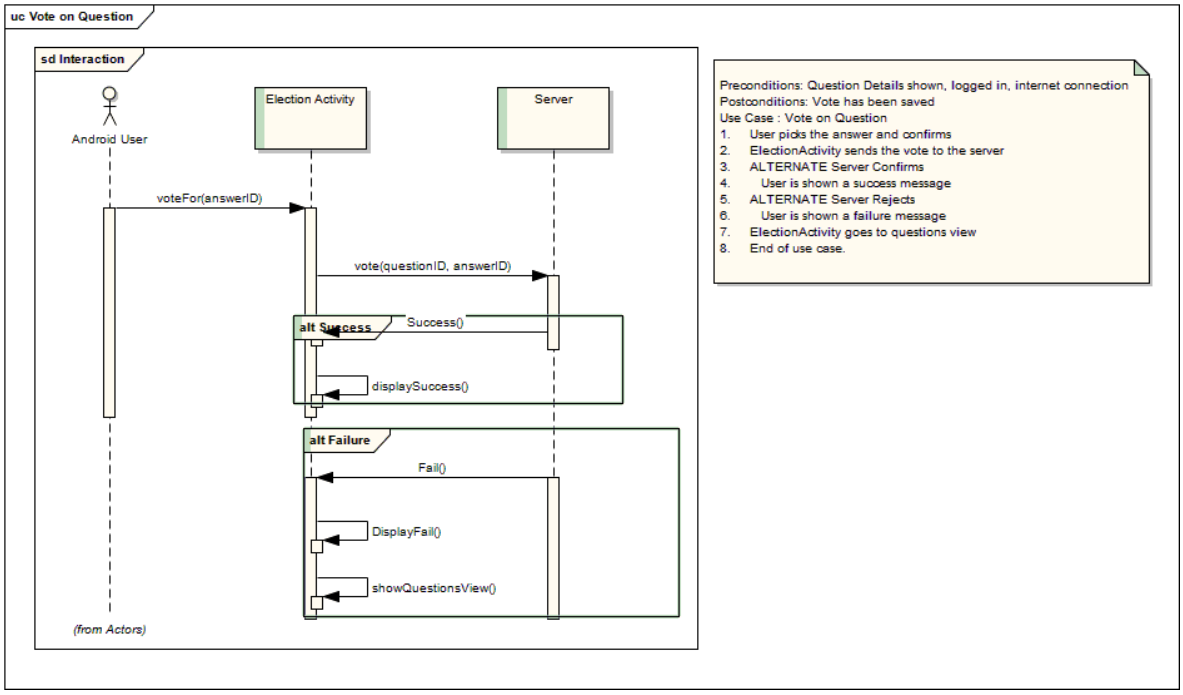

sd Interaction
Android User
Voting Application
Android System
StartActivity()
startActivity(intent)
isConnected() :
boolean
alt NotConnected
[isConnected()==false]:
AlertNoConnection()
[isConnected()
== false]:
onDestroy()
alt FirstLaunch
[isSetUp() == false]:
startActivity(ServerSettings)
alt SettingsExist
startServerView(LoginScreen)
(from Actors)
Preconditions: None
Postconditions: Application is launched
Use Case : Launch Application
Steps:
1. User starts the Activity
2. System launches the Activity
3. ALTERNATE: fresh start
4. Activity asks the system if it is connected to the internet
5. ALTERNATE : system is not connected
6. Exit Activity
7. ALTERNATE : system is connected
8. ALTERNATE: first launch
9. proceed to server settings
10. ALTERNATE: server settings exist
11. proceed to serverView
12. End of use case.

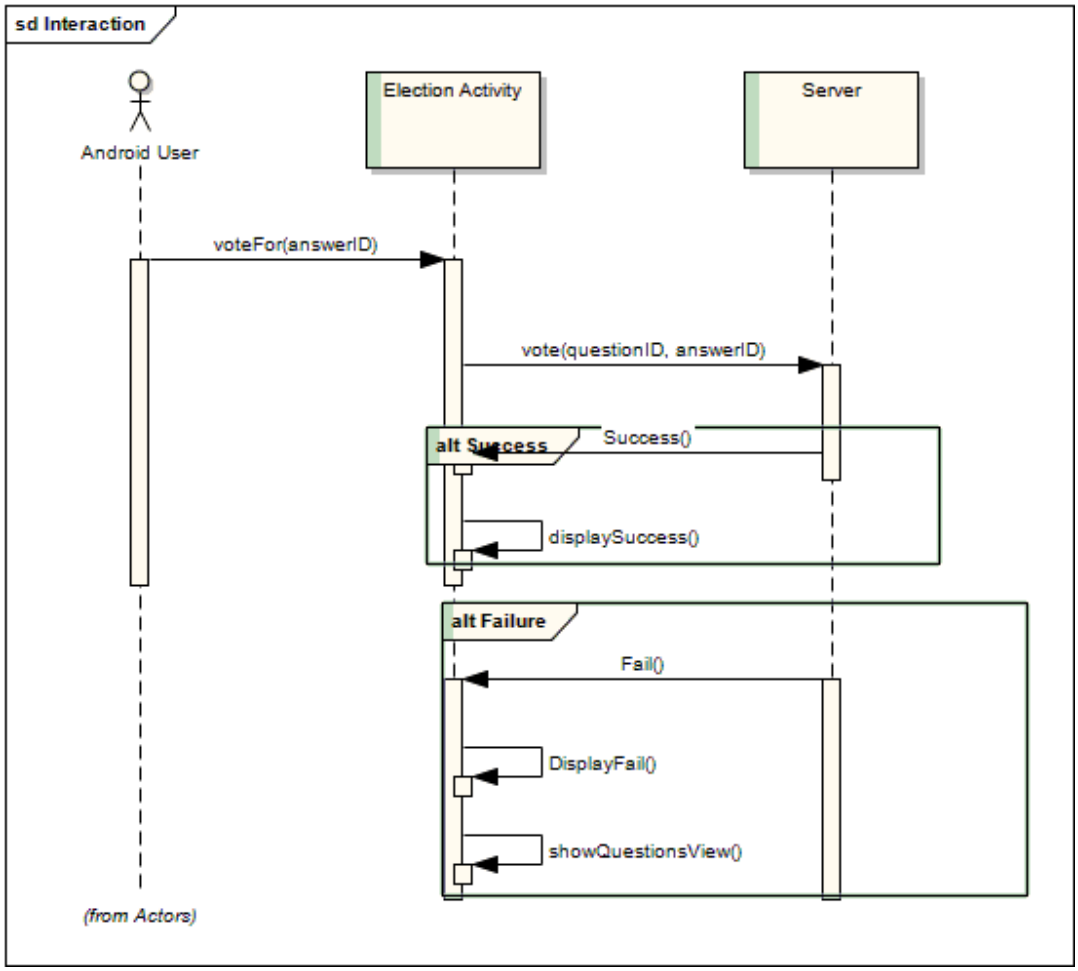

sd Interaction

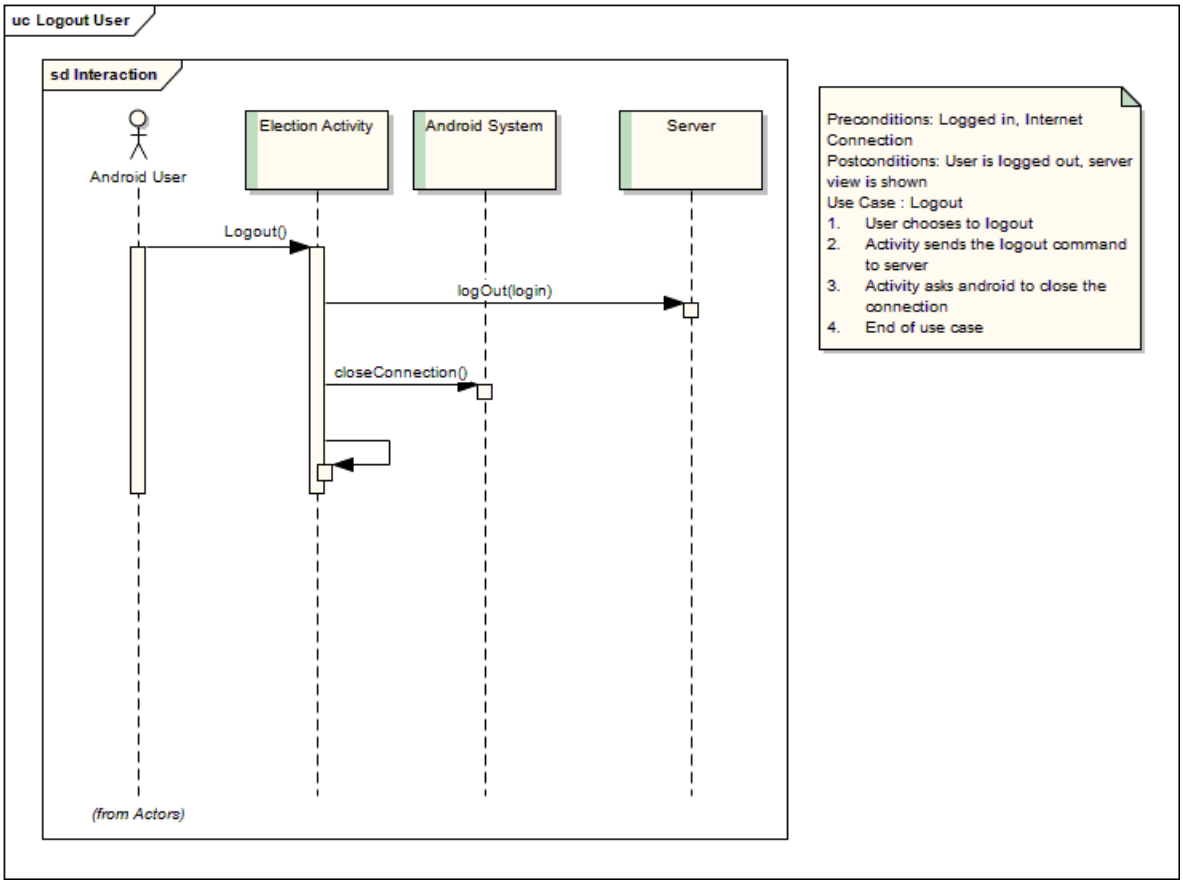

uc Remove Server
sd Interaction
Android User
listServersActivity
Android System
RemoveServer(String FriendlyName)
delete(Object) : boolean
alt Success
[deletion was successful]
SuccessMessage()
alt Fail
[deletion was NOT successful]
failMessage()
(from Actors)
Preconditions: listServers activity has been launched
Postconditions: Server deleted from persistent memmory
Use Case : Remove Server
1. User chooses the server to remove
2. Activity tries to delete id
3. ALTERNATE success
4. Activity informs the user about the success
5. ALTERNATE fail
6. Activity informs the user about the failure
7. End of use case

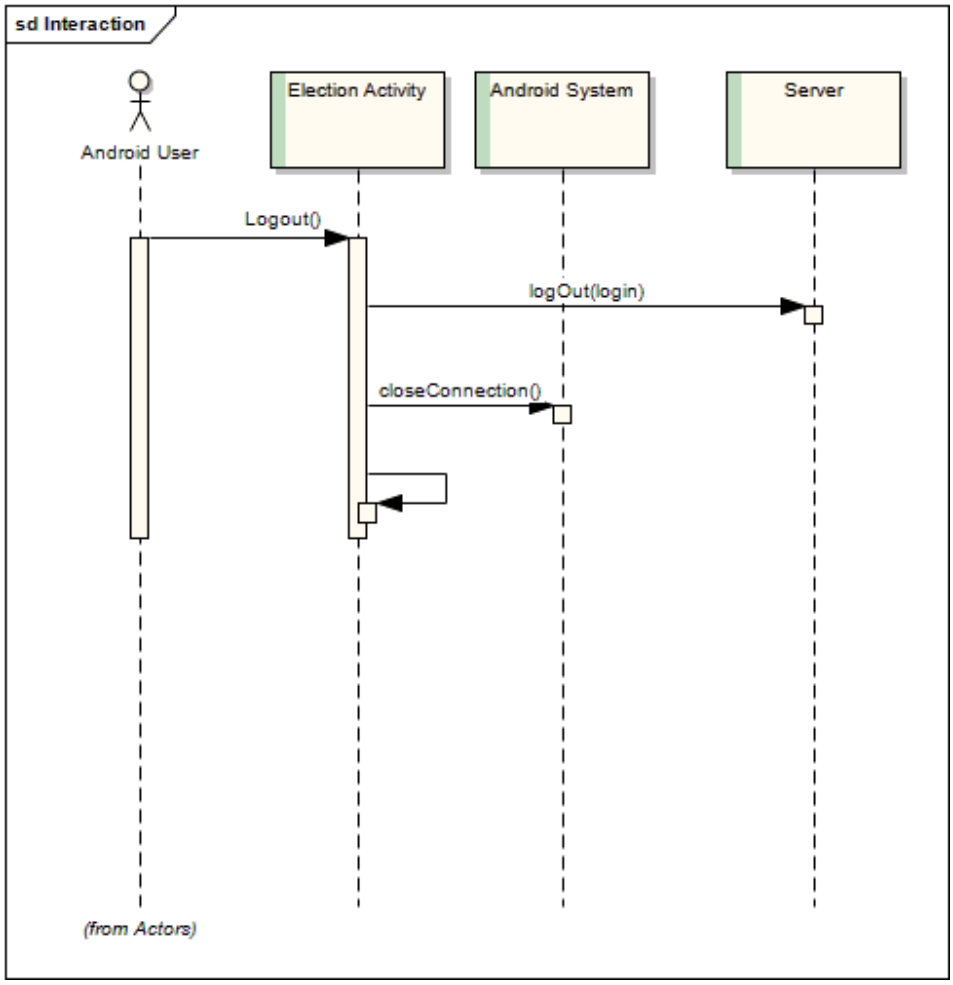

uc Exit Application
sd Interaction
Android System
Time
Android User
ElectionActivity
callOnDestroy()
callOnDestroy()
callOnDestoy()
logout()
onDestroy()
(from Actors)
(from Actors)
(from Actors)
Preconditions: Activity is started
Postconditions: User homescreen is shown
Use Case : Exit Application
1. Android OR Time or User chooses to quit
2. ALTERNATE user logged in
3. Activity logs out
4. Activity quits
5. End of use case

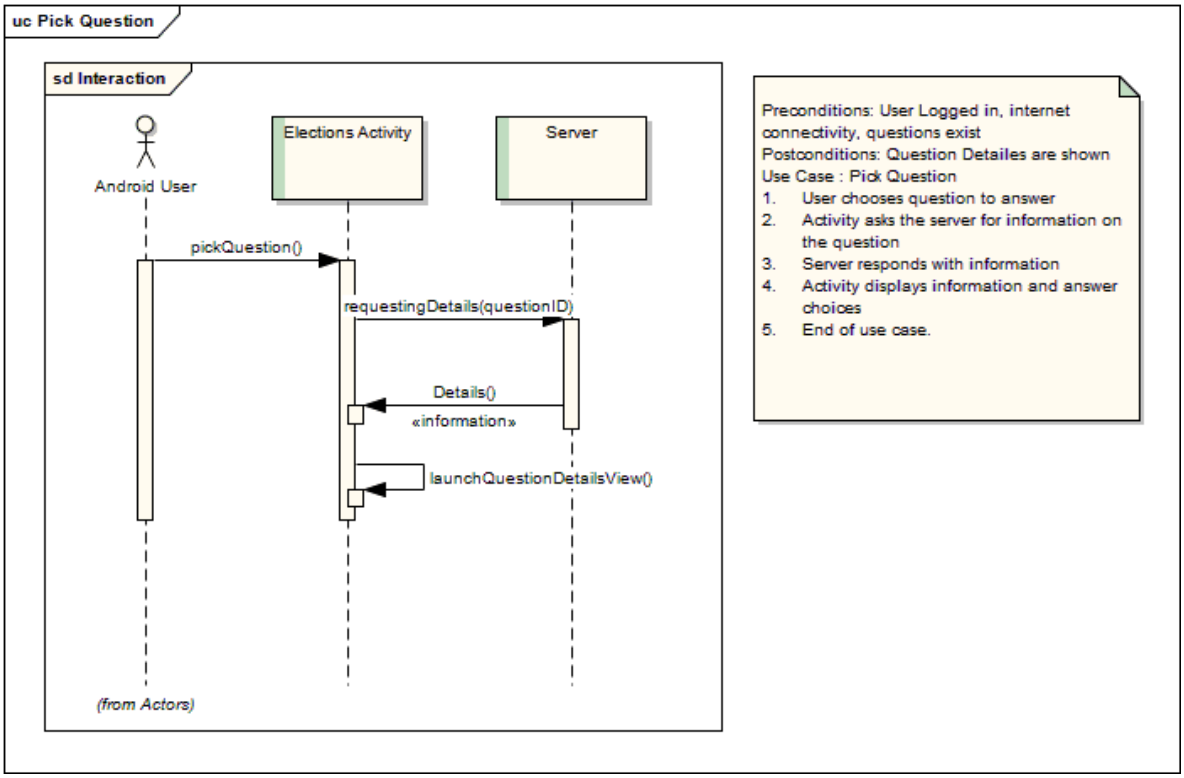

sd Interaction
Android System
Time
Android User
ElectionActivity
callOnDestroy()
callOnDestroy()
callOnDestoy()
logout()
onDestroy()
(from Actors)
(from Actors)
(from Actors)

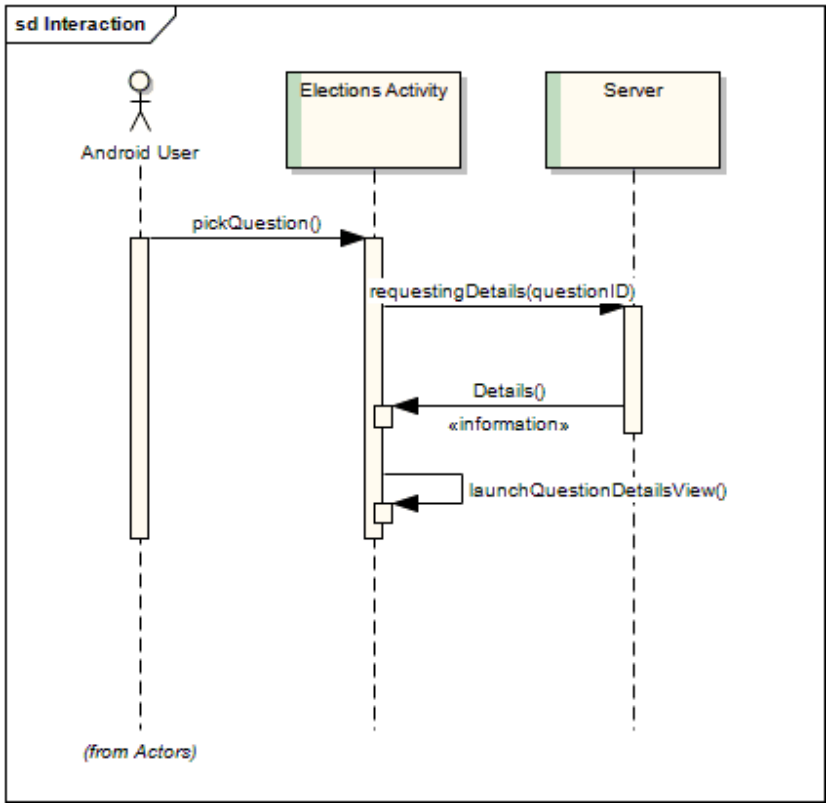

uc Vote on Question
sd Interaction
Android User
Election Activity
Server
voteFor(answerID)
vote(questionID, answerID)
alt Success
Success()
displaySuccess()
alt Failure
Fail()
DisplayFail()
showQuestionsView()
(from Actors)
Preconditions: Question Details shown, logged in, internet connection
Postconditions: Vote has been saved
Use Case : Vote on Question
1. User picks the answer and confirms
2. ElectionActivity sends the vote to the server
3. ALTERNATE Server Confirms
4. User is shown a success message
5. ALTERNATE Server Rejects
6. User is shown a failure message
7. ElectionActivity goes to questions view
8. End of use case.

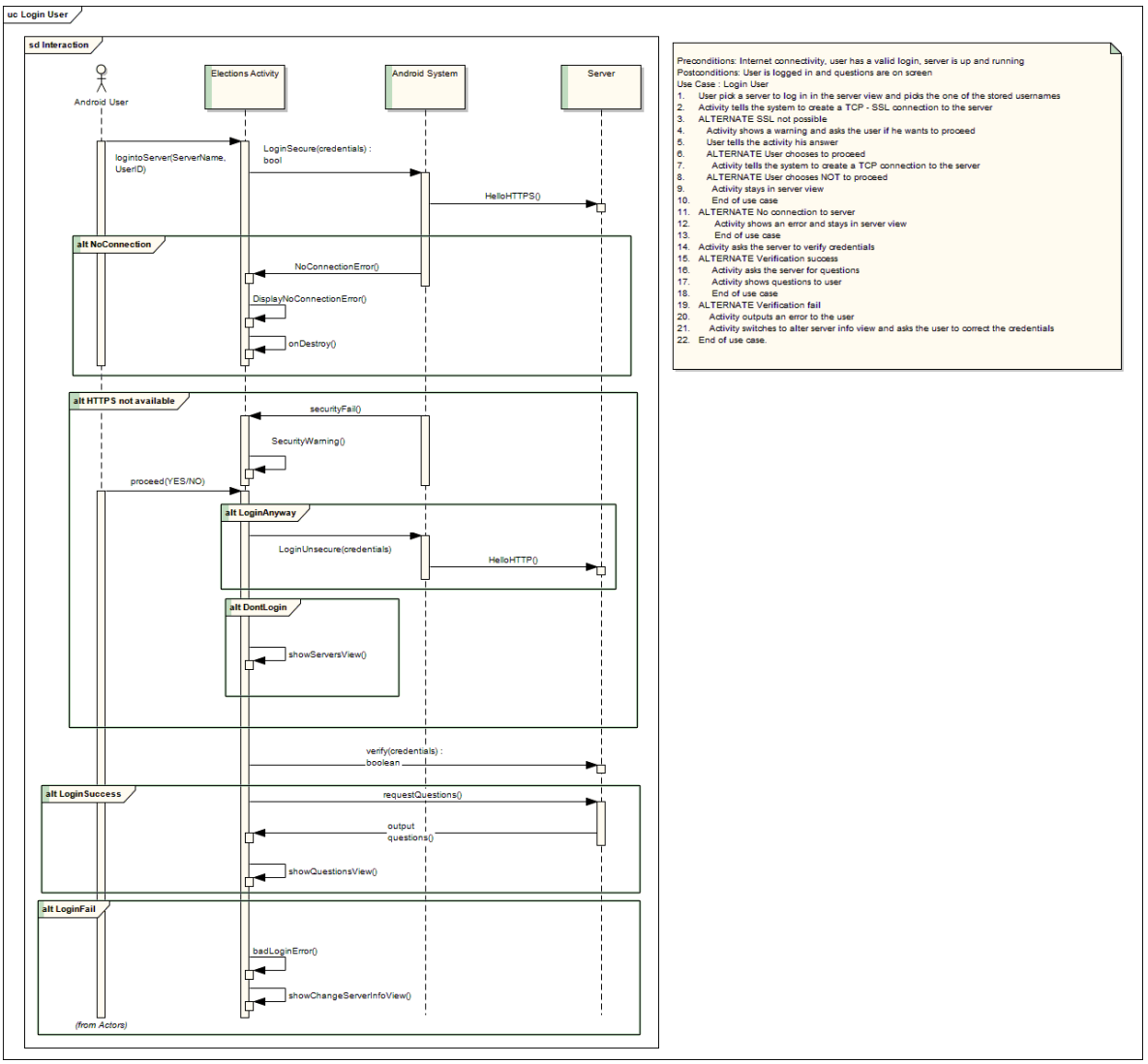

sd Interaction
Android User
Election Activity
Server
voteFor(answerID)
vote(questionID, answerID)
alt Success
Success()
displaySuccess()
alt Failure
Fail()
DisplayFail()
showQuestionsView()
(from Actors)


uc Logout User
sd Interaction
Android User
Election Activity
Android System
Server
Logout()
logOut(login)
closeConnection()
(from Actors)
Preconditions: Logged in, Internet Connection
Postconditions: User is logged out, server view is shown
Use Case : Logout
1. User chooses to logout
2. Activity sends the logout command to server
3. Activity asks android to close the connection
4. End of use case


sd Interaction
Android User
Election Activity
Android System
Server
Logout()
logOut(login)
closeConnection()
(from Actors)

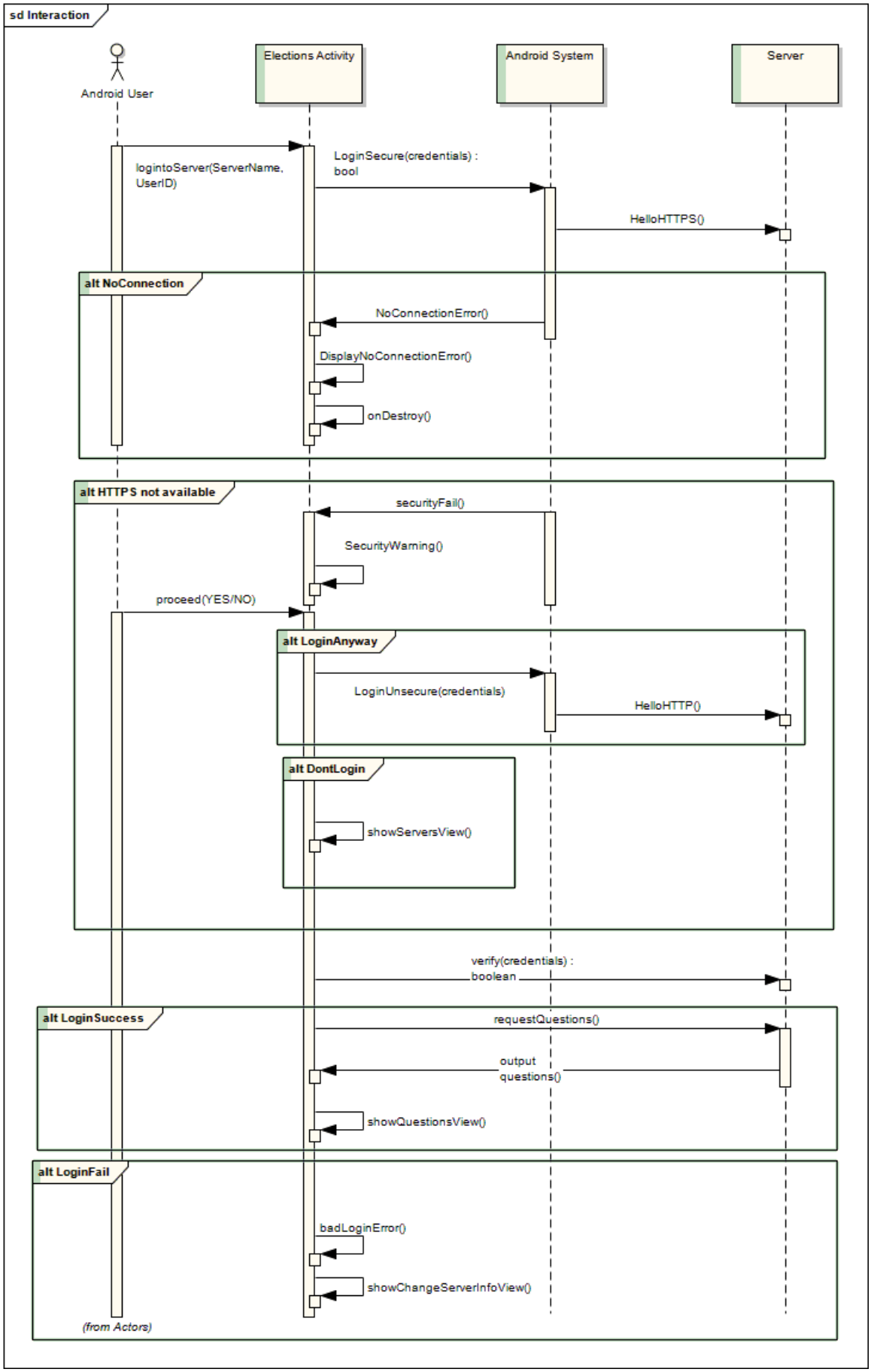

uc Pick Question
sd Interaction
Android User
Elections Activity
Server
pickQuestion()
requestingDetails(questionID)
Details()
«information»
launchQuestionDetailsView()
(from Actors)
Preconditions: User Logged in, internet connectivity, questions exist
Postconditions: Question Detailes are shown
Use Case : Pick Question
1. User chooses question to answer
2. Activity asks the server for information on the question
3. Server responds with information
4. Activity displays information and answer choices
5. End of use case.

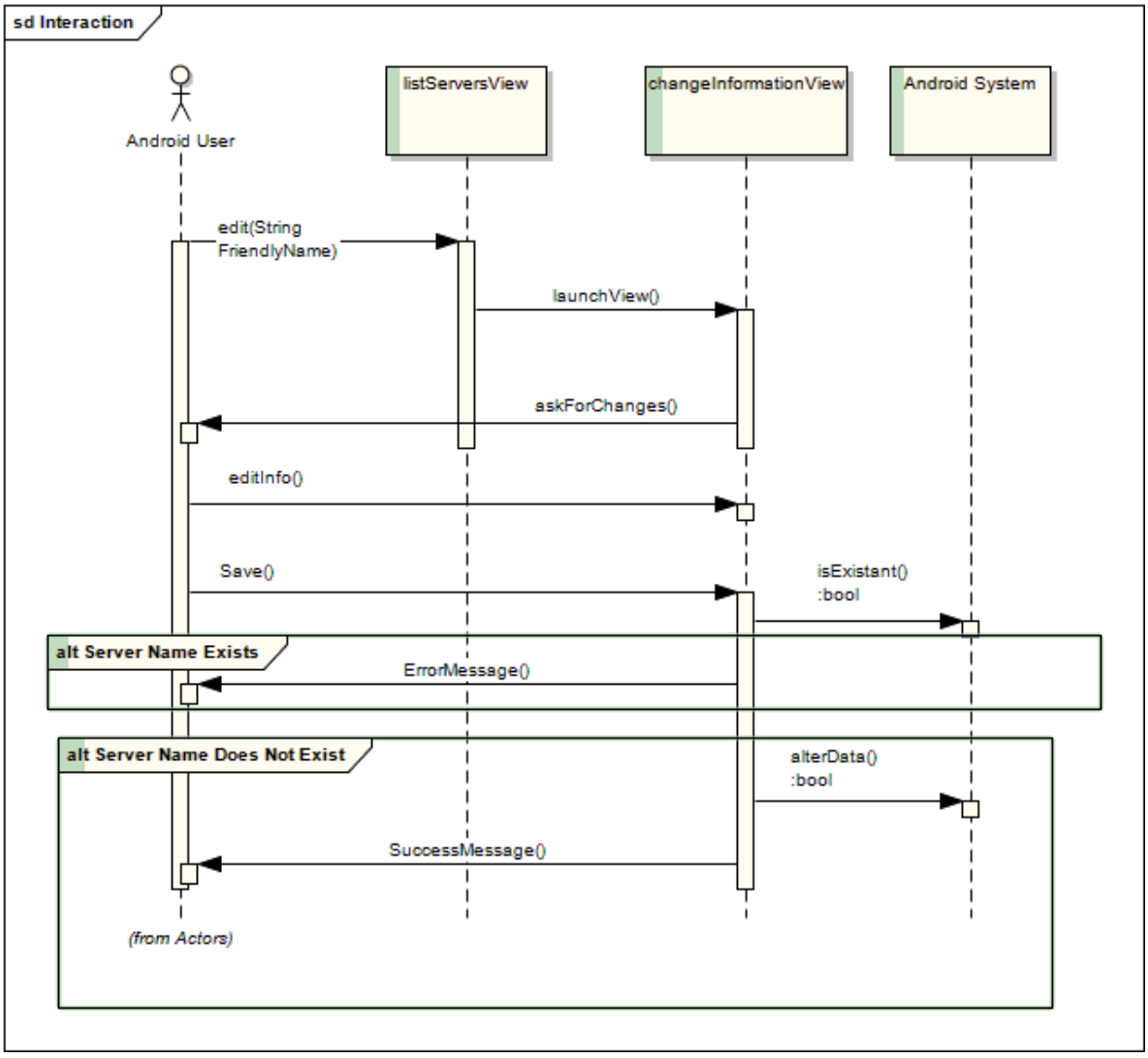

sd Interaction
Android User
Elections Activity
Server
pickQuestion()
requestingDetails(questionID)
Details()
«information»
launchQuestionDetailsView()
(from Actors)

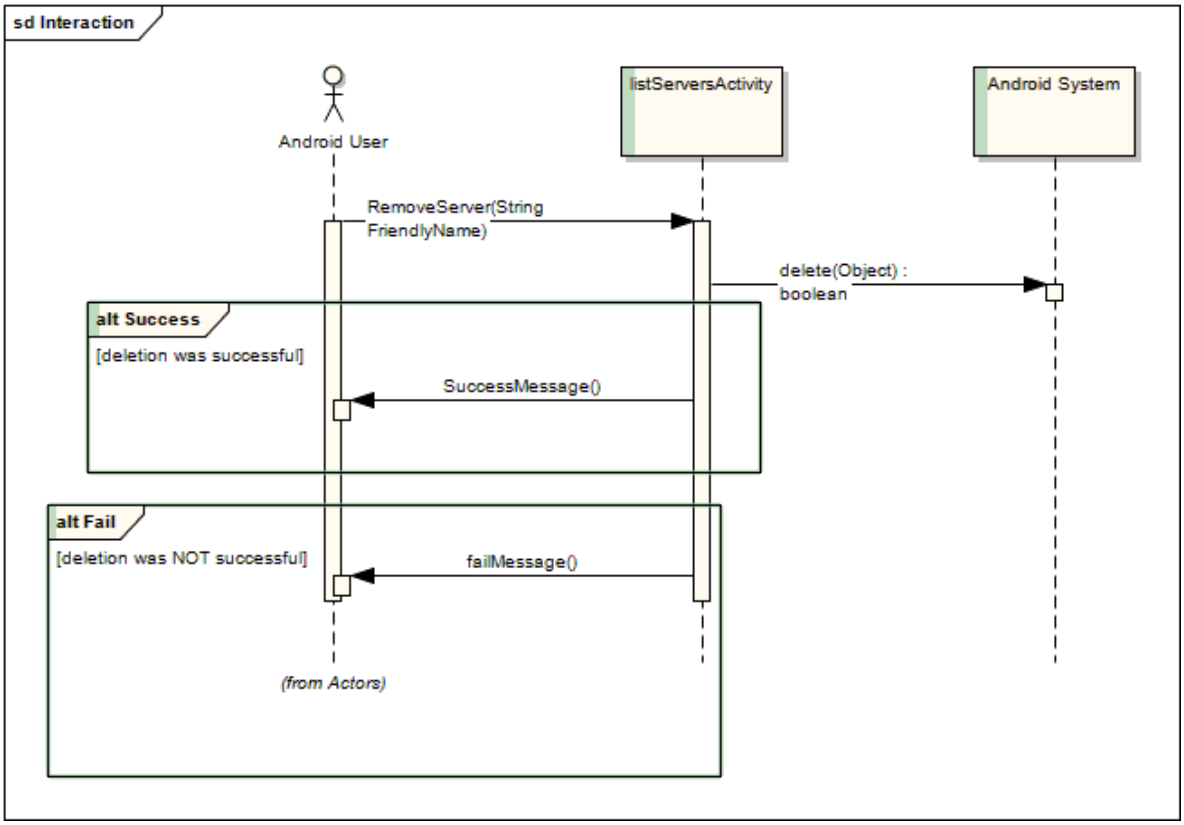


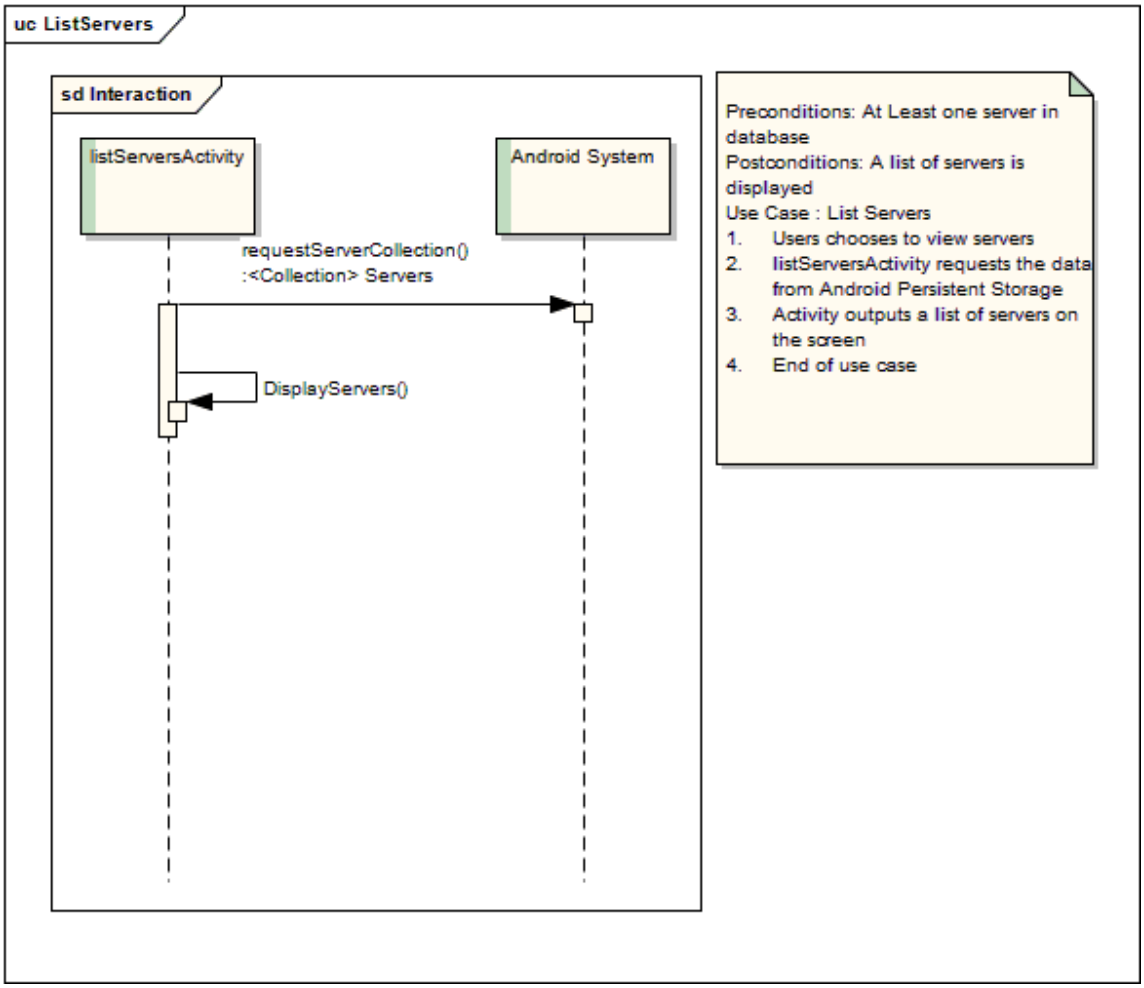
Preconditions: Internet connectivity, user has a valid login, server is up and running
Postconditions: User is logged in and questions are on screen
Use Case : Login User

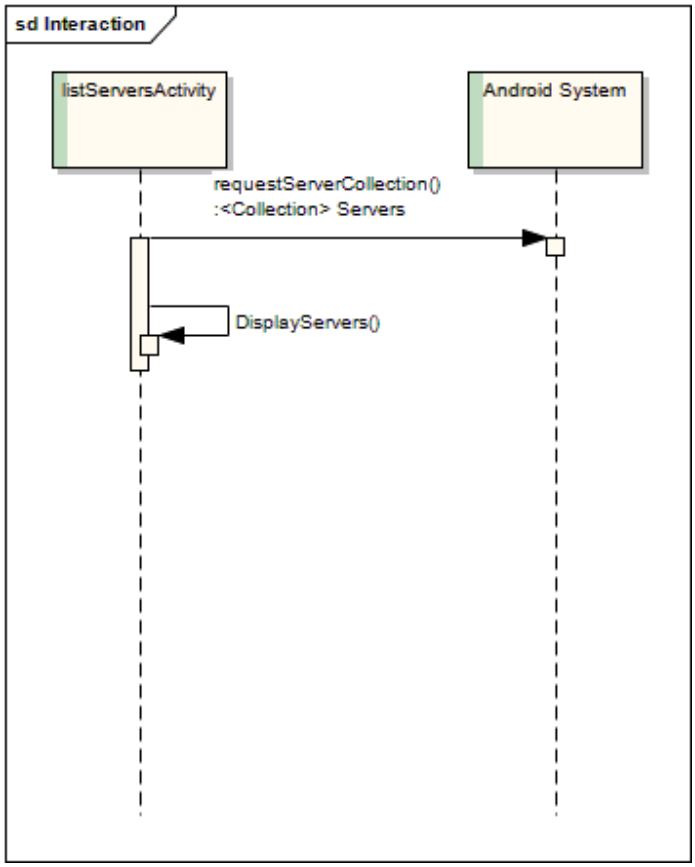
1. User pick a server to log in in the server view and picks the one of the stored usernames
2. Activity tells the system to create a TCP - SSL connection to the server
3. ALTERNATE SSL not possible
4. Activity shows a warning and asks the user if he wants to proceed
5. User tells the activity his answer
6. ALTERNATE User chooses to proceed
7. Activity tells the system to create a TCP connection to the server
8. ALTERNATE User chooses NOT to proceed
9. Activity stays in server view
10. End of use case
11. ALTERNATE No connection to server
12. Activity shows an error and stays in server view
13. End of use case
14. Activity asks the server to verify credentials
15. ALTERNATE Verification success
16. Activity asks the server for questions
17. Activity shows questions to user
18. End of use case
19. ALTERNATE Verification fail
20. Activity outputs an error to the user
21. Activity switches to alter server info view and asks the user to correct the credentials
22. End of use case.

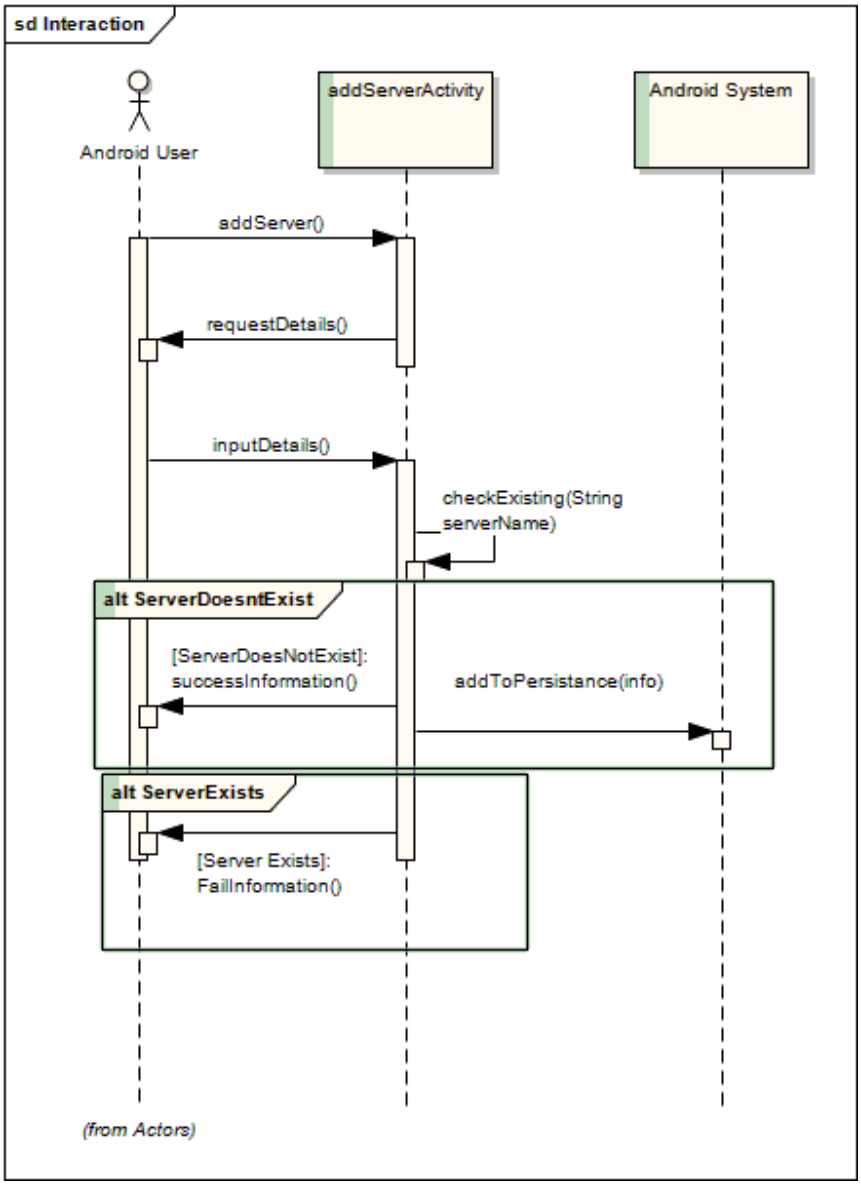

sd Interaction
Android User
Elections Activity
Android System
Server
logintoServer(ServerName, UserID)
LoginSecure(credentials) : bool
HelloHTTPS()
alt NoConnection
NoConnectionError()
DisplayNoConnectionError()
onDestroy()
alt HTTPS not available
securityFail()
SecurityWarning()
proceed(YES/NO)
alt LoginAnyway
LoginUnsecure(credentials)
HelloHTTP()
alt DontLogin
showServersView()
verify(credentials) : boolean
alt LoginSuccess
requestQuestions()
output questions()
showQuestionsView()
alt LoginFail
badLoginError()
showChangeServerInfoView()
(from Actors)

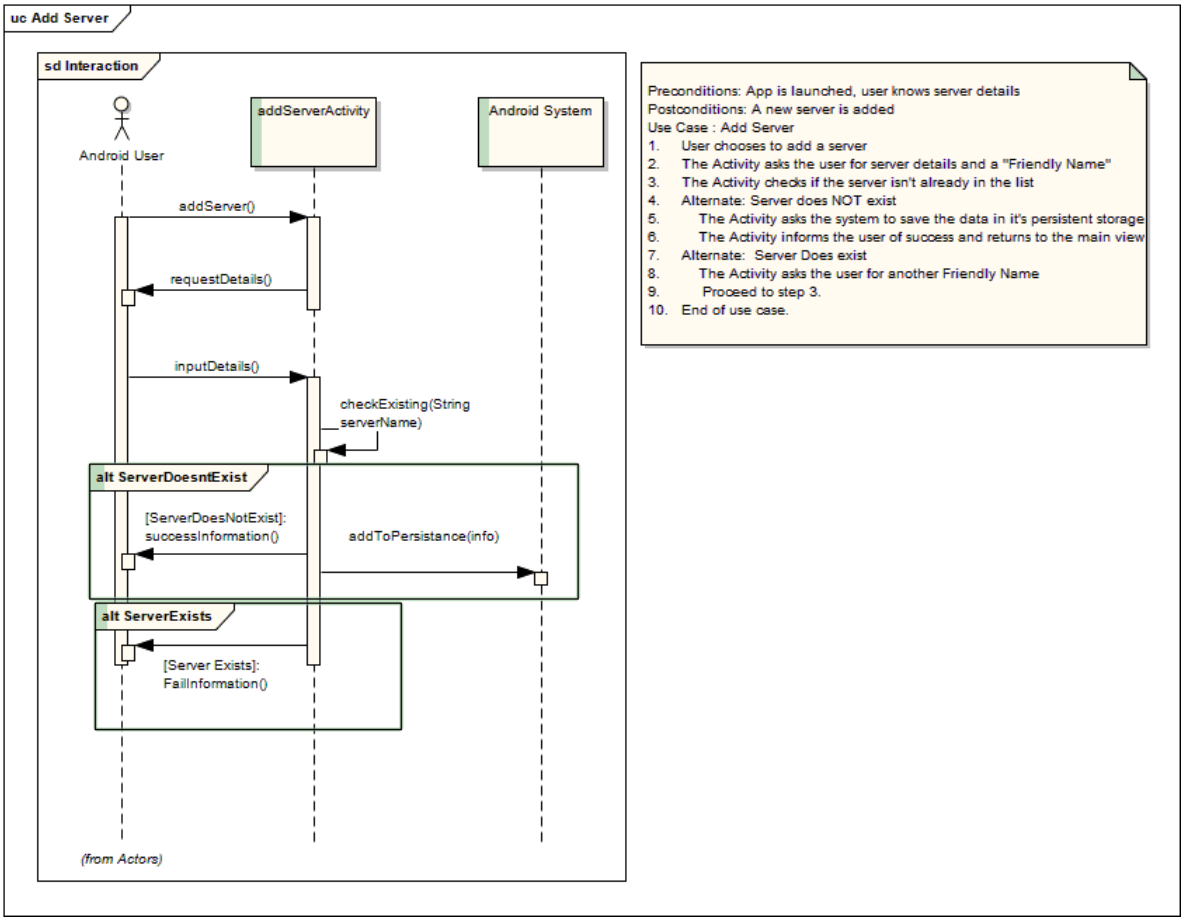

sd Interaction
Android User
listServersView
changeInformationView
Android System
edit(String FriendlyName)
launchView()
askForChanges()
editInfo()
Save()
isExistant() :bool
alt Server Name Exists
ErrorMessage()
alt Server Name Does Not Exist
alterData() :bool
SuccessMessage()
(from Actors)

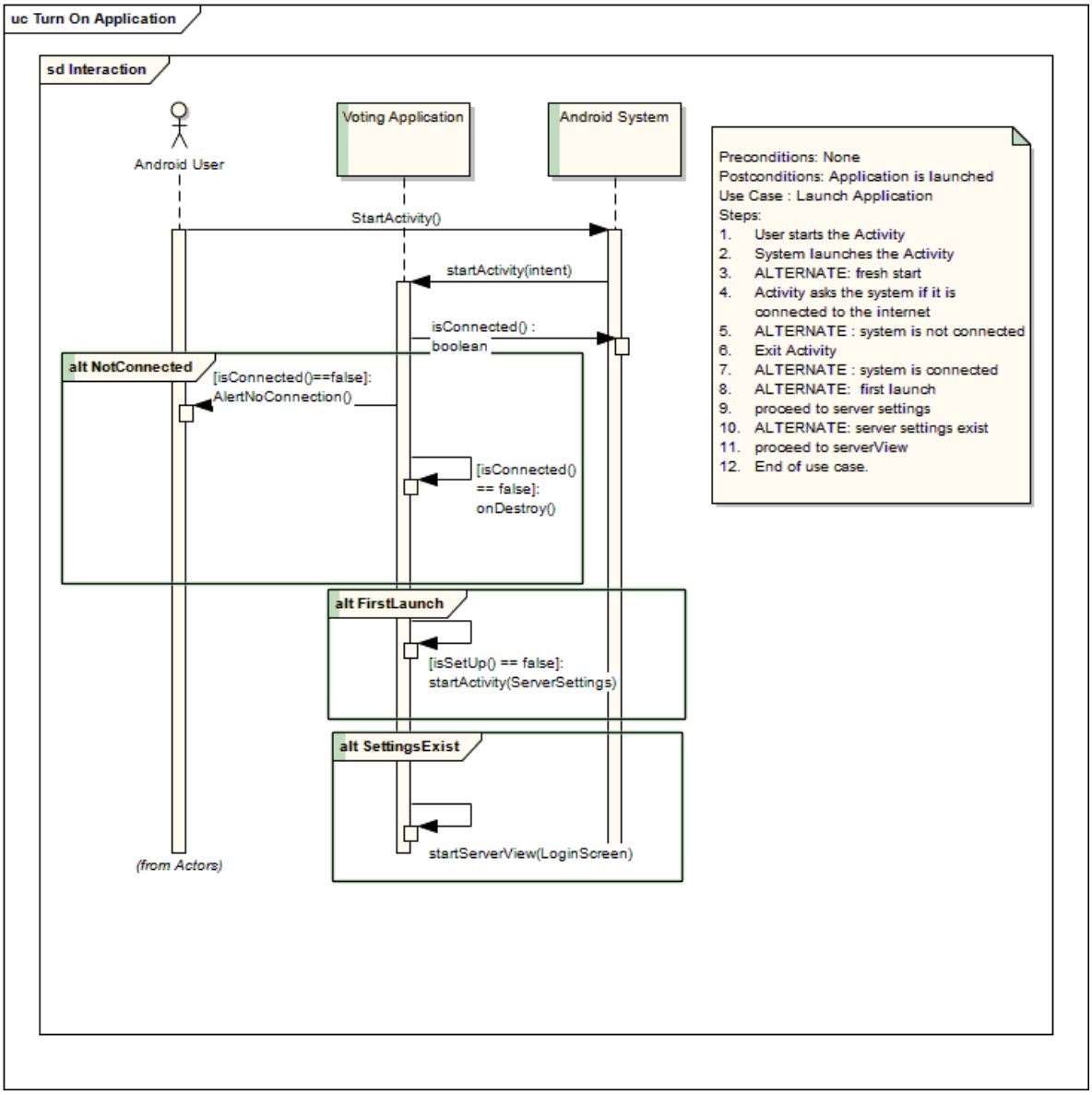

sd Interaction
Android User
listServersActivity
Android System
RemoveServer(String FriendlyName)
delete(Object) : boolean
alt Success
[deletion was successful]
SuccessMessage()
alt Fail
[deletion was NOT successful]
failMessage()
(from Actors)

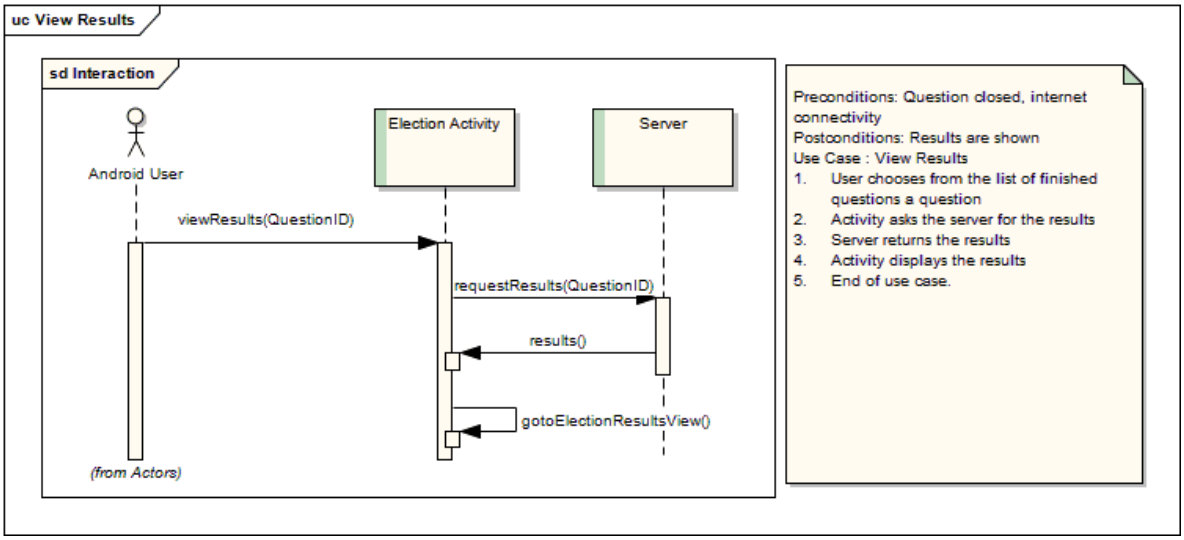

uc ListServers
sd Interaction
listServersActivity
Android System
requestServerCollection()
:<Collection> Servers
DisplayServers()
Preconditions: At Least one server in database
Postconditions: A list of servers is displayed
Use Case : List Servers
1. Users chooses to view servers
2. listServersActivity requests the data from Android Persistent Storage
3. Activity outputs a list of servers on the screen
4. End of use case

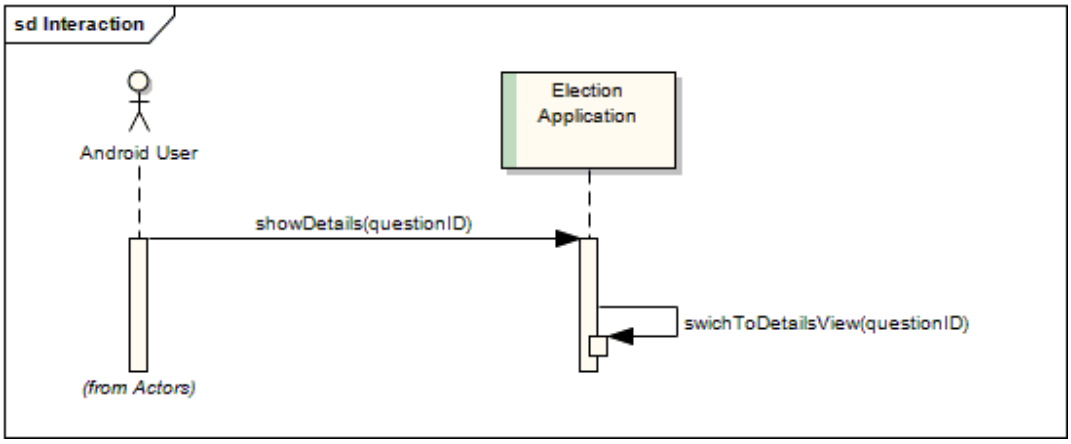

sd Interaction
listServersActivity
Android System
requestServerCollection()
:<Collection> Servers
DisplayServers()

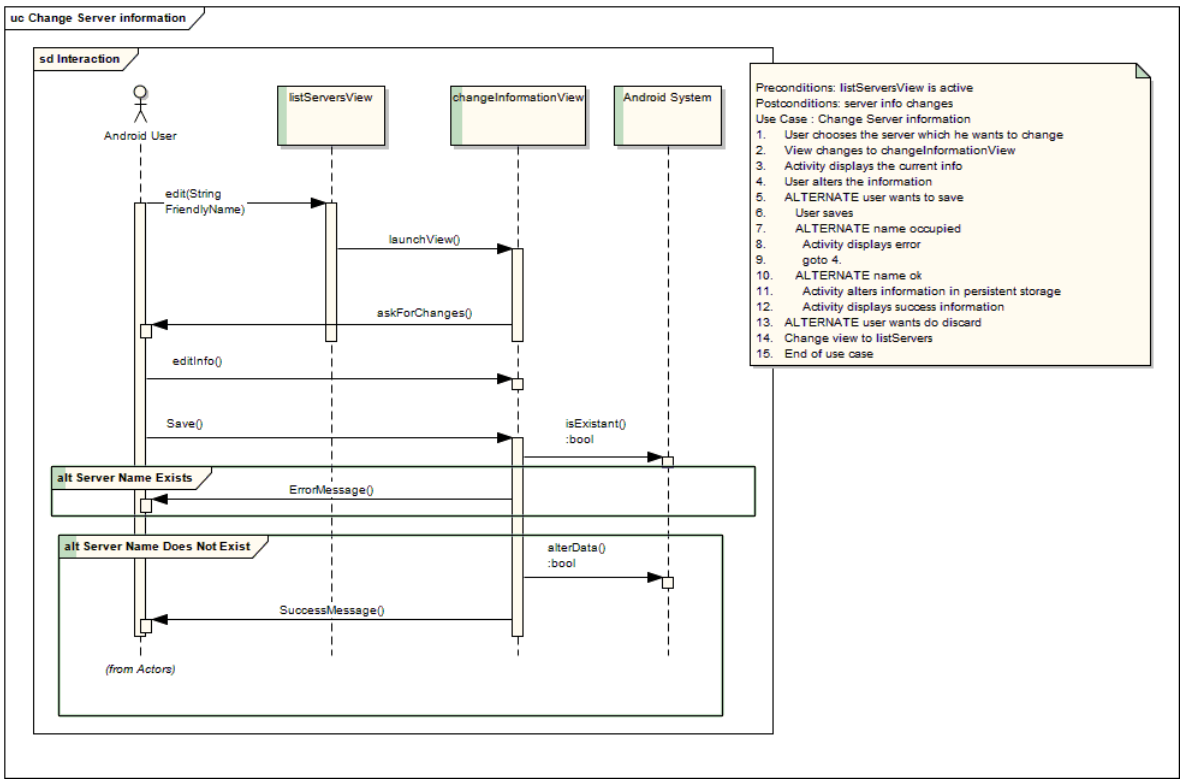

sd Interaction
Android User
addServerActivity
Android System
addServer()
requestDetails()
inputDetails()
checkExisting(String serverName)
alt ServerDoesntExist
[ServerDoesNotExist]: successInformation()
addToPersistance(info)
alt ServerExists
[Server Exists]: FailInformation()
(from Actors)


uc Add Server
sd Interaction
Android User
addServerActivity
Android System
addServer()
requestDetails()
inputDetails()
checkExisting(String serverName)
alt ServerDoesntExist
[ServerDoesNotExist]: successInformation()
addToPersistance(info)
alt ServerExists
[Server Exists]: FailInformation()
(from Actors)
Preconditions: App is launched, user knows server details
Postconditions: A new server is added
Use Case : Add Server
1. User chooses to add a server
2. The Activity asks the user for server details and a "Friendly Name"
3. The Activity checks if the server isn't already in the list
4. Alternate: Server does NOT exist
5. The Activity asks the system to save the data in it's persistent storage
6. The Activity informs the user of success and returns to the main view
7. Alternate: Server Does exist
8. The Activity asks the user for another Friendly Name
9. Proceed to step 3.
10. End of use case.


uc Turn On Application
sd Interaction
Android User
Voting Application
Android System
StartActivity()
startActivity(intent)
isConnected() :
boolean
alt NotConnected
[isConnected()==false]:
AlertNoConnection()
[isConnected()
== false]:
onDestroy()
alt FirstLaunch
[isSetUp() == false]:
startActivity(ServerSettings)
alt SettingsExist
startServerView(LoginScreen)
(from Actors)
Preconditions: None
Postconditions: Application is launched
Use Case : Launch Application
Steps:
1. User starts the Activity
2. System launches the Activity
3. ALTERNATE: fresh start
4. Activity asks the system if it is connected to the internet
5. ALTERNATE : system is not connected
6. Exit Activity
7. ALTERNATE : system is connected
8. ALTERNATE: first launch
9. proceed to server settings
10. ALTERNATE: server settings exist
11. proceed to serverView
12. End of use case.


uc View Results
sd Interaction
Android User
Election Activity
Server
viewResults(QuestionID)
requestResults(QuestionID)
results()
gotoElectionResultsView()
(from Actors)
Preconditions: Question closed, internet connectivity
Postconditions: Results are shown
Use Case : View Results
1. User chooses from the list of finished questions a question
2. Activity asks the server for the results
3. Server returns the results
4. Activity displays the results
5. End of use case.


sd Interaction
Android User
Election Application
showDetails(questionID)
swichToDetailsView(questionID)
(from Actors)


uc Change Server information
sd Interaction
Android User
listServersView
changeInformationView
Android System
edit(String FriendlyName)
launchView()
askForChanges()
editInfo()
Save()
isExistant() :bool
alt Server Name Exists
ErrorMessage()
alt Server Name Does Not Exist
alterData() :bool
SuccessMessage()
(from Actors)
Preconditions: listServersView is active
Postconditions: server info changes
Use Case : Change Server information
1. User chooses the server which he wants to change
2. View changes to changeInformationView
3. Activity displays the current info
4. User alters the information
5. ALTERNATE user wants to save
6. User saves
7. ALTERNATE name occupied
8. Activity displays error
9. goto 4.
10. ALTERNATE name ok
11. Activity alters information in persistent storage
12. Activity displays success information
13. ALTERNATE user wants do discard
14. Change view to listServers
15. End of use case