

FIU Sequence Diagrams

This document includes BDD features for FIU along with code for generating sequence diagrams and accompanying pictures.

Note:

You can input the code in this [website](#), and it will generate the sequence diagram.

FIU_1001_POST Consent.feature:

Scenario: 1001_1:

title FIU_1001_POST Consent.feature

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->FIU:1. Trigger the Post-Consent request [VR].

Runner<-FIU:2. Trigger Response [VR].

FIU->MockAA:3. Call the Post-Consent API [R].

Runner<-MockAA:4. Forward the consent request [VR].

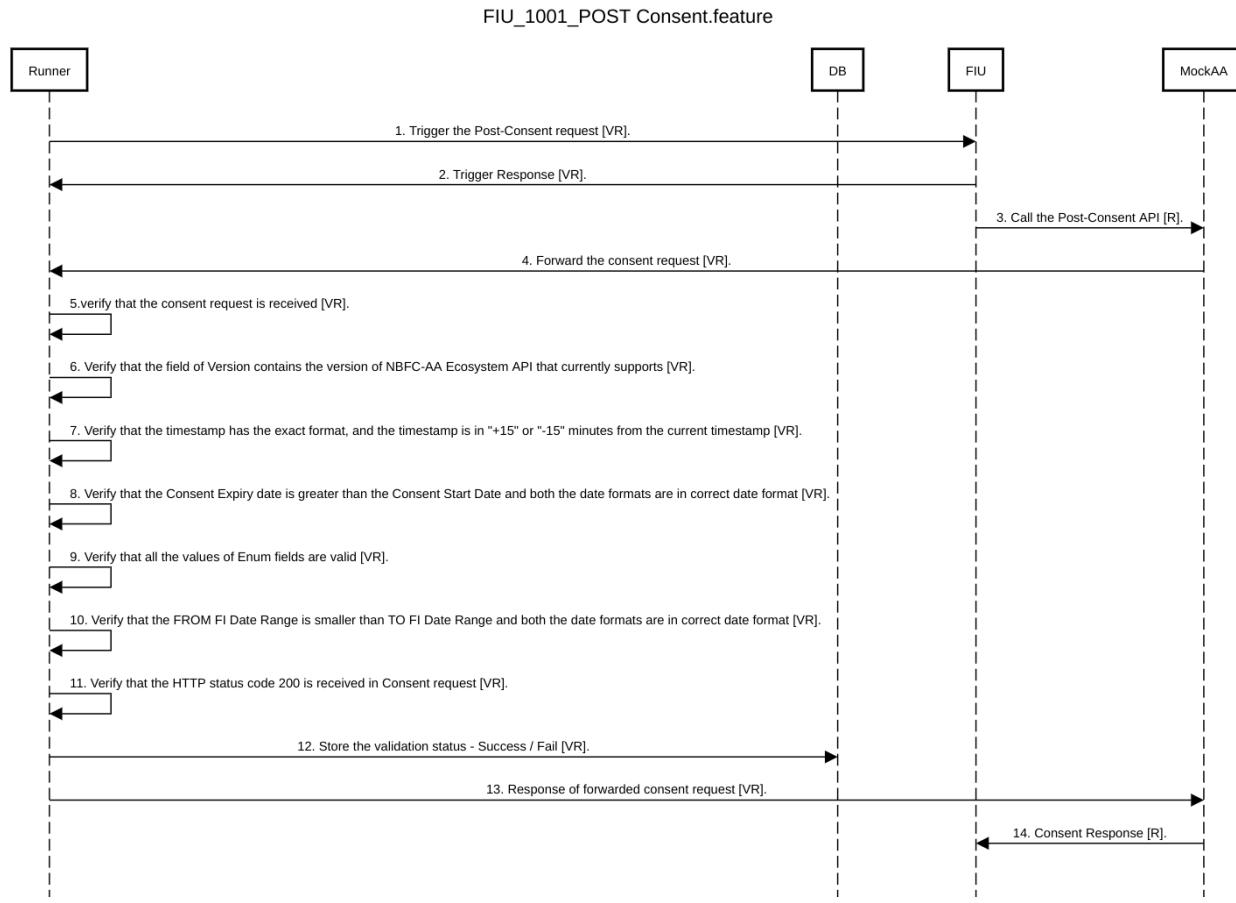
Runner->Runner:5.verify that the consent request is received [VR].

Runner->Runner:6. Verify that the field of Version contains the version of NBFC-AA Ecosystem API that currently supports [VR].

Runner->Runner:7. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current timestamp [VR].

Runner->Runner:8. Verify that the Consent Expiry date is greater than the Consent Start Date and both the date formats are in correct date format [VR].

- Runner->Runner:9. Verify that all the values of Enum fields are valid [VR].
- Runner->Runner:10. Verify that the FROM FI Date Range is smaller than TO FI Date Range and both the date formats are in correct date format [VR].
- Runner->Runner:11. Verify that the HTTP status code 200 is received in Consent request [VR].
- Runner -> DB: 12. Store the validation status - Success / Fail [VR].
- Runner->MockAA:13. Response of forwarded consent request [VR].
- FIU<-MockAA:14. Consent Response [R].



FIU_1002_GET ConsentHandle.feature

Scenario:1002_1

title FIU_1002_GET ConsentHandle.feature

participant Runner

participant DB

participant FIU(SUT)

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->FIU(SUT):1. Trigger the Post-Consent request [VR].

Runner<-FIU(SUT):2. Trigger Response [VR].

FIU(SUT)->MockAA:3. Call the Post-Consent API [R].

FIU(SUT)<-MockAA:4. Consent Response [R].

MockAA->DB:5.Store Consent handle ID [VR]

Runner ->FIU(SUT):6.Trigger the Get-Consent handle [VR].

Runner<-FIU(SUT):7.Trigger Response [VR]

FIU(SUT) ->MockAA:8. Call the Get-Consent handle Api [R].

MockAA->Runner:9. Forward the consent handle request [VR].

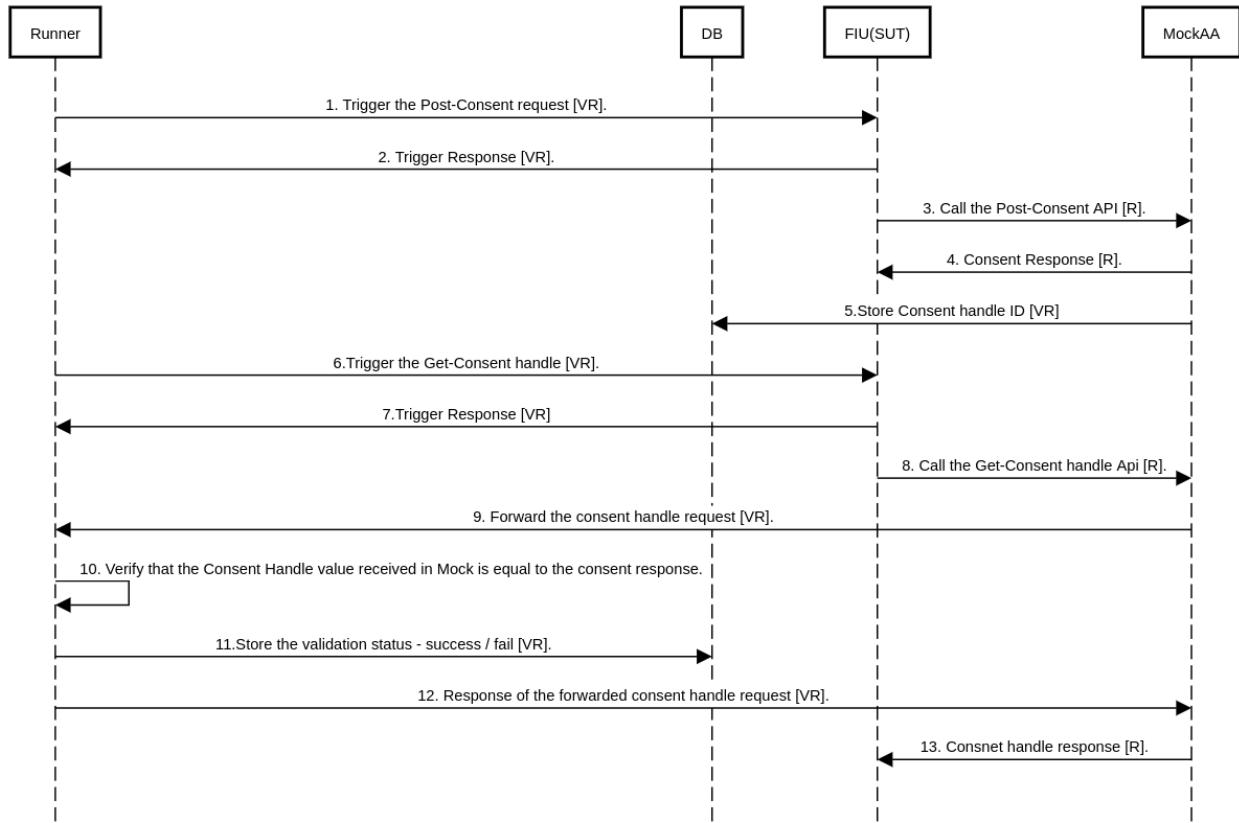
Runner ->Runner : 10. Verify that the Consent Handle value received in Mock is equal to the consent response.

Runner -> DB:11.Store the validation status - success / fail [VR].

Runner->MockAA:12. Response of the forwarded consent handle request [VR].

MockAA ->FIU(SUT):13. Consnet handle response [R].

FIU_1002_GET ConsentHandle.feature



FIU_1003_GET_ConsentId.feature

Scenario:1003_1

title FIU_1003_GET_ConsentId.feature Scenario 1003_1

participant Runner
 participant DB
 participant FIU
 participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

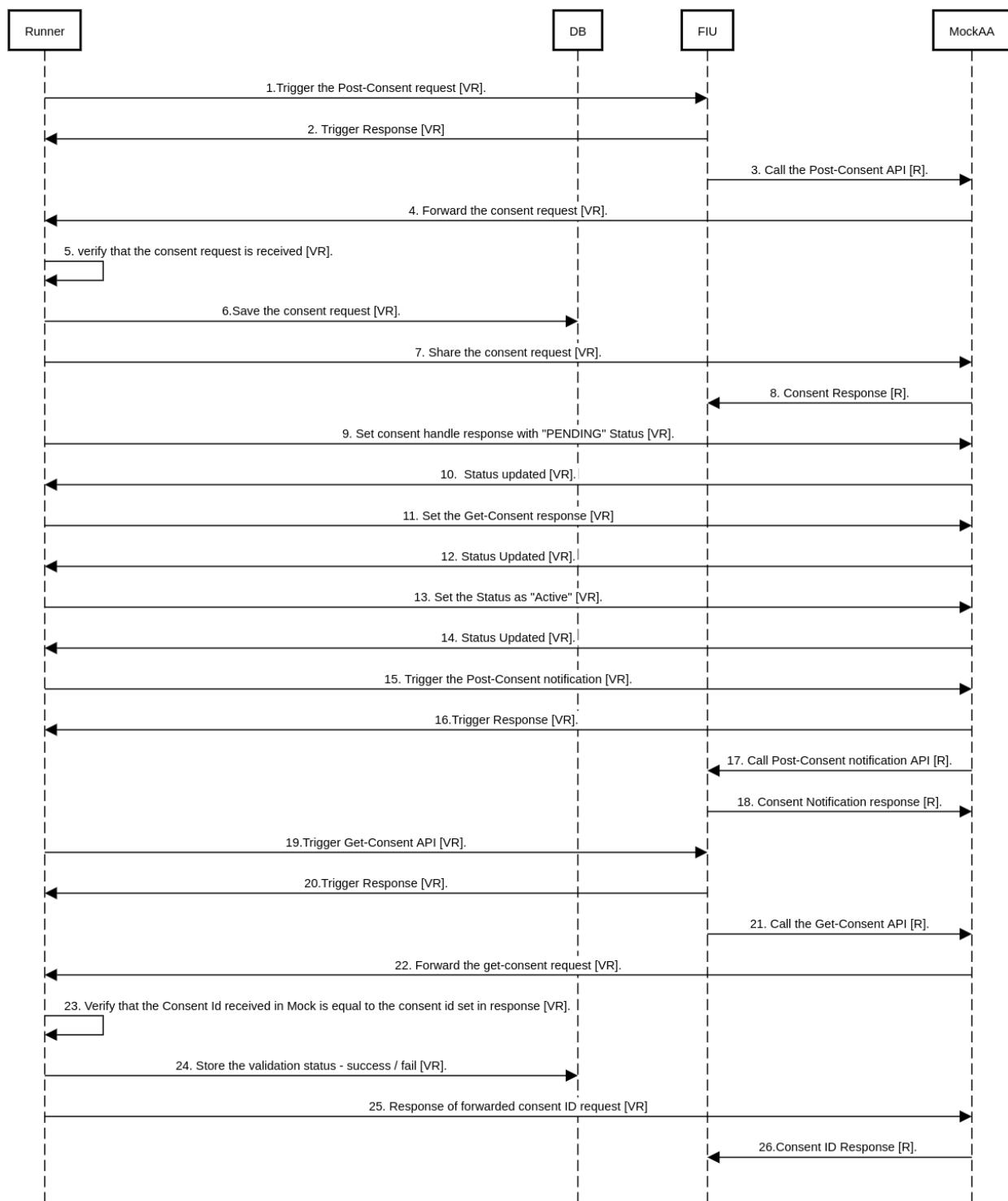
Runner->MockAA:1. Set the post consent response [VR]

Runner<-MockAA:2.Status Updated [VR].

Runner->FIU:3.Trigger the Post-Consent request [VR].

Runner<-FIU:4. Trigger Response [VR]
FIU->MockAA:5. Call the Post-Consent API [R].
Runner<-MockAA:6. Forward the consent request [VR].
Runner->Runner:7. verify that the consent request is received [VR].
Runner -> DB:8. Save the consent request [VR].
Runner->MockAA:9. Share the consent request [VR].
FIU<-MockAA:10. Consent Response [R].
Runner->MockAA:11. Set consent handle response with "PENDING" Status [VR].
Runner<-MockAA:12. Status updated [VR].
Runner->MockAA:13. Set the Get-Consent response [VR]
Runner<-MockAA:14. Status Updated [VR].
Runner->MockAA:15. Set the Status as "Active" [VR].
Runner<-MockAA:16. Status Updated [VR].
Runner->MockAA:17. Trigger the Post-Consent notification [VR].
Runner<-MockAA:18. Trigger Response [VR].
MockAA->FIU:19. Call Post-Consent notification API [R].
FIU->MockAA:20. Consent Notification response [R].
Runner->FIU:21. Trigger Get-Consent API [VR].
Runner<-FIU:22. Trigger Response [VR].
FIU->MockAA:23. Call the Get-Consent API [R].
MockAA->Runner:24. Forward the get-consent request [VR].
Runner -> Runner: 25. Verify that the Consent Id received in Mock is equal to the consent id set in response [VR].
Runner -> DB:26. Store the validation status - success / fail [VR].
Runner->MockAA:27. Response of forwarded consent ID request [VR]
FIU<-MockAA:28. Consent ID Response [R].

FIU_1003_GET_ConsentId.feature



FIU_1004_POST_ConsentNotification.feature

Scenario:1004_1

title FIU_1004_POST_ConsentNotification.feature Scenario 1004_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent response [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set the status as "Active" [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21. Forward the consent notification response [VR].

Runner ->Runner:22. Verify that the Response code 200 is displayed [VR].

Runner->Runner:23.Verify that the field of Version contains the version of NBFC-MockAA Ecosystem API that currently supports [VR].

Runner->Runner:24.Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current timestamp [VR].

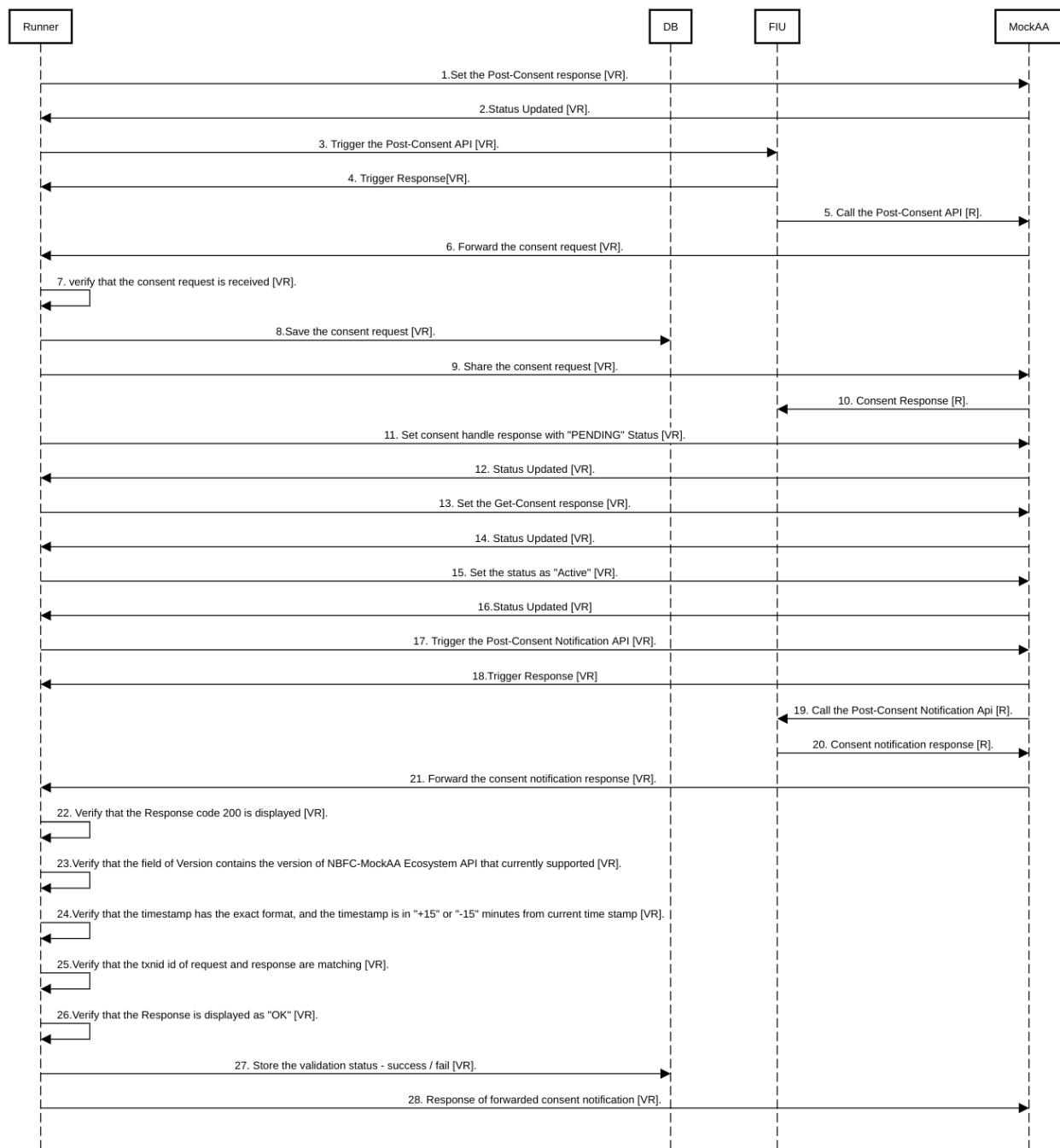
Runner->Runner:25.Verify that the txnid id of request and response are matching [VR].

Runner->Runner:26.Verify that the Response is displayed as "OK" [VR].

Runner -> DB : 27. Store the validation status - success / fail [VR].

Runner->MockAA:28. Response of forwarded consent notification [VR].

FIU_1004_POST_ConsentNotification.feature Scenario 1004_1



FIU_1005_POST_ConsentNotification_AltAA.feature

Scenario:1005_1

title FIU_1005_POST_ConsentNotification_AltAA.feature

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent response [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set the status as "Active" with an alternate AA Id [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21. Forward the consent notification response [VR].

Runner ->Runner:22.Verify that the Response code 400 is displayed [VR].

Runner->Runner:23. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:24. Verify that the Error code is InvalidNotifier [VR].

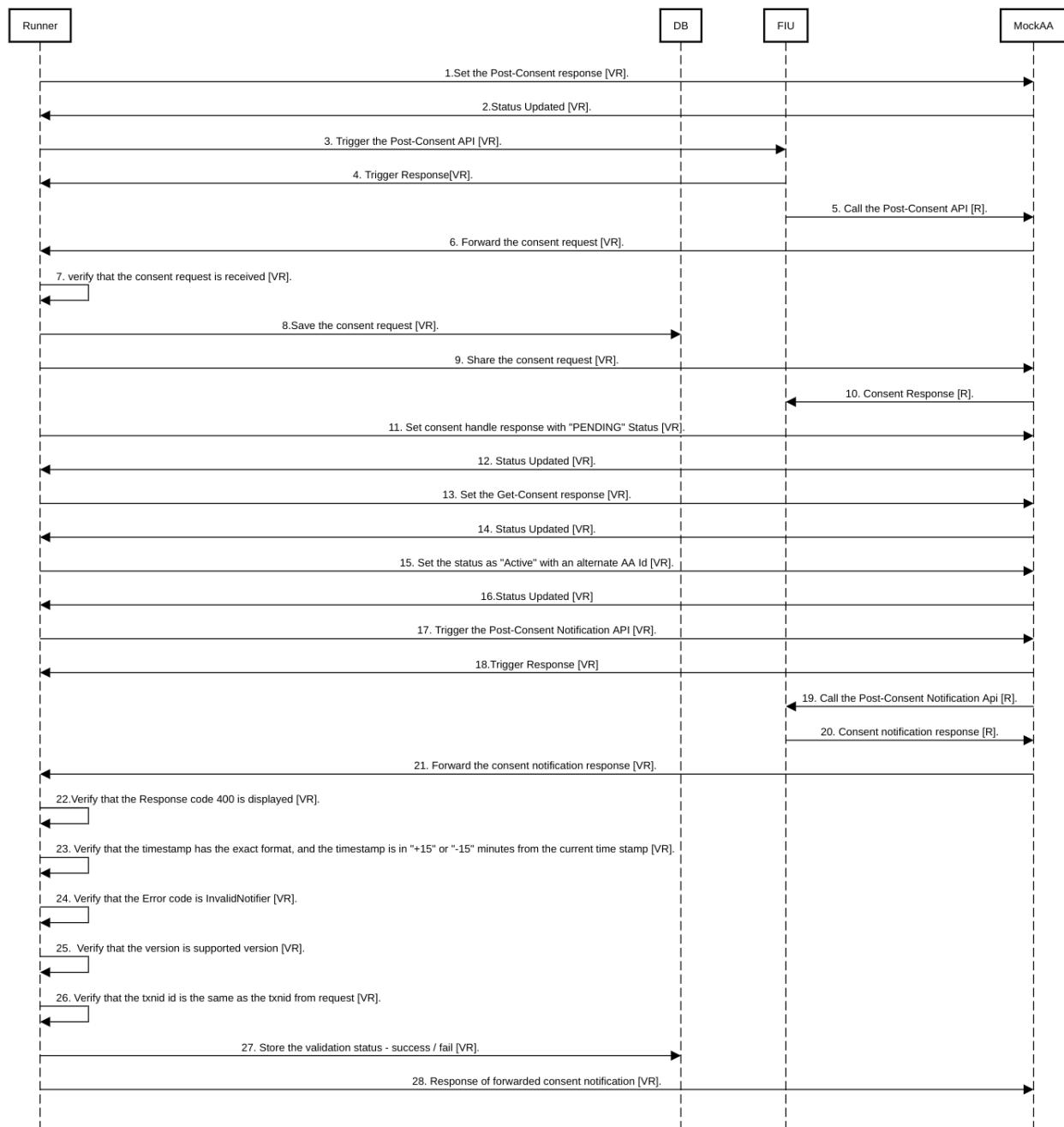
Runner->Runner:25. Verify that the version is supported version [VR].

Runner->Runner:26. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB : 27. Store the validation status – success / fail [VR].

Runner->MockAA:28. Response of forwarded consent notification [VR].

FIU_1005_POST_ConsentNotification_AltAA.feature



FIU_1006_POST_ConsentNotification_FIP.feature

Scenario:1006_1

title FIU_1006_POST_ConsentNotification_FIP.feature

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent response [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set the status as "Active" with Notifier Type as FIP [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21. Forward the consent notification response [VR].

Runner ->Runner:22.Verify that the Response code 400 is displayed [VR].

Runner->Runner:23. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:24. Verify that the Error code is InvalidNotifier [VR].

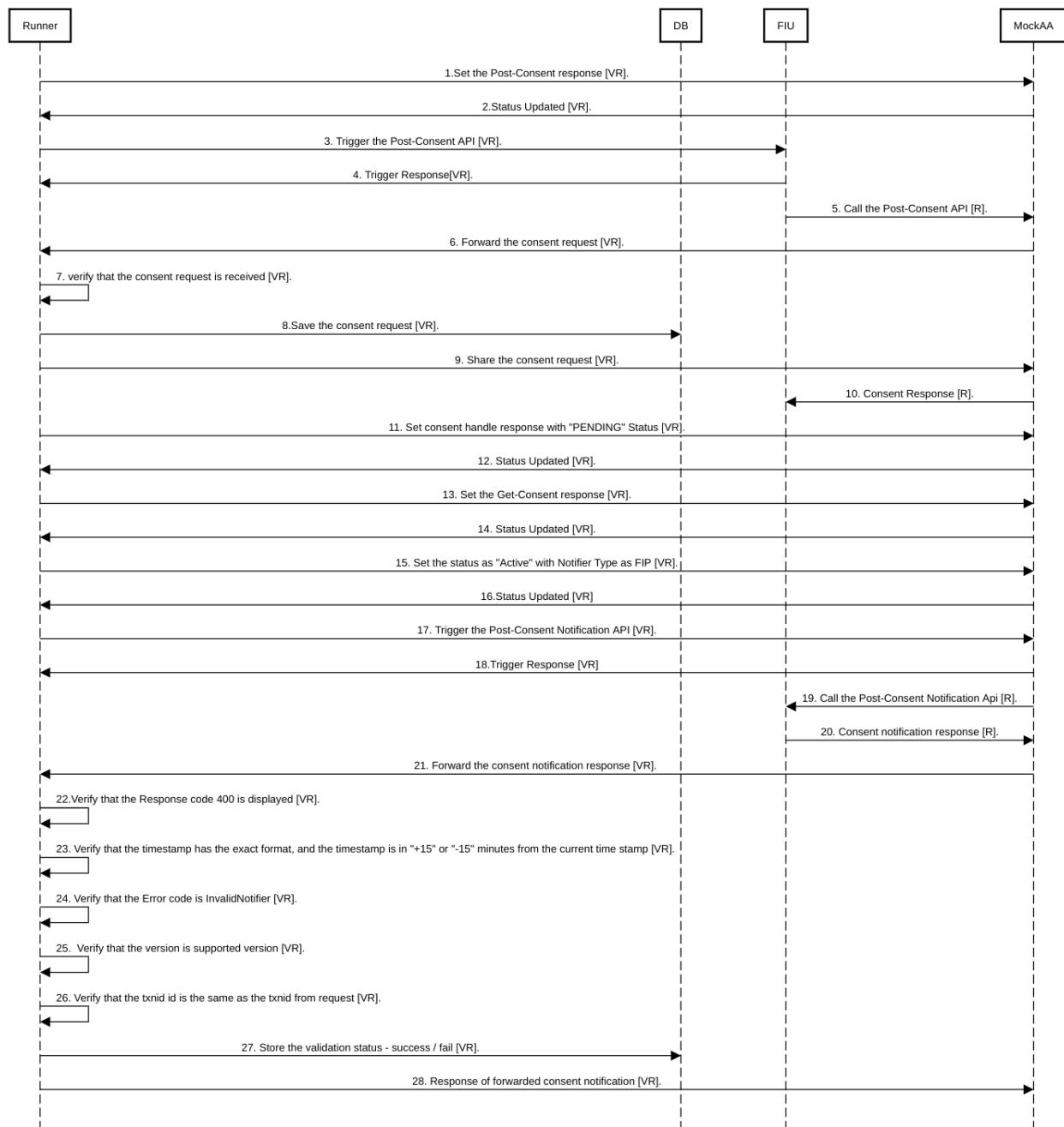
Runner->Runner:25. Verify that the version is supported version [VR].

Runner->Runner:26. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB : 27. Store the validation status – success / fail [VR].

Runner->MockAA:28. Response of forwarded consent notification [VR].

FIU_1006_POST_ConsentNotification_FIP.feature



FIU_1007_POST_ConsentNotification.feature

Scenario:1007_1

title FIU_1007_POST_ConsentNotification.feature Scenario 1007_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent request [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Remove the last character from consent id[VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21. Forward the consent notification response [VR].

Runner ->Runner:22.Verify that the Response code 400 is displayed [VR].

Runner->Runner:23. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:24. Verify that the Error code is InvalidConsentId [VR].

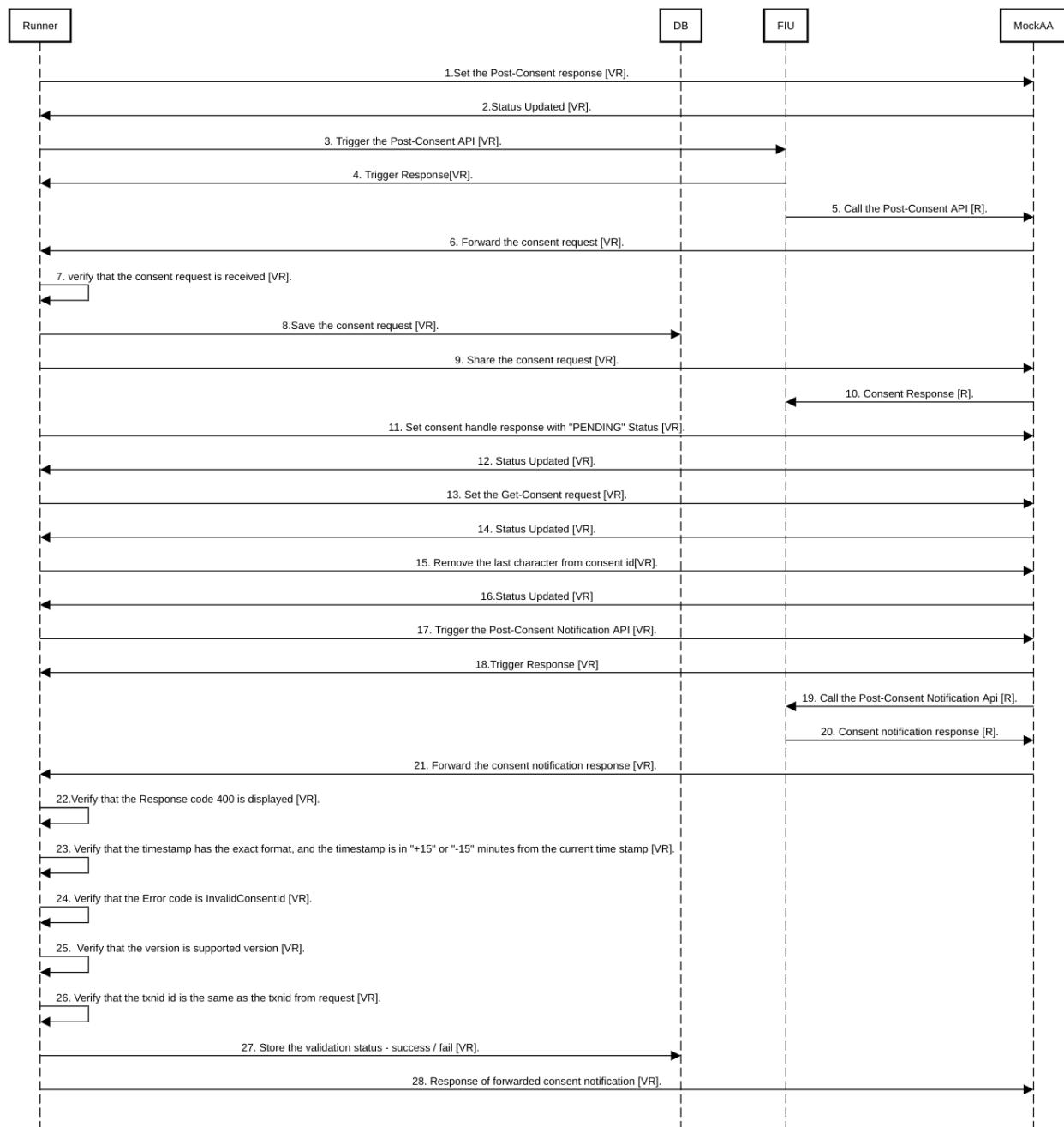
Runner->Runner:25. Verify that the version is supported version [VR].

Runner->Runner:26. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB : 27. Store the validation status – success / fail [VR].

Runner->MockAA:28. Response of forwarded consent notification [VR].

FIU_1007_POST _ConsentNotification.feature Scenario 1007_1



Scenario:1007_2

title FIU_1007_POST_ConsentNotification.feature Scenario 1007_2

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent request [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set the consent ID of another consent request [VR].

Runner<-MockAA:16.Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18.Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21. Forward the consent notification response [VR].

Runner ->Runner:22.Verify that the Response code 400 is displayed [VR].

Runner->Runner:23. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:24. Verify that the Error code is InvalidConsentId [VR].

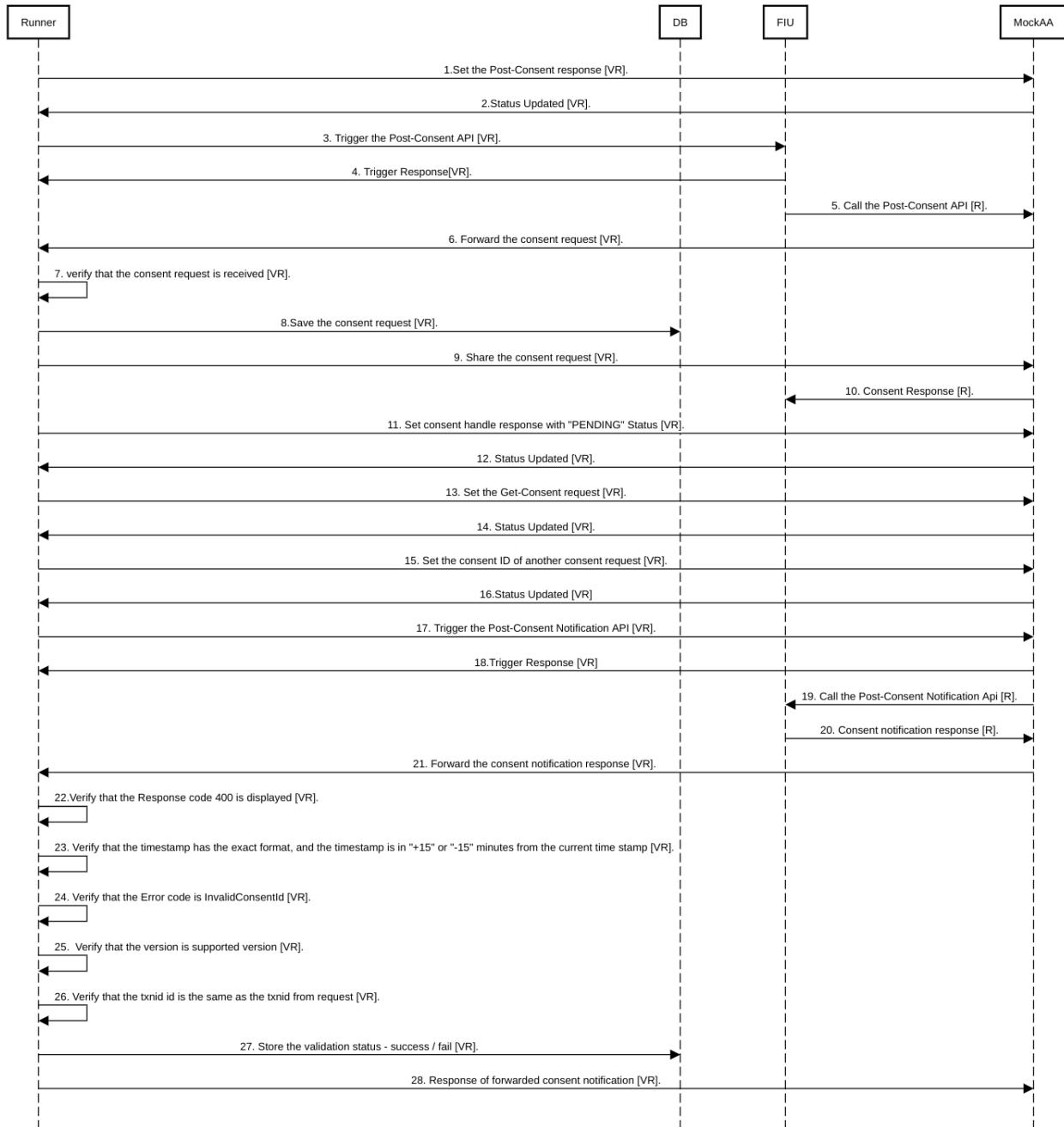
Runner->Runner:25. Verify that the version is supported version [VR].

Runner->Runner:26. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB : 27. Store the validation status – success / fail [VR].

Runner->MockAA:28. Response of forwarded consent notification [VR].

FIU_1007_POST _ConsentNotification.feature Scenario 1007_2



Scenario:1007_3

title FIU_1007_POST_ConsentNotification.feature Scenario 1007_3

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent request [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Add one character at the end of consent id [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21. Forward the consent notification response [VR].

Runner ->Runner:22. Verify that the Response code 400 is displayed [VR].

Runner->Runner:23. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:24. Verify that the Error code is InvalidConsentId [VR].

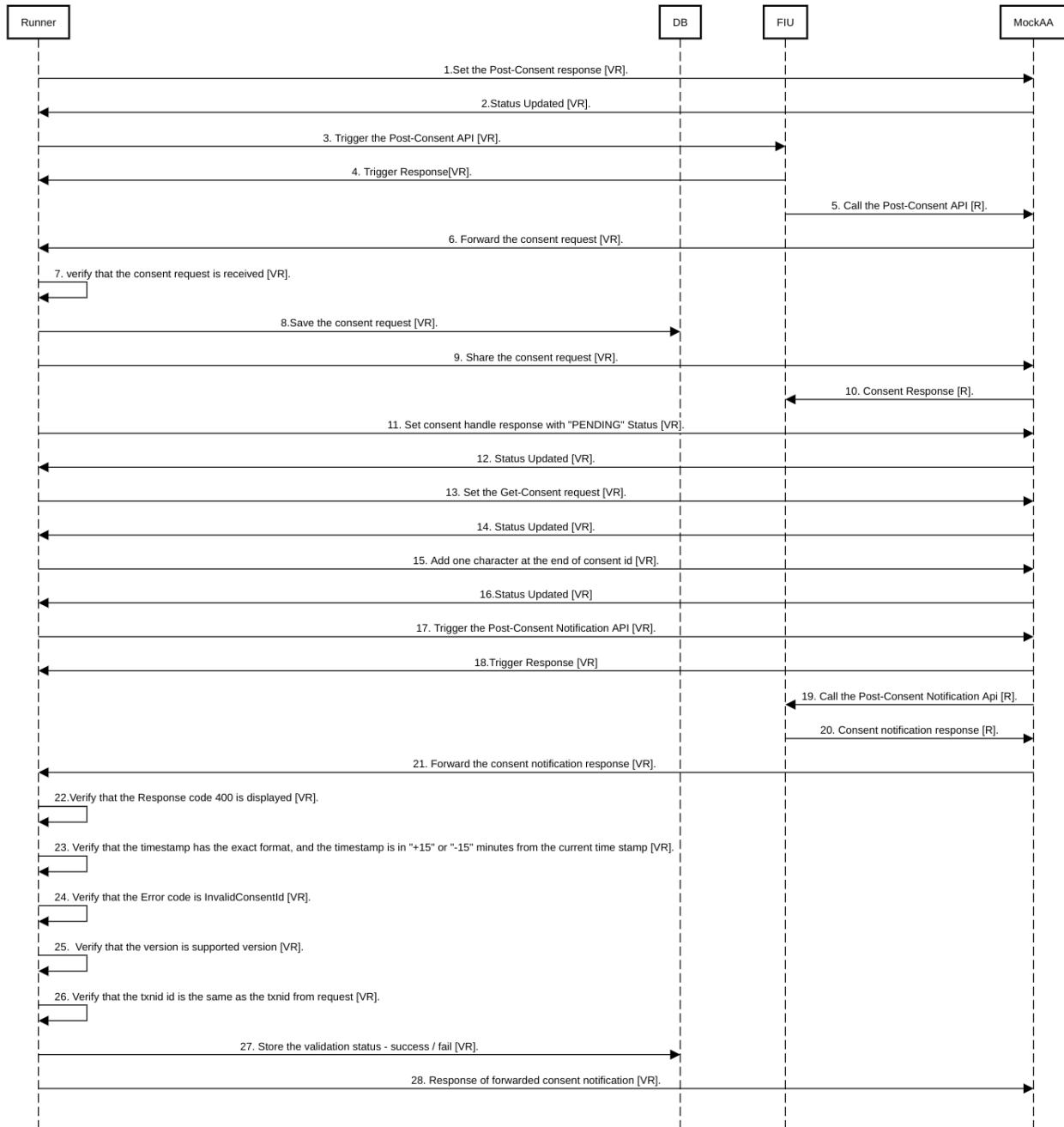
Runner->Runner:25. Verify that the version is supported version [VR].

Runner->Runner:26. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB : 27. Store the validation status – success / fail [VR].

Runner->MockAA:28. Response of forwarded consent notification [VR].

FIU_1007_POST _ConsentNotification.feature Scenario 1007_3



FIU_1008_POST_ConsentNotification.feature

Scenario:1008_1

title FIU_1008_POST_ConsentNotification.feature Scenario 1008_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent request [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set random consent handle [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21. Forward the consent notification response [VR].

Runner ->Runner:22.Verify that the Response code 400 is displayed [VR].

Runner->Runner:23. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:24. Verify that the Error code is InvalidRequest [VR].

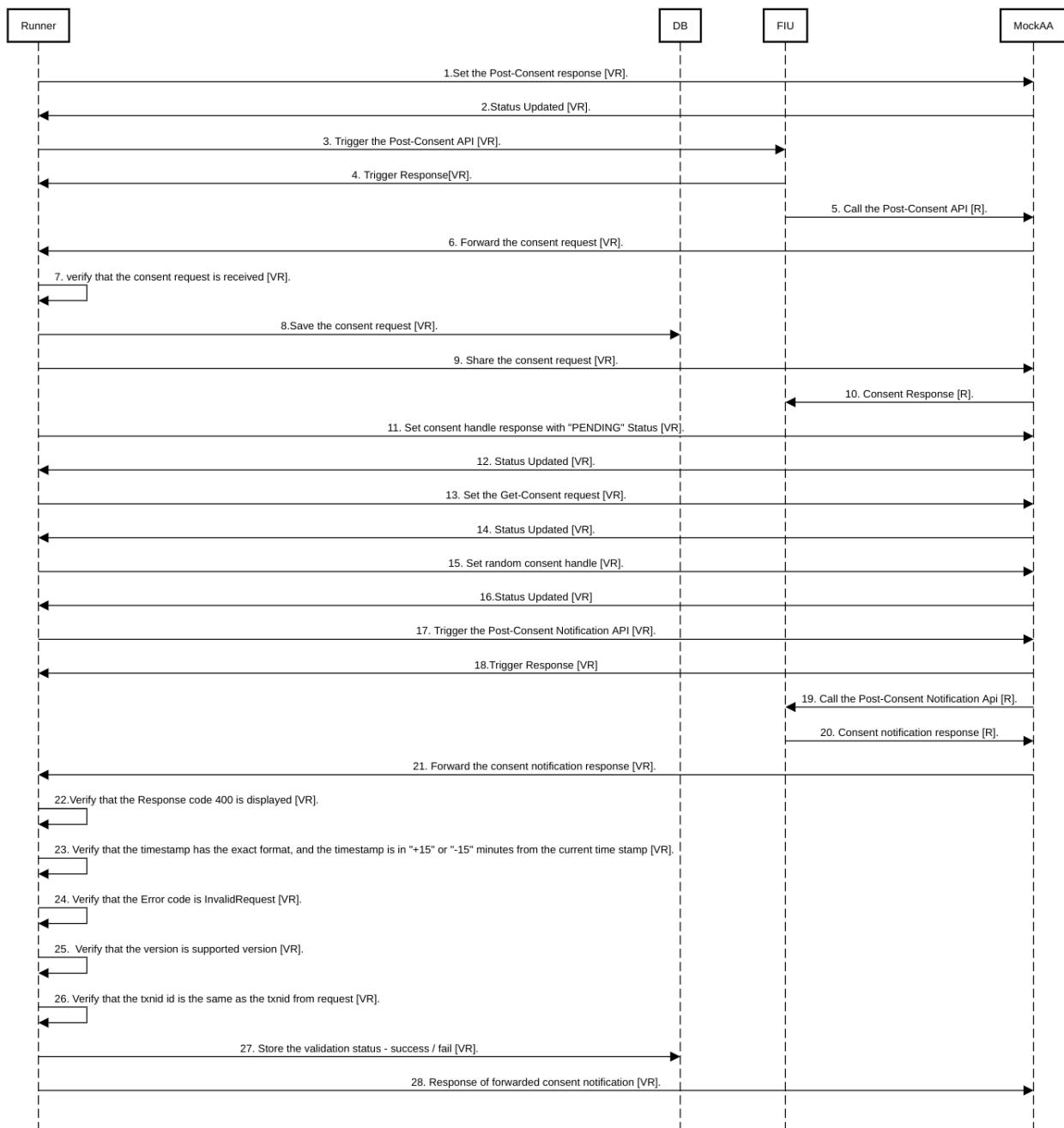
Runner->Runner:25. Verify that the version is supported version [VR].

Runner->Runner:26. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB : 27. Store the validation status – success / fail [VR].

Runner->MockAA:28. Response of forwarded consent notification [VR].

FIU_1008__POST _ConsentNotification.feature Scenario 1008_1



Scenario: 1008_2

title FIU_1008__POST _ConsentNotification.feature Scenario 1008_2

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent request [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Remove the last character from the consent handle [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21. Forward the consent notification response [VR].

Runner ->Runner:22. Verify that the Response code 400 is displayed [VR].

Runner->Runner:23. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

- Runner->Runner:24. Verify that the Error code is InvalidRequest [VR].
 Runner->Runner:25. Verify that the version is supported version [VR].
 Runner->Runner:26. Verify that the txnid id is the same as the txnid from request [VR].
 Runner -> DB : 27. Store the validation status – success / fail [VR].
 Runner->MockAA:28. Response of forwarded consent notification [VR].



Scenario: 1008_3

title FIU_1008__POST _ConsentNotification.feature Scenario 1008_3

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent request [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Add extra characters to the consent handle [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

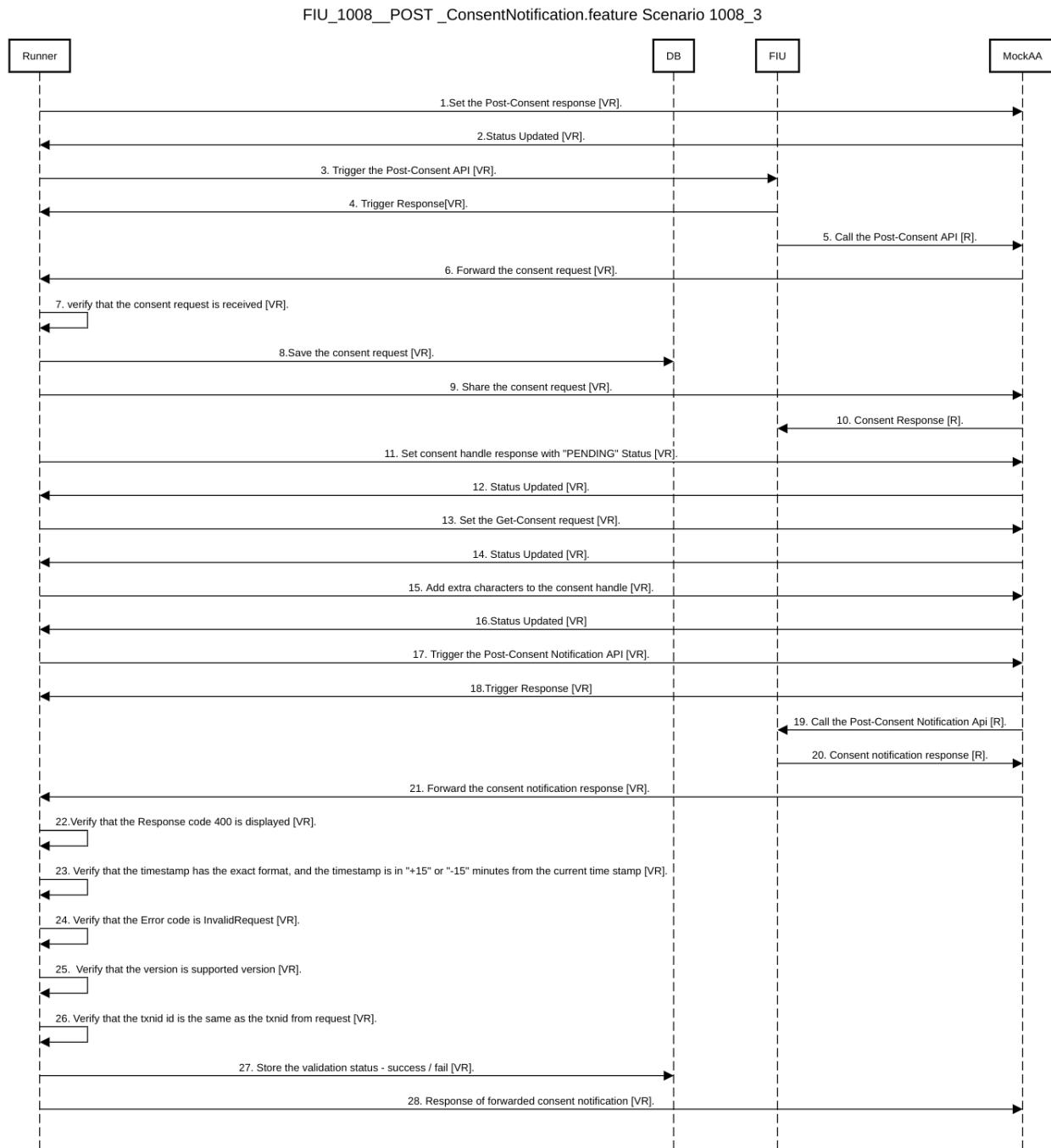
FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21. Forward the consent notification response [VR].

Runner ->Runner:22. Verify that the Response code 400 is displayed [VR].

Runner->Runner:23. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

- Runner->Runner:24. Verify that the Error code is InvalidRequest [VR].
 Runner->Runner:25. Verify that the version is supported version [VR].
 Runner->Runner:26. Verify that the txnid id is the same as the txnid from request [VR].
 Runner -> DB : 27. Store the validation status – success / fail [VR].
 Runner->MockAA:28. Response of forwarded consent notification [VR].



FIU_1009_POST_ConsentNotification.feature

Scenario: 1009_1

title FIU_1009_POST_ConsentNotification.feature Scenario 1009_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent request [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set a null value in the required field [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21. Forward the consent notification response [VR].

Runner ->Runner:22.Verify that the Response code 400 is displayed [VR].

Runner->Runner:23. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:24. Verify that the Error code is InvalidRequest [VR].

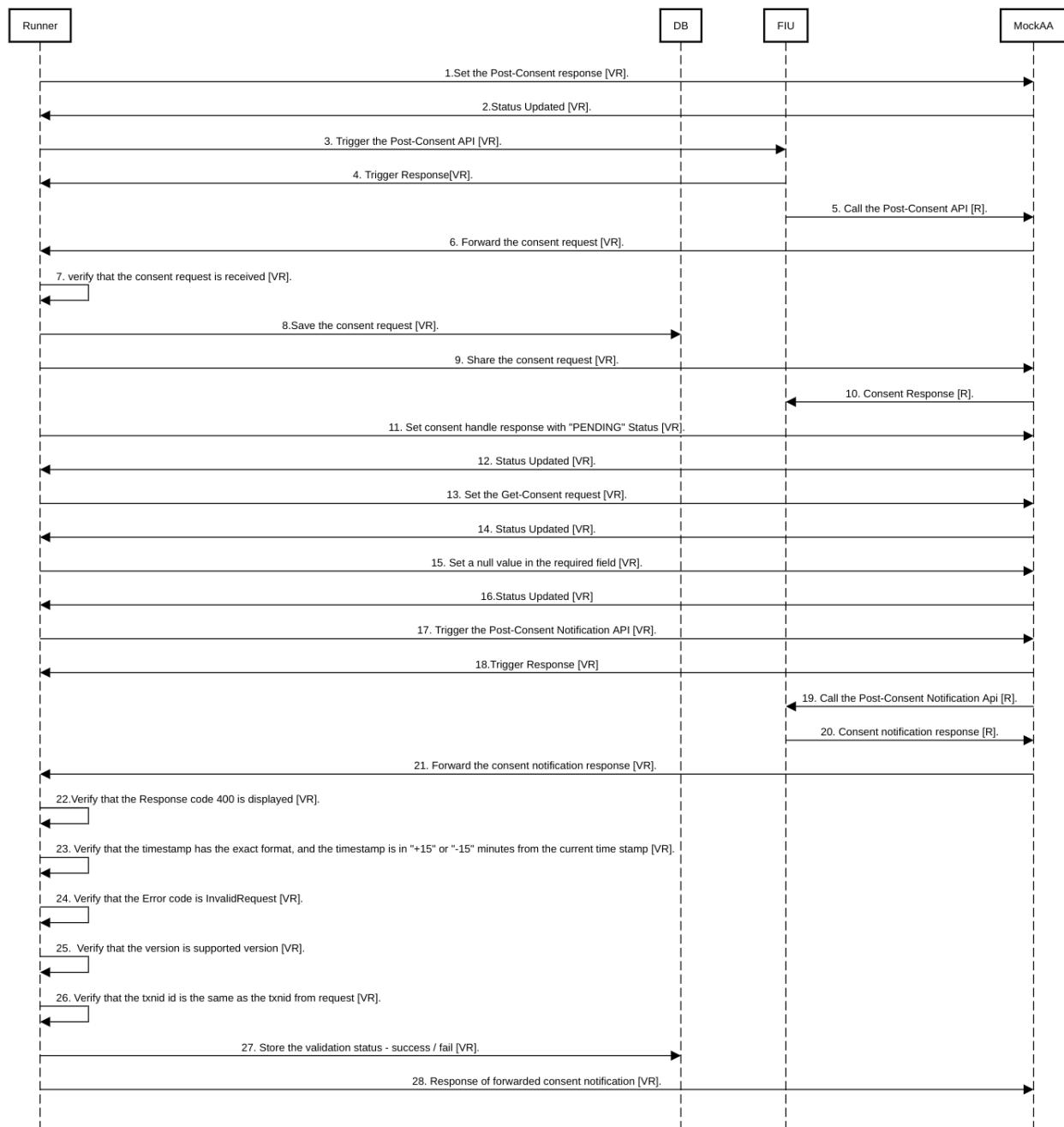
Runner->Runner:25. Verify that the version is supported version [VR].

Runner->Runner:26. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB : 27. Store the validation status – success / fail [VR].

Runner->MockAA:28. Response of forwarded consent notification [VR].

FIU_1009__POST _ConsentNotification.feature Scenario 1009_1



Scenario: 1009_2

title FIU_1009__POST _ConsentNotification.feature Scenario 1009_2

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent request [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set an empty string in the required field [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

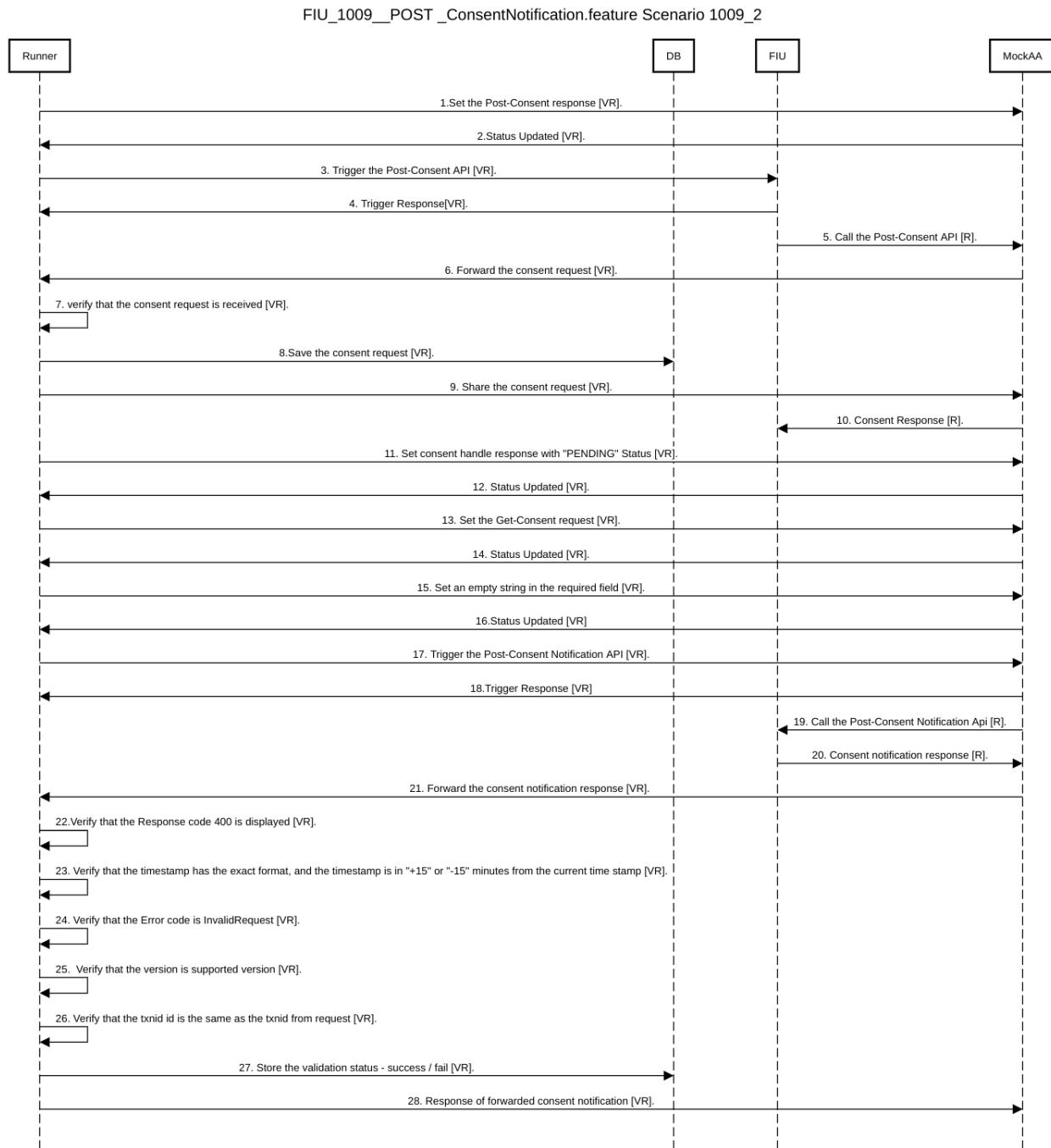
FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21. Forward the consent notification response [VR].

Runner ->Runner:22.Verify that the Response code 400 is displayed [VR].

Runner->Runner:23. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:24. Verify that the Error code is InvalidRequest [VR].
 Runner->Runner:25. Verify that the version is supported version [VR].
 Runner->Runner:26. Verify that the txnid id is the same as the txnid from request [VR].
 Runner -> DB : 27. Store the validation status – success / fail [VR].
 Runner->MockAA:28. Response of forwarded consent notification [VR].



Scenario: 1009_3

title FIU_1009__POST _ConsentNotification.feature Scenario 1009_3

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent request [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set '0'(Zero) in the required field [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21. Forward the consent notification response [VR].

Runner ->Runner:22. Verify that the Response code 400 is displayed [VR].

Runner->Runner:23. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

- Runner->Runner:24. Verify that the Error code is InvalidRequest [VR].
 Runner->Runner:25. Verify that the version is supported version [VR].
 Runner->Runner:26. Verify that the txnid id is the same as the txnid from request [VR].
 Runner -> DB : 27. Store the validation status – success / fail [VR].
 Runner->MockAA:28. Response of forwarded consent notification [VR].



Scenario: 1009_4

title FIU_1009__POST _ConsentNotification.feature Scenario 1009_4

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent request [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set one enum field to non-enum value (Add an extra character at the end of the last enum & remove the last character at the end of the first enum) [VR].

Runner<-MockAA:16.Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18.Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21. Forward the consent notification response [VR].

Runner ->Runner:22.Verify that the Response code 400 is displayed [VR].

Runner->Runner:23. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

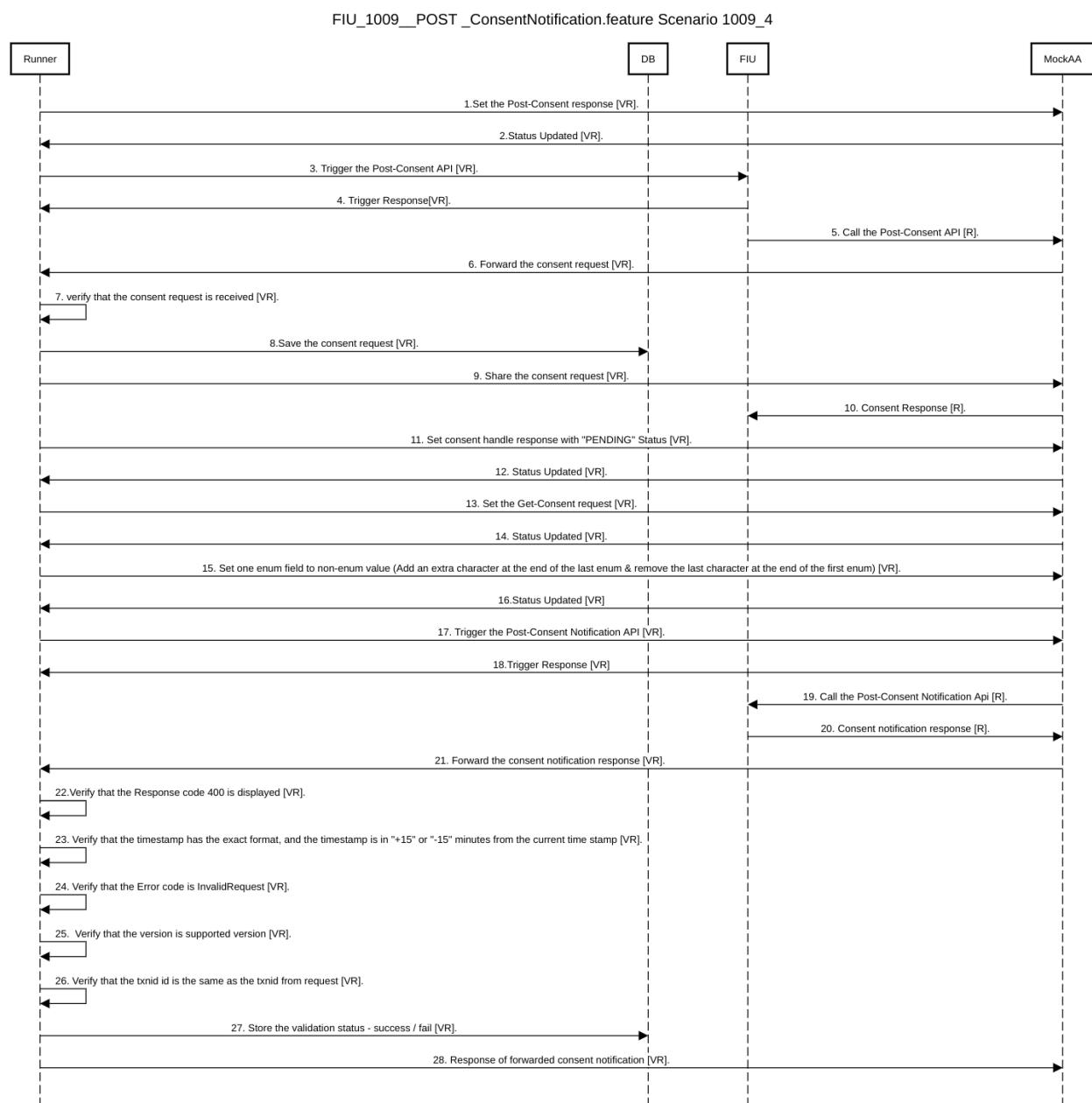
Runner->Runner:24. Verify that the Error code is InvalidRequest [VR].

Runner->Runner:25. Verify that the version is supported version [VR].

Runner->Runner:26. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB : 27. Store the validation status - success / fail [VR].

Runner->MockAA:28. Response of forwarded consent notification [VR].



Scenario: 1009_5

title FIU_1009__POST _ConsentNotification.feature Scenario 1009_5

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent request [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set one array field to empty array [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21. Forward the consent notification response [VR].

Runner ->Runner:22.Verify that the Response code 400 is displayed [VR].

Runner->Runner:23. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

- Runner->Runner:24. Verify that the Error code is InvalidRequest [VR].
 Runner->Runner:25. Verify that the version is supported version [VR].
 Runner->Runner:26. Verify that the txnid id is the same as the txnid from request [VR].
 Runner -> DB : 27. Store the validation status – success / fail [VR].
 Runner->MockAA:28. Response of forwarded consent notification [VR].



Scenario: 1009_6

title FIU_1009__POST _ConsentNotification.feature Scenario 1009_6

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent request [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set one object field to empty object [VR].

Runner<-MockAA:16.Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18.Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

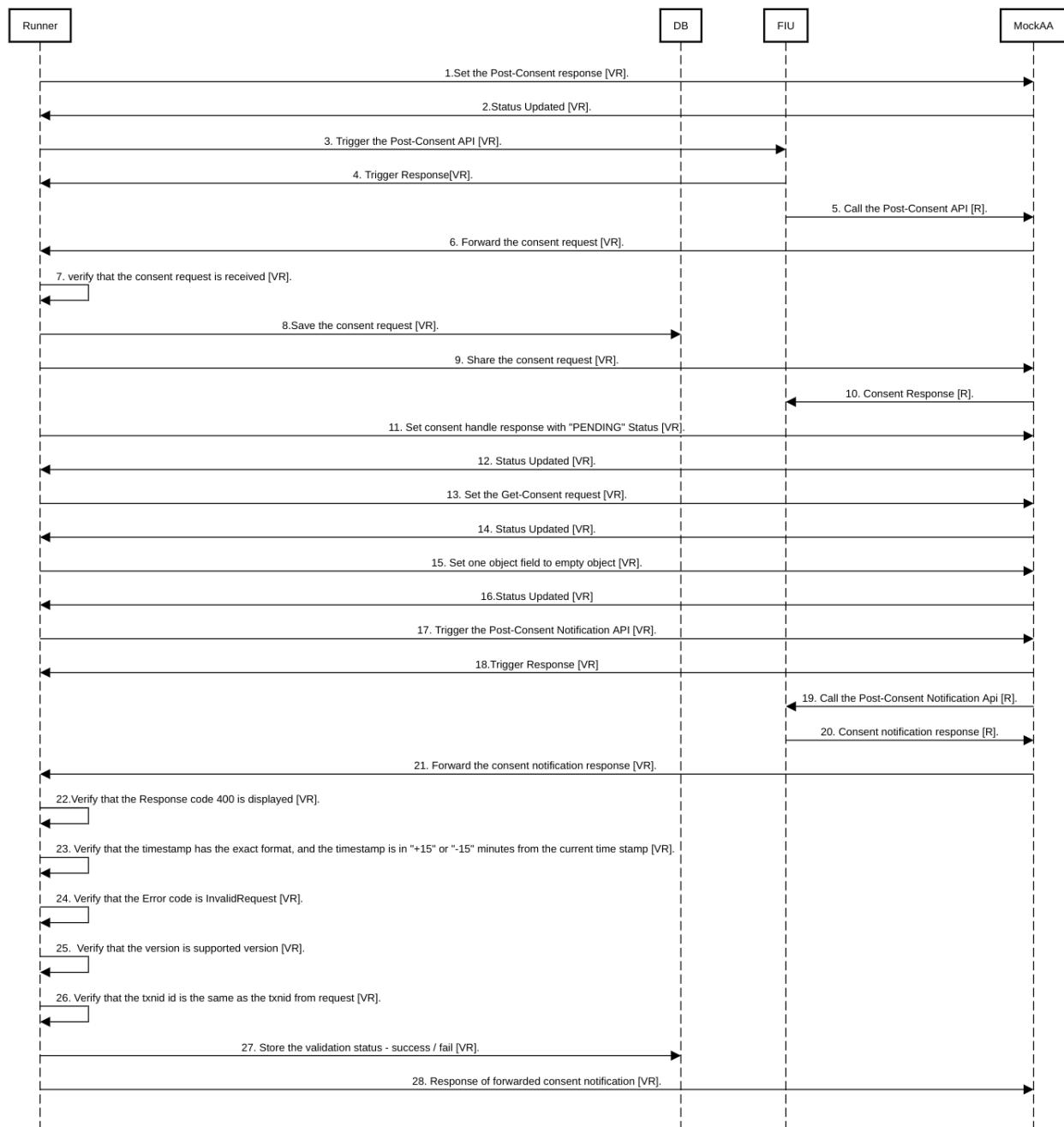
Runner<-MockAA:21. Forward the consent notification response [VR].

Runner ->Runner:22.Verify that the Response code 400 is displayed [VR].

Runner->Runner:23. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:24. Verify that the Error code is InvalidRequest [VR].
Runner->Runner:25. Verify that the version is supported version [VR].
Runner->Runner:26. Verify that the txnid id is the same as the txnid from request [VR].
Runner -> DB : 27. Store the validation status – success / fail [VR].
Runner->MockAA:28. Response of forwarded consent notification [VR].

FIU_1009__POST _ConsentNotification.feature Scenario 1009_6



FIU_1010__POST _ConsentNotification.feature

Scenario: 1010_1

title FIU_1010__POST _ConsentNotification.feature Scenario 1010_1

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent request [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set the Status as "Active" and timestamp with current time + 15 minutes [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21. Forward the consent notification response [VR].

Runner ->Runner:22.Verify that the Response code 400 is displayed [VR].

Runner->Runner:23. Verify that the timestamp has the exact format, and the timestamp is in "+15" minutes from the current time stamp [VR].

Runner->Runner:24. Verify that the Error code is InvalidRequest [VR].

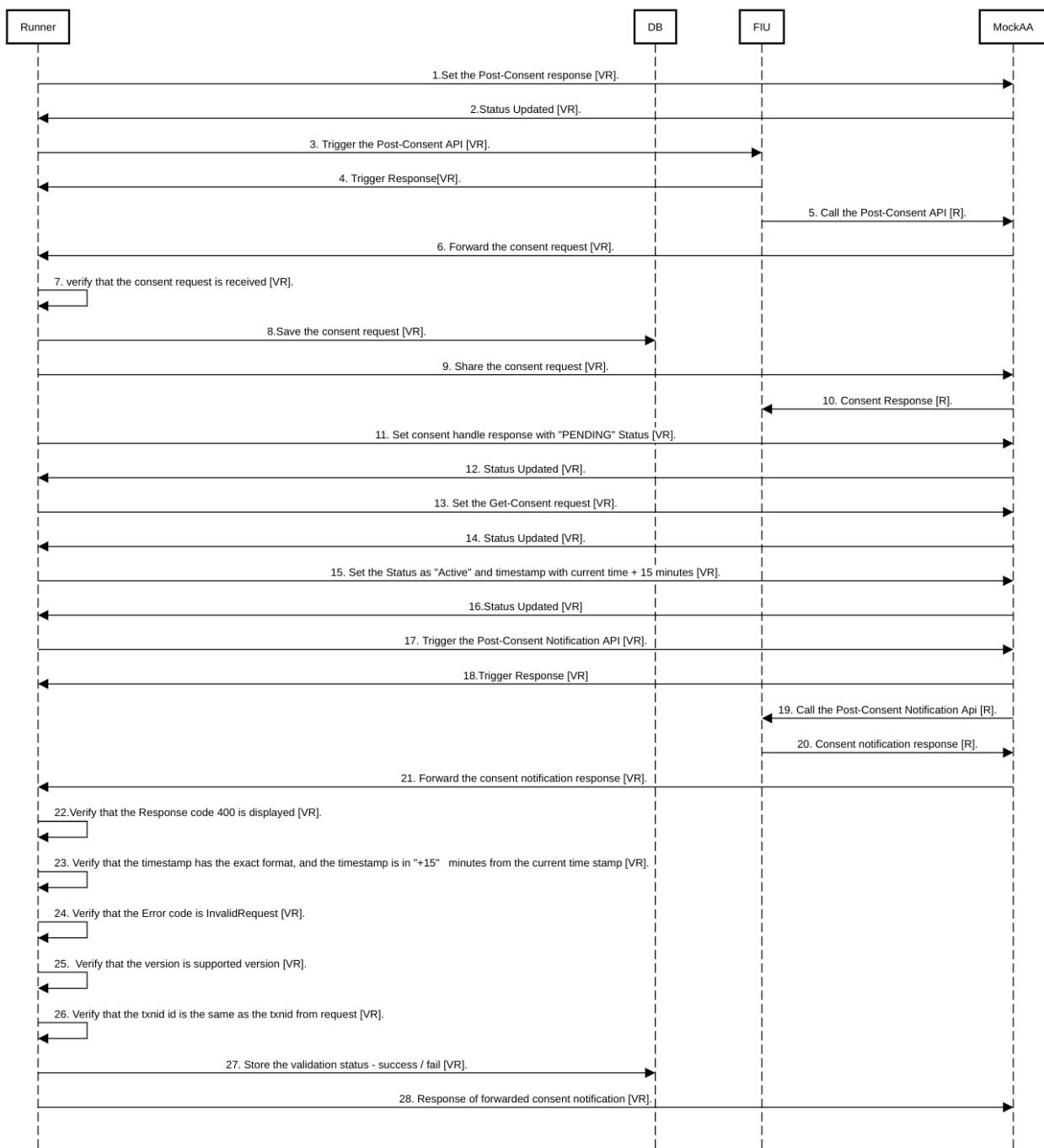
Runner->Runner:25. Verify that the version is supported version [VR].

Runner->Runner:26. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB : 27. Store the validation status - success / fail [VR].

Runner->MockAA:28. Response of forwarded consent notification [VR].

FIU_1010__POST _ConsentNotification.feature Scenario 1010_1



Scenario: 1010_2

title FIU_1010__POST _ConsentNotification.feature Scenario 1010_2

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent request [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set the Status as "Active" and timestamp with current time -15 minutes [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21. Forward the consent notification response [VR].

Runner ->Runner:22.Verify that the Response code 400 is displayed [VR].

Runner->Runner:23. Verify that the timestamp has the exact format, and the timestamp is in "-15" minutes from the current time stamp [VR].

Runner->Runner:24. Verify that the Error code is InvalidRequest [VR].

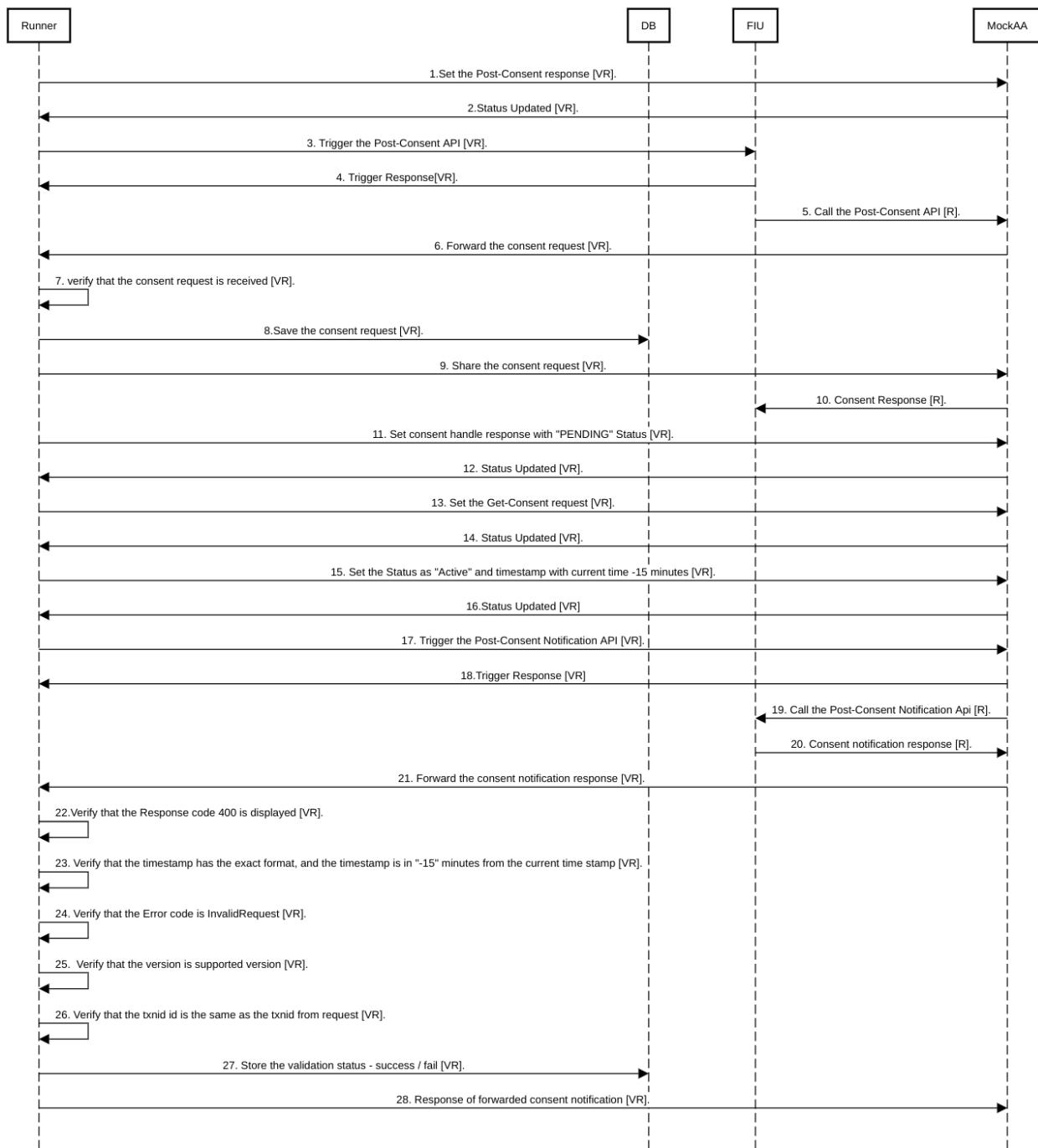
Runner->Runner:25. Verify that the version is supported version [VR].

Runner->Runner:26. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB : 27. Store the validation status - success / fail [VR].

Runner->MockAA:28. Response of forwarded consent notification [VR].

FIU_1010__POST _ConsentNotification.feature Scenario 1010_2



FIU_1011_POST_ConsentNotification.feature

Scenario: 1011_1

title FIU_1011_POST_ConsentNotification.feature Scenario 1011_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response with null values in one required field [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the Status as "Active" [VR].

Runner<-MockAA:12. Status Updated [VR]

Runner->MockAA:13. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:14. Trigger Response [VR]

FIU<-MockAA:15. Call the Post-Consent Notification Api [R].

FIU->MockAA:16. Consent notification response [R].

Runner<-MockAA:17. Forward the consent notification response [VR].

Runner -> Runner:18. Verify that the Response code 400 is displayed [VR].

Runner->Runner:19. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

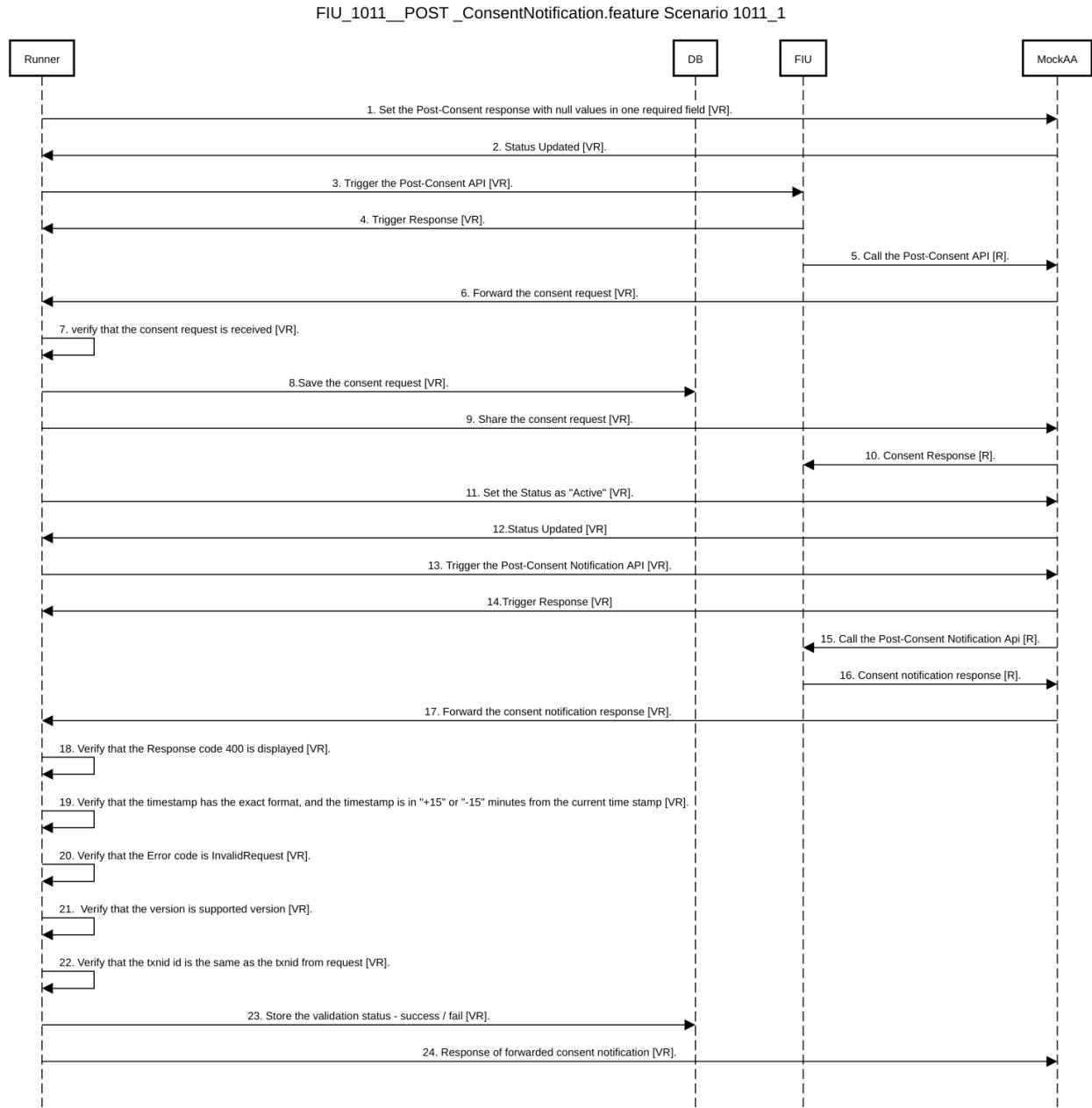
Runner->Runner:20. Verify that the Error code is InvalidRequest [VR].

Runner->Runner:21. Verify that the version is supported version [VR].

Runner->Runner:22. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB:23. Store the validation status - success / fail [VR].

Runner->MockAA:24. Response of forwarded consent notification [VR].



Scenario 1011_2

title FIU_1011__POST _ConsentNotification.feature Scenario 1011_2

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response with an empty string in one required field [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the Status as "Active" [VR].

Runner<-MockAA:12. Status Updated [VR]

Runner->MockAA:13. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:14. Trigger Response [VR]

FIU<-MockAA:15. Call the Post-Consent Notification Api [R].

FIU->MockAA:16. Consent notification response [R].

Runner<-MockAA:17. Forward the consent notification response [VR].

Runner ->Runner:18. Verify that the Response code 400 is displayed [VR].

Runner->Runner:19. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:20. Verify that the Error code is InvalidRequest [VR].

Runner->Runner:21. Verify that the version is supported version [VR].

Runner->Runner:22. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB:23. Store the validation status - success / fail [VR].

Runner->MockAA:24. Response of forwarded consent notification [VR].



Scenario: 1011_3

title FIU_1011_POST_ConsentNotification.feature Scenario 1011_3

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response with '0'(Zero) in one required field [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the Status as "Active" [VR].

Runner<-MockAA:12. Status Updated [VR]

Runner->MockAA:13. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:14. Trigger Response [VR]

FIU<-MockAA:15. Call the Post-Consent Notification Api [R].

FIU->MockAA:16. Consent notification response [R].

Runner<-MockAA:17. Forward the consent notification response [VR].

Runner ->Runner:18. Verify that the Response code 400 is displayed [VR].

Runner->Runner:19. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:20. Verify that the Error code is InvalidRequest [VR].

Runner->Runner:21. Verify that the version is supported version [VR].

Runner->Runner:22. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB:23. Store the validation status - success / fail [VR].

Runner->MockAA:24. Response of forwarded consent notification [VR].

FIU_1011__POST _ConsentNotification.feature Scenario 1011_3



Scenario 1011_4

title FIU_1011__POST _ConsentNotification.feature Scenario 1011_4

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response with an empty object in one required field [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the Status as "Active" [VR].

Runner<-MockAA:12. Status Updated [VR]

Runner->MockAA:13. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:14. Trigger Response [VR]

FIU<-MockAA:15. Call the Post-Consent Notification Api [R].

FIU->MockAA:16. Consent notification response [R].

Runner<-MockAA:17. Forward the consent notification response [VR].

Runner ->Runner:18. Verify that the Response code 400 is displayed [VR].

Runner->Runner:19. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:20. Verify that the Error code is InvalidRequest [VR].

Runner->Runner:21. Verify that the version is supported version [VR].

Runner->Runner:22. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB:23. Store the validation status - success / fail [VR].

Runner->MockAA:24. Response of forwarded consent notification [VR].



FIU_1012_POST_ConsentNotification.feature

Scenario 1012_1:

title FIU_1012_POST_ConsentNotification.feature Scenario 1012_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status and null value in the required field [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->FIU:13. Trigger the Get-Consent handle Api [VR].

Runner<-FIU:14. Trigger Response [VR]

FIU->MockAA:15. Call the Get-Consent handle Api [R].

Runner<-MockAA:16. Forward the consent handle request [VR].

Runner->Runner:17. Verify the request is received and ask the FIU SPOC if the consentHandle response is discarded [VR].

Runner->DB:18. Save the consent handle request [VR].

Runner->MockAA:19. Share the consent handle request [VR].

FIU<-MockAA:20. Consent handle Response [R].

Runner->MockAA:21. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:22. Trigger Response [VR]

FIU<-MockAA:23. Call the Post-Consent Notification Api [R].

FIU->MockAA:24. Consent notification response [R].

Runner<-MockAA:25. Forward the consent notification response [VR].

Runner ->Runner:26. Verify that the Response code 400 is displayed [VR].

Runner -> DB:27. Store the validation status - success / fail [VR].

Runner->MockAA:28. Response of forwarded consent notification [VR].

FIU_1012__POST _ConsentNotification.feature Scenario 1012_1



Scenario 1012_2:

title FIU_1012__POST _ConsentNotification.feature Scenario 1012_2

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status and an empty string in the required field [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->FIU:13. Trigger the Get-Consent handle Api [VR].

Runner<-FIU:14. Trigger Response [VR]

FIU->MockAA:15. Call the Get-Consent handle Api [R].

Runner<-MockAA:16. Forward the consent handle request [VR].

Runner->Runner:17. Verify the request is received and ask the FIU SPOC if the consentHandle response is discarded [VR].

Runner->DB:18. Save the consent handle request [VR].

Runner->MockAA:19. Share the consent handle request [VR].

FIU<-MockAA:20. Consent handle Response [R].

Runner->MockAA:21. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:22. Trigger Response [VR]

FIU<-MockAA:23. Call the Post-Consent Notification Api [R].

FIU->MockAA:24. Consent notification response [R].

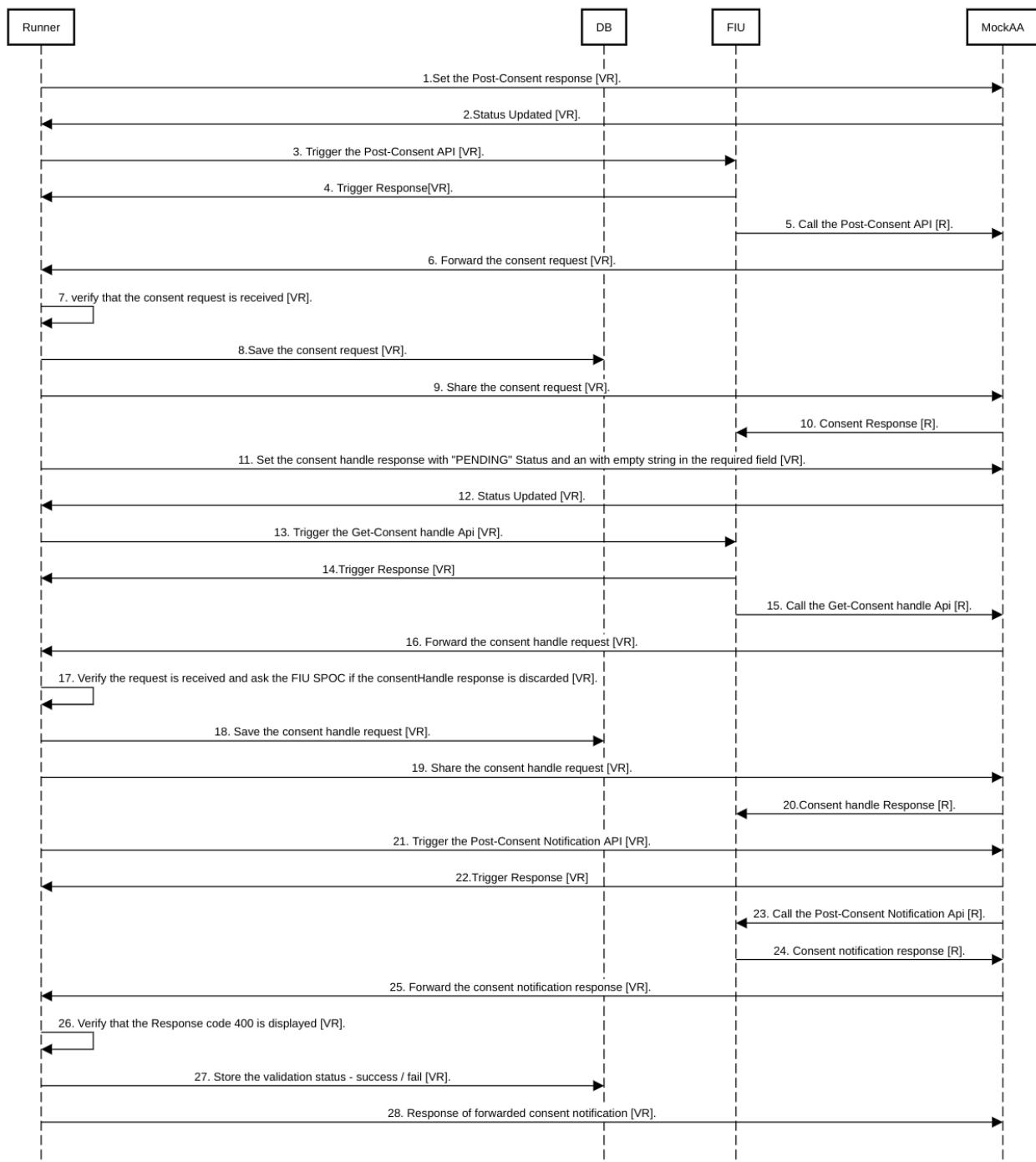
Runner<-MockAA:25. Forward the consent notification response [VR].

Runner ->Runner:26. Verify that the Response code 400 is displayed [VR].

Runner -> DB:27. Store the validation status - success / fail [VR].

Runner->MockAA:28. Response of forwarded consent notification [VR].

FIU_1012__POST _ConsentNotification.feature Scenario 1012_2



Scenario 1012_3

title FIU_1012__POST _ConsentNotification.feature Scenario 1012_3

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call
R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].
Runner<-MockAA:2. Status Updated [VR].
Runner->FIU:3. Trigger the Post-Consent API [VR].
Runner<-FIU:4. Trigger Response[VR].
FIU->MockAA:5. Call the Post-Consent API [R].
Runner<-MockAA:6. Forward the consent request [VR].
Runner->Runner:7. verify that the consent request is received [VR].
Runner -> DB:8. Save the consent request [VR].
Runner->MockAA:9. Share the consent request [VR].
FIU<-MockAA:10. Consent Response [R].
Runner->MockAA:11. Set the consent handle response with "PENDING" Status and 'o' (Zero) in the required field [VR].
Runner<-MockAA:12. Status Updated [VR].
Runner->FIU:13. Trigger the Get-Consent handle Api [VR].
Runner<-FIU:14. Trigger Response [VR]
FIU->MockAA:15. Call the Get-Consent handle Api [R].
Runner<-MockAA:16. Forward the consent handle request [VR].
Runner->Runner:17. Verify the request is received and ask the FIU SPOC if the consentHandle response is discarded [VR].
Runner->DB:18. Save the consent handle request [VR].
Runner->MockAA:19. Share the consent handle request [VR].
FIU<-MockAA:20. Consent handle Response [R].
Runner->MockAA:21. Trigger the Post-Consent Notification API [VR].
Runner<-MockAA:22. Trigger Response [VR]

FIU<-MockAA:23. Call the Post-Consent Notification Api [R].

FIU->MockAA:24. Consent notification response [R].

Runner<-MockAA:25. Forward the consent notification response [VR].

Runner ->Runner:26. Verify that the Response code 400 is displayed [VR].

Runner -> DB:27. Store the validation status - success / fail [VR].

Runner->MockAA:28. Response of forwarded consent notification [VR].

FIU_1012__POST _ConsentNotification.feature Scenario 1012_3



Scenario 1012_4

title FIU_1012__POST _ConsentNotification.feature Scenario 1012_4

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status and empty object in the required field [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->FIU:13. Trigger the Get-Consent handle Api [VR].

Runner<-FIU:14. Trigger Response [VR]

FIU->MockAA:15. Call the Get-Consent handle Api [R].

Runner<-MockAA:16. Forward the consent handle request [VR].

Runner->Runner:17. Verify the request is received and ask the FIU SPOC if the consentHandle response is discarded [VR].

Runner->DB:18. Save the consent handle request [VR].

Runner->MockAA:19. Share the consent handle request [VR].

FIU<-MockAA:20. Consent handle Response [R].

Runner->MockAA:21. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:22. Trigger Response [VR]

FIU<-MockAA:23. Call the Post-Consent Notification Api [R].

FIU->MockAA:24. Consent notification response [R].

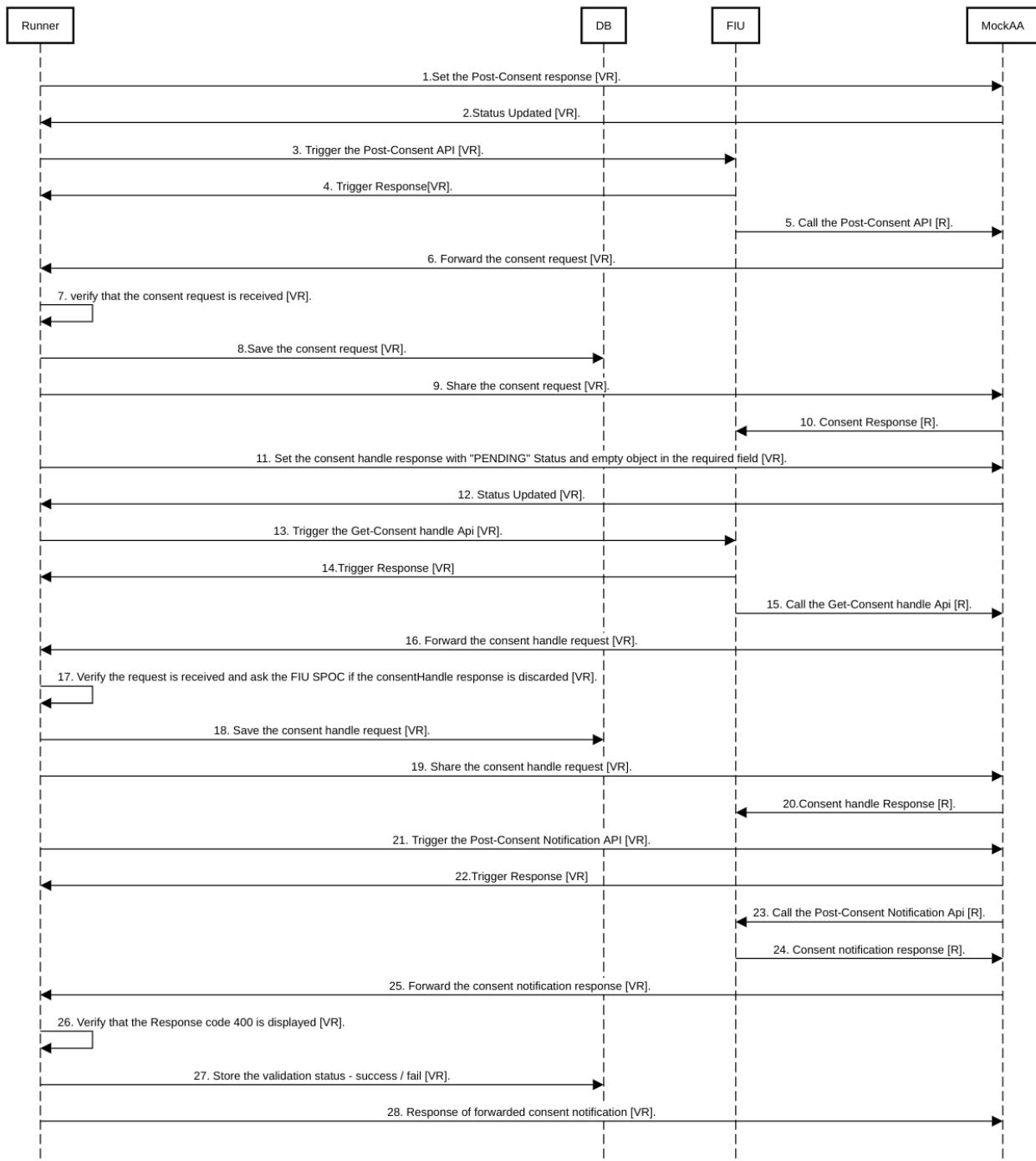
Runner<-MockAA:25. Forward the consent notification response [VR].

Runner ->Runner:26. Verify that the Response code 400 is displayed [VR].

Runner -> DB:27. Store the validation status - success / fail [VR].

Runner->MockAA:28. Response of forwarded consent notification [VR].

FIU_1012__POST _ConsentNotification.feature Scenario 1012_4



FIU_1013__POST _ConsentNotification.feature

Scenario 1013_1

title FIU_1013__POST _ConsentNotification.feature Scenario 1013_1

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8.Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent response [VR].

Runner<-MockAA:14.Status Updated [VR].

Runner->MockAA:15. Set incorrect version number with 'ACTIVE' status [VR].

Runner<-MockAA:16.Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18.Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21. Forward the consent notification response [VR].

Runner ->Runner:22. Verify that the Response code 404 is displayed [VR].

Runner->Runner:22. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:23. Verify that the Error code is NoSuchVersion [VR].

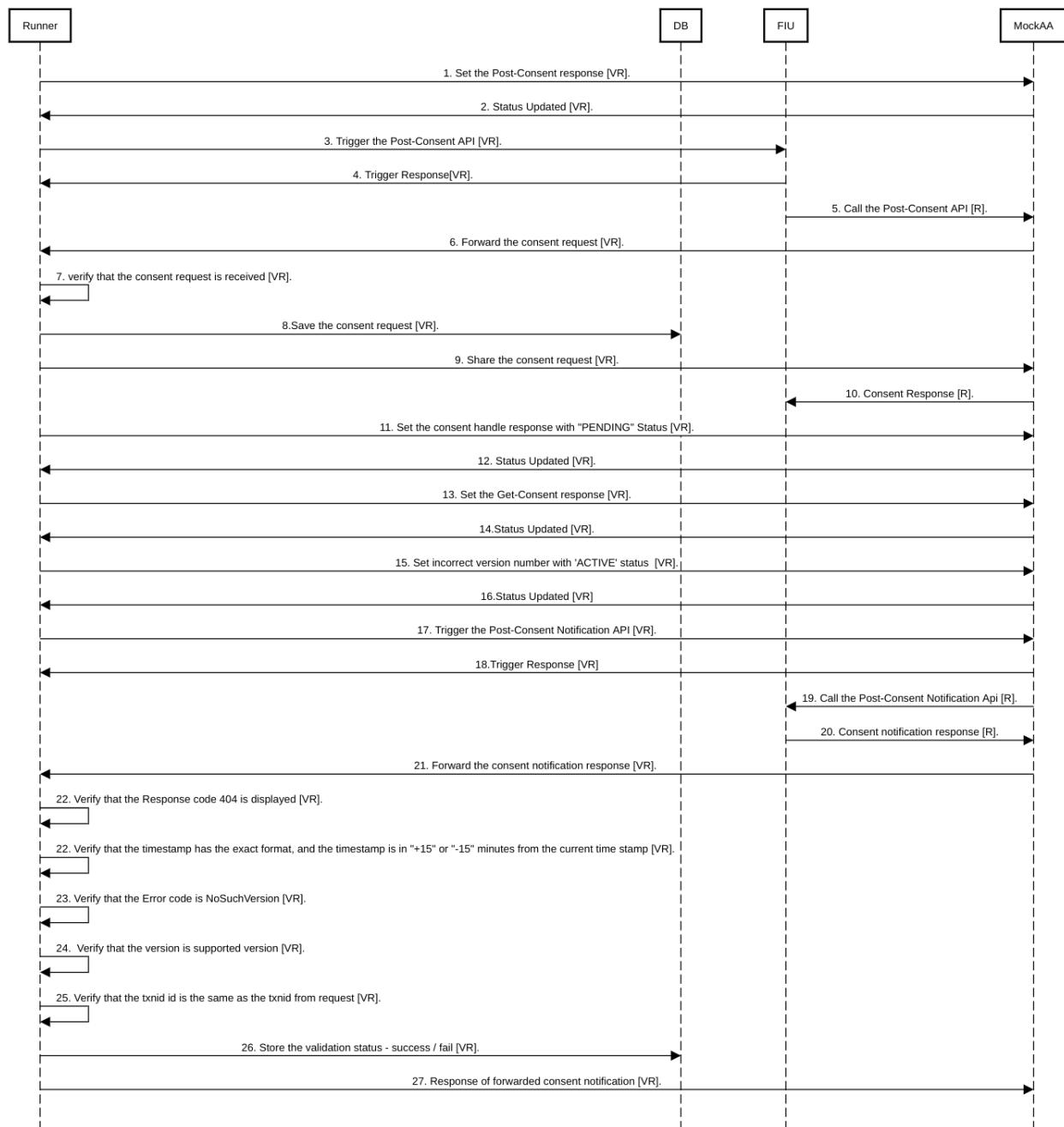
Runner->Runner:24. Verify that the version is supported version [VR].

Runner->Runner:25. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB:26. Store the validation status - success / fail [VR].

Runner->MockAA:27. Response of forwarded consent notification [VR].

FIU_1013__POST _ConsentNotification.feature Scenario 1013_1



Scenario 1013_2

title FIU_1013__POST _ConsentNotification.feature Scenario 1013_2

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8.Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent response [VR].

Runner<-MockAA:14.Status Updated [VR].

Runner->MockAA:15. Set an extra digit or character at the end of the correct version number with 'ACTIVE' status [VR].

Runner<-MockAA:16.Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18.Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21. Forward the consent notification response [VR].

Runner ->Runner:22. Verify that the Response code 404 is displayed [VR].

Runner->Runner:22. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:23. Verify that the Error code is NoSuchVersion [VR].

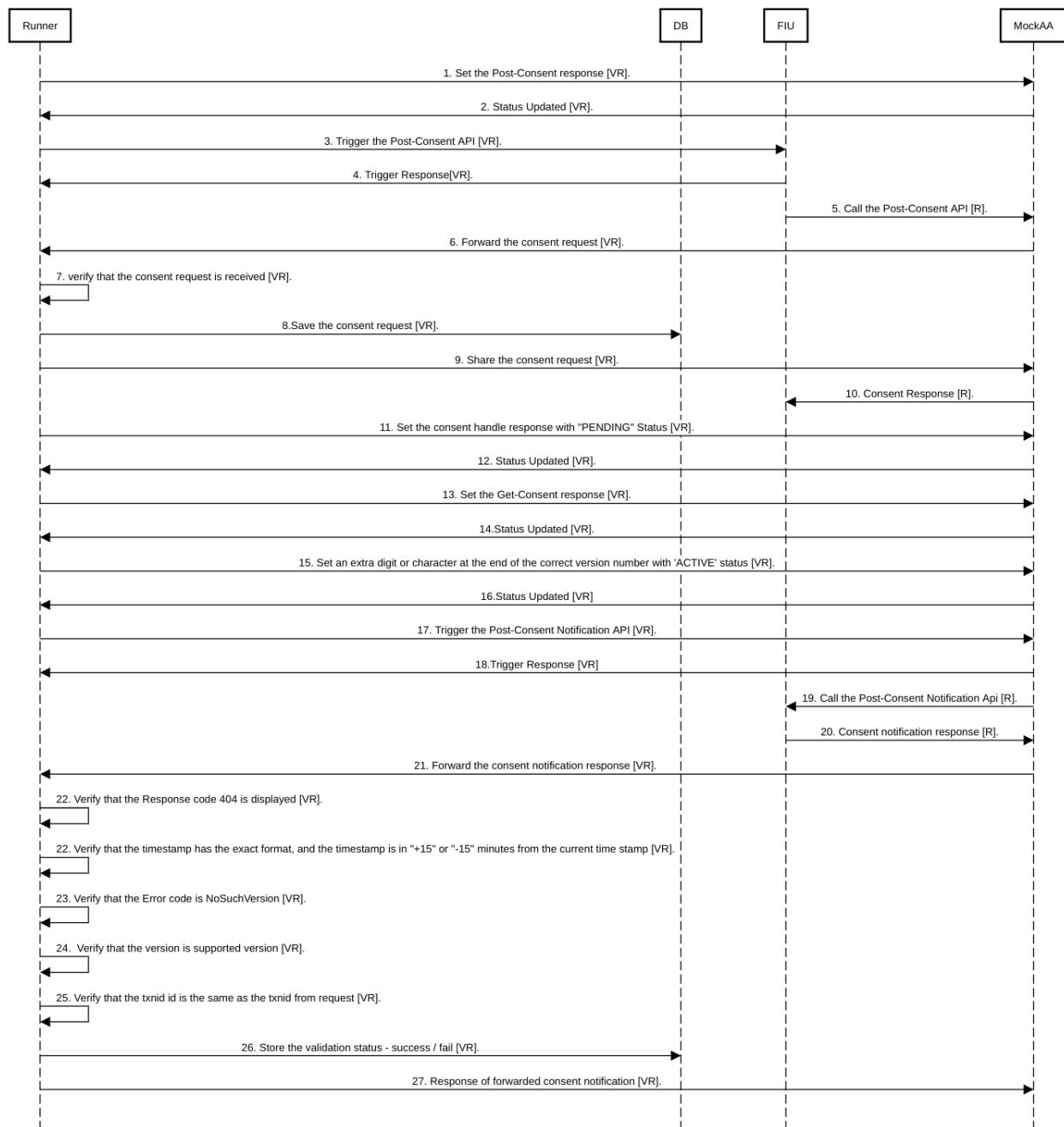
Runner->Runner:24. Verify that the version is supported version [VR].

Runner->Runner:25. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB:26. Store the validation status - success / fail [VR].

Runner->MockAA:27. Response of forwarded consent notification [VR].

FIU_1013__POST _ConsentNotification.feature Scenario 1013_2



Scenario 1013_3

title FIU_1013__POST _ConsentNotification.feature Scenario 1013_3

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8.Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent response [VR].

Runner<-MockAA:14.Status Updated [VR].

Runner->MockAA:15. Remove the extra digit or character at the end of the correct version number with 'ACTIVE' status [VR].

Runner<-MockAA:16.Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18.Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21. Forward the consent notification response [VR].

Runner ->Runner:22. Verify that the Response code 404 is displayed [VR].

Runner->Runner:22. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:23. Verify that the Error code is NoSuchVersion [VR].

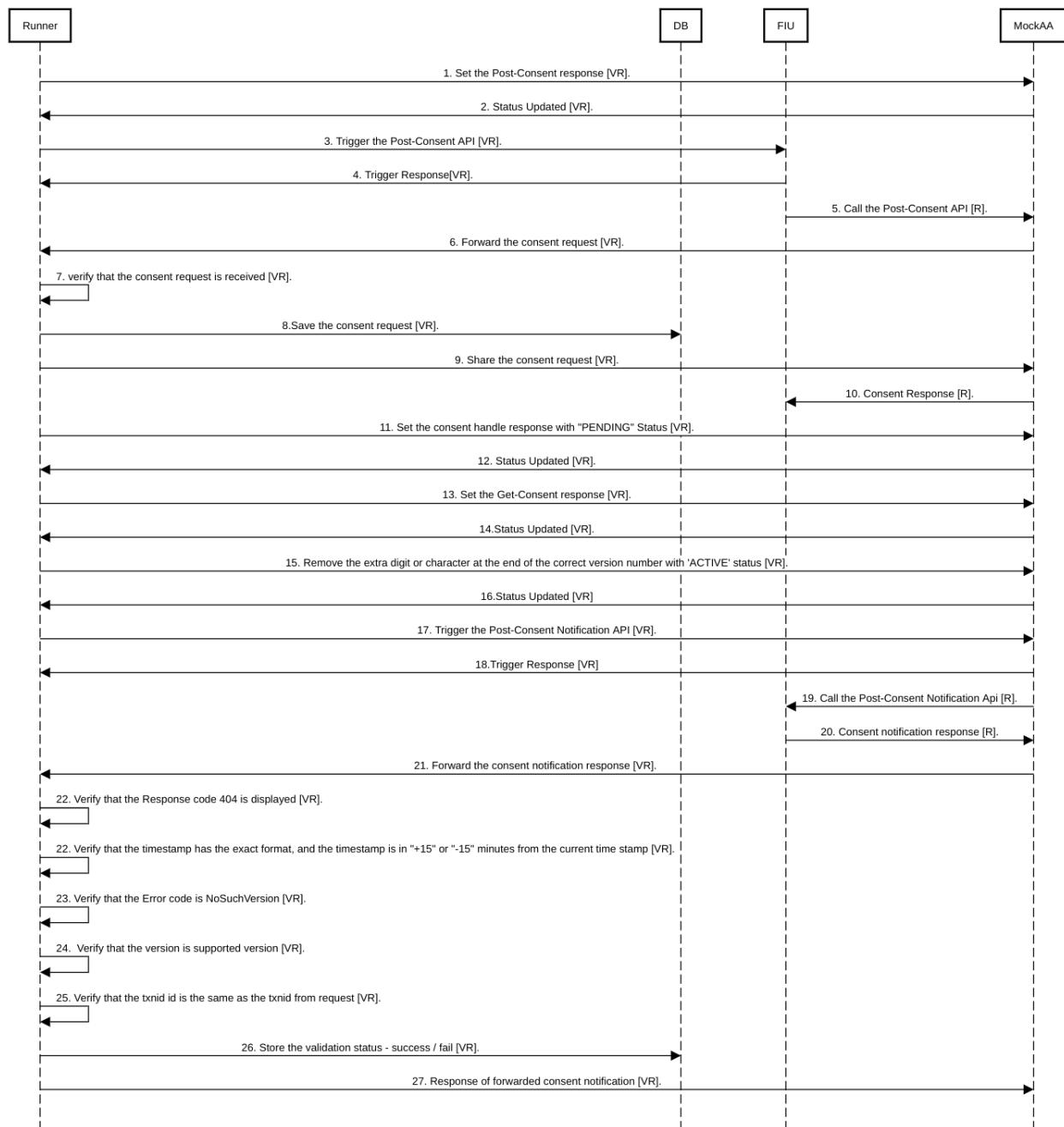
Runner->Runner:24. Verify that the version is supported version [VR].

Runner->Runner:25. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB:26. Store the validation status - success / fail [VR].

Runner->MockAA:27. Response of forwarded consent notification [VR].

FIU_1013__POST _ConsentNotification.feature Scenario 1013_3



FIU_1014_POST_ConsentNotification.feature

Scenario 1014_1

title FIU_1014_POST_ConsentNotification.feature Scenario 1014_1

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8.Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent response [VR].

Runner<-MockAA:14.Status Updated [VR].

Runner->MockAA:15. Set two or three incorrect timestamp formats in each timestamp field that are not followed in the specification with 'ACTIVE' status [VR].

Runner<-MockAA:16.Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18.Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21. Forward the consent notification response [VR].

Runner ->Runner:22. Verify that the Response code 404 is displayed [VR].

Runner->Runner:22. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:23. Verify that the Error code is NoSuchVersion [VR].

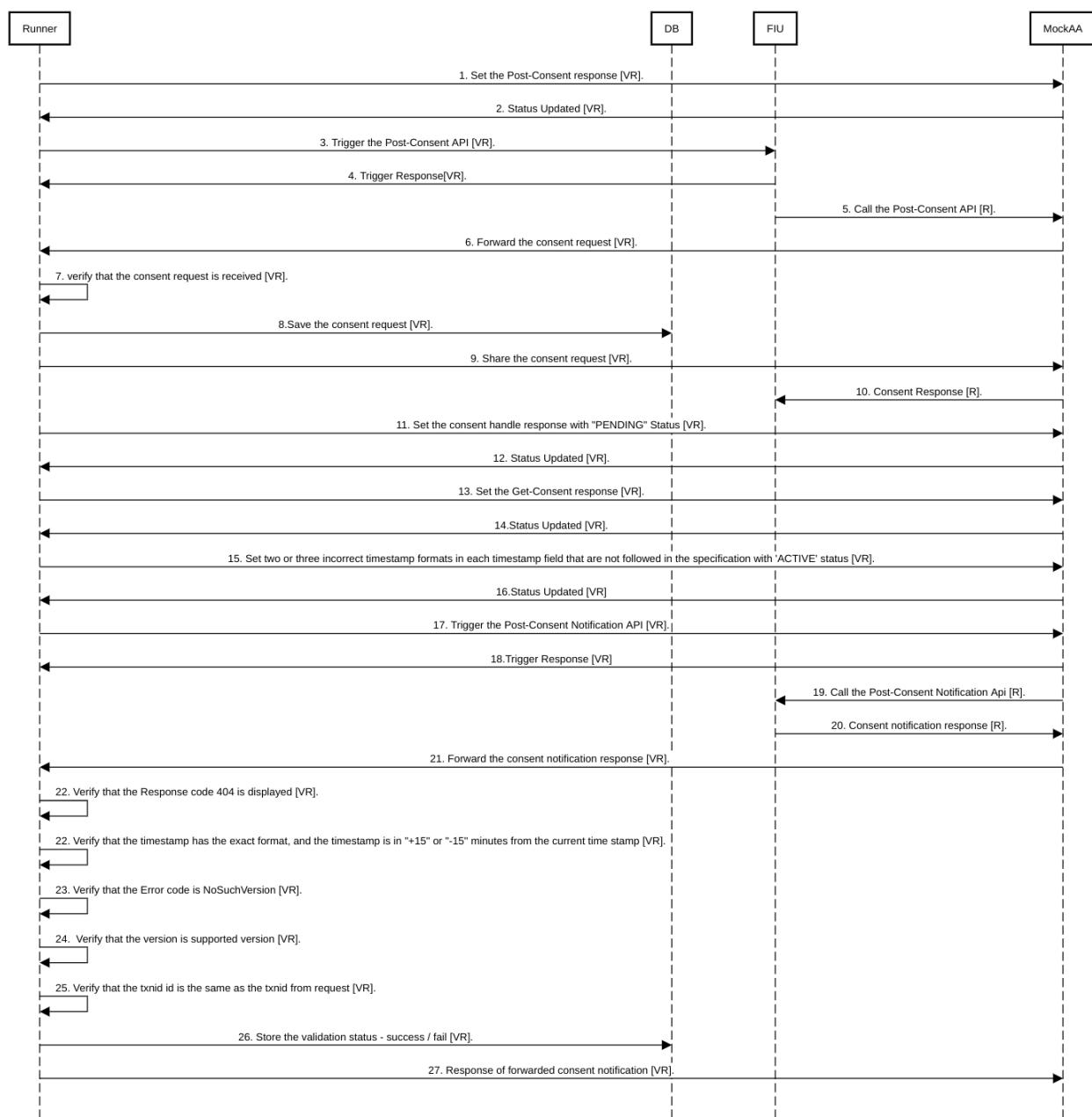
Runner->Runner:24. Verify that the version is supported version [VR].

Runner->Runner:25. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB:26. Store the validation status – success / fail [VR].

Runner->MockAA:27. Response of forwarded consent notification [VR].

FIU_1014__POST _ConsentNotification.feature Scenario 1014_1



Scenario 1014_2

title FIU_1014_POST_ConsentNotification.feature Scenario 1014_2

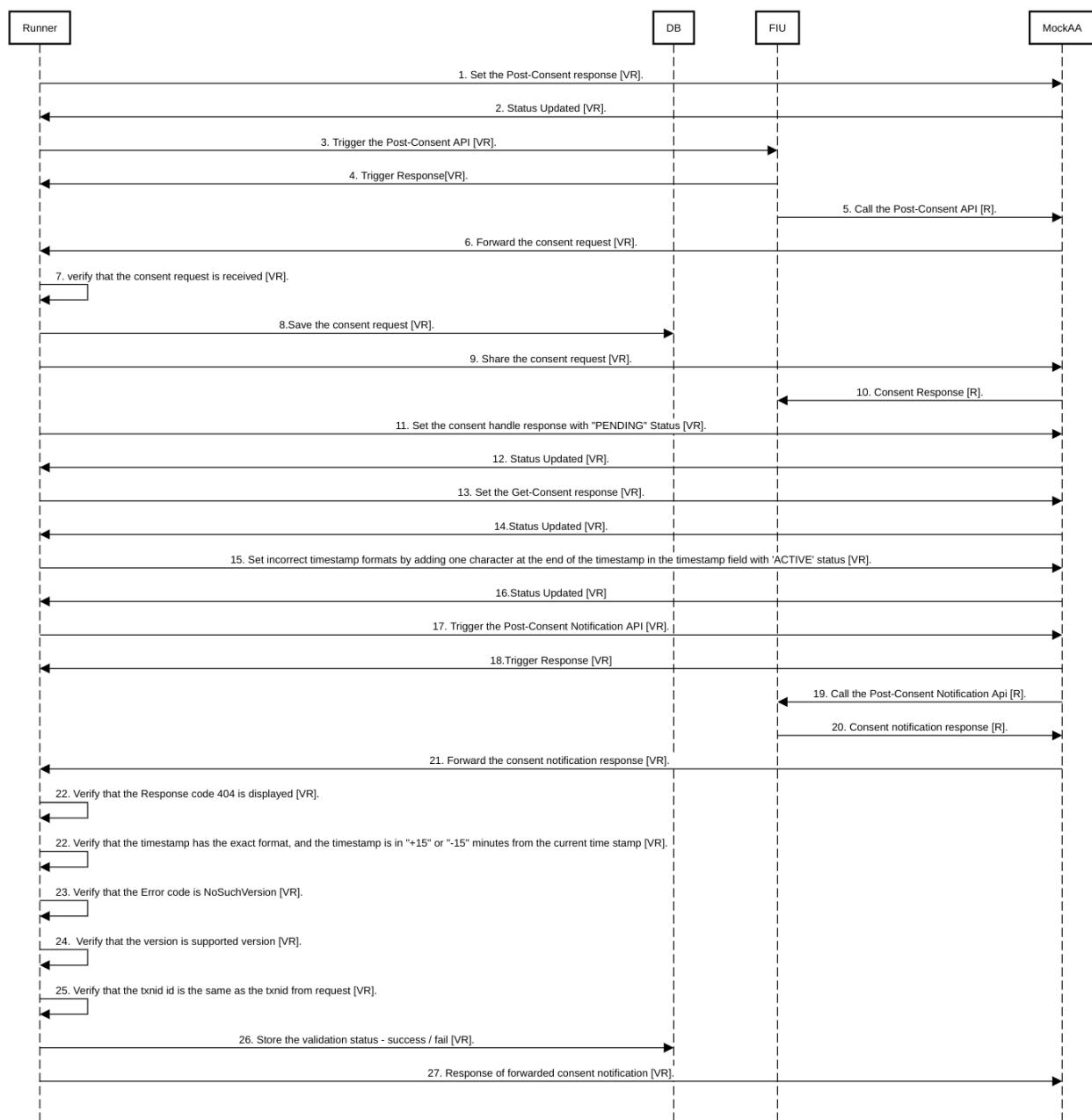
participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call
R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].
Runner<-MockAA:2. Status Updated [VR].
Runner->FIU:3. Trigger the Post-Consent API [VR].
Runner<-FIU:4. Trigger Response[VR].
FIU->MockAA:5. Call the Post-Consent API [R].
Runner<-MockAA:6. Forward the consent request [VR].
Runner->Runner:7. verify that the consent request is received [VR].
Runner -> DB:8. Save the consent request [VR].
Runner->MockAA:9. Share the consent request [VR].
FIU<-MockAA:10. Consent Response [R].
Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].
Runner<-MockAA:12. Status Updated [VR].
Runner->MockAA:13. Set the Get-Consent response [VR].
Runner<-MockAA:14. Status Updated [VR].
Runner->MockAA:15. Set incorrect timestamp formats by adding one character at the end of the timestamp in the timestamp field with 'ACTIVE' status [VR].
Runner<-MockAA:16. Status Updated [VR]
Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].
Runner<-MockAA:18. Trigger Response [VR]
FIU<-MockAA:19. Call the Post-Consent Notification Api [R].
FIU->MockAA:20. Consent notification response [R].
Runner<-MockAA:21. Forward the consent notification response [VR].

Runner ->Runner:22. Verify that the Response code 404 is displayed [VR].
Runner->Runner:22. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].
Runner->Runner:23. Verify that the Error code is NoSuchVersion [VR].
Runner->Runner:24. Verify that the version is supported version [VR].
Runner->Runner:25. Verify that the txnid id is the same as the txnid from request [VR].
Runner -> DB:26. Store the validation status - success / fail [VR].
Runner->MockAA:27. Response of forwarded consent notification [VR].

FIU_1014__POST _ConsentNotification.feature Scenario 1014_2



FIU_1015_POST_ConsentNotification.feature

Scenario 1015_1

title FIU_1015_POST_ConsentNotification.feature Scenario 1015_1

participant Runner
participant DB
participant FIU
participant MockAA
participant Alternate MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent Request [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->Alternate MockAA:15. Set the Get-Consent Request [VR].

Runner<-Alternate MockAA:16. Status Updated [VR].

Runner->MockAA:17. Set the Consent Handle of alternate AA [VR].

Runner<-MockAA:18. Status Updated [VR]

Runner->MockAA:19. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:20. Trigger Response [VR]

FIU<-MockAA:21. Call the Post-Consent Notification Api [R].

FIU->MockAA:22. Consent notification response [R].

Runner<-MockAA:23. Forward the consent notification response [VR].

Runner ->Runner:24. Verify that the Response code 400 is displayed [VR].

Runner->Runner:25. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:26. Verify that the Error code is InvalidRequest [VR].

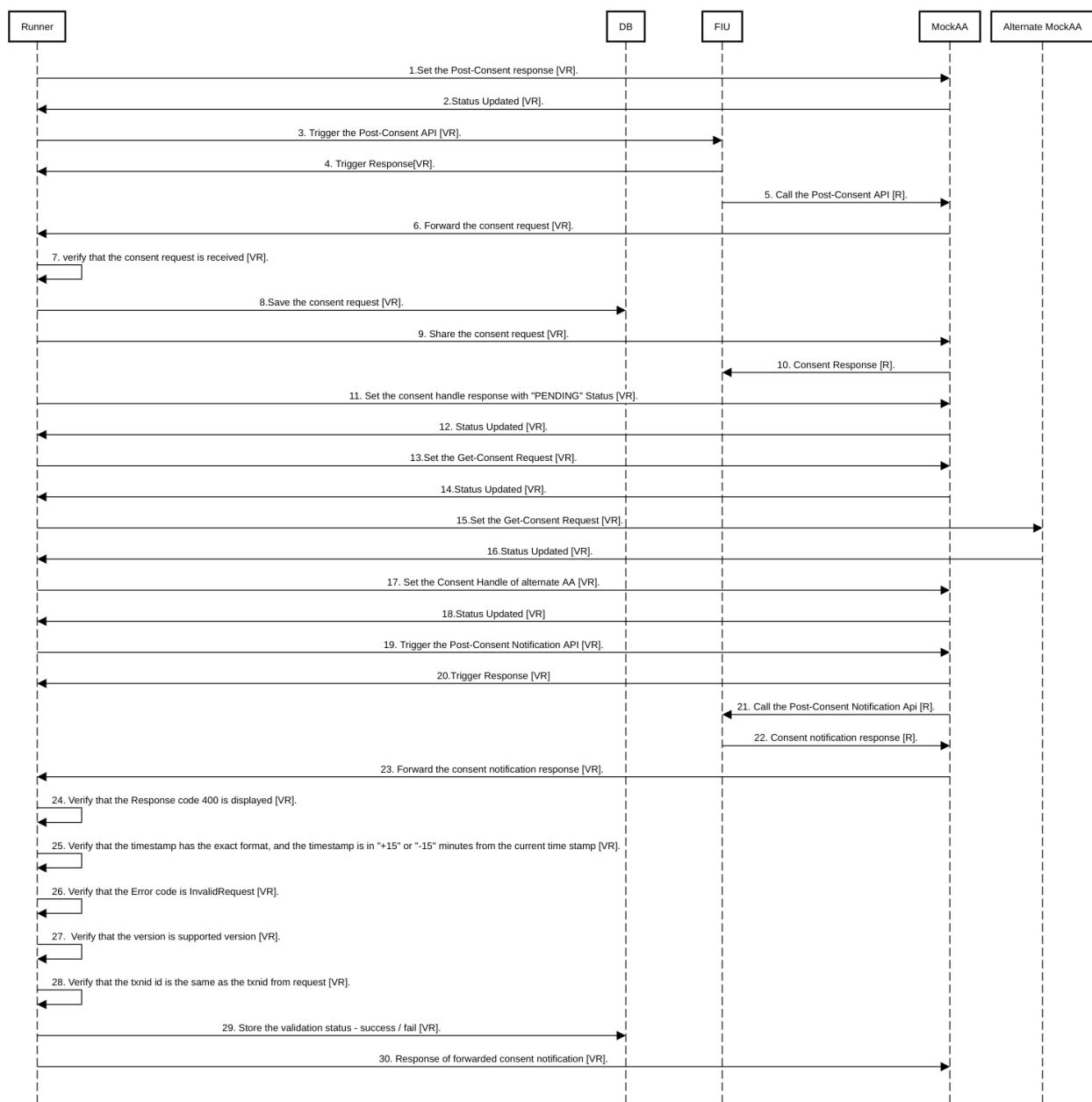
Runner->Runner:27. Verify that the version is supported version [VR].

Runner->Runner:28. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB:29. Store the validation status - success / fail [VR].

Runner->MockAA:30. Response of forwarded consent notification [VR].

FIU_1015__POST _ConsentNotification.feature Scenario 1015_1



FIU_1016_POST_ConsentNotification.feature

Scenario 1016_1

title FIU_1016_POST_ConsentNotification.feature Scenario 1016_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8.Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13.Set the Get-Consent response [VR]

Runner<-MockAA:14.Status Updated [VR]

Runner->MockAA:15. Set the status as "Active" [VR].

Runner<-MockAA:16.Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18.Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21.Forward the consent response [VR]

Runner->Runner:22. verify consent id request received [VR].

Runner->MockAA:23. Set the status as "Paused" [VR].

Runner<-MockAA:24.Status Updated [VR]

Runner->MockAA:25. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:26.Trigger Response [VR]

FIU<-MockAA:27. Call the Post-Consent Notification Api [R].

FIU->MockAA:28. Consent notification response [R].

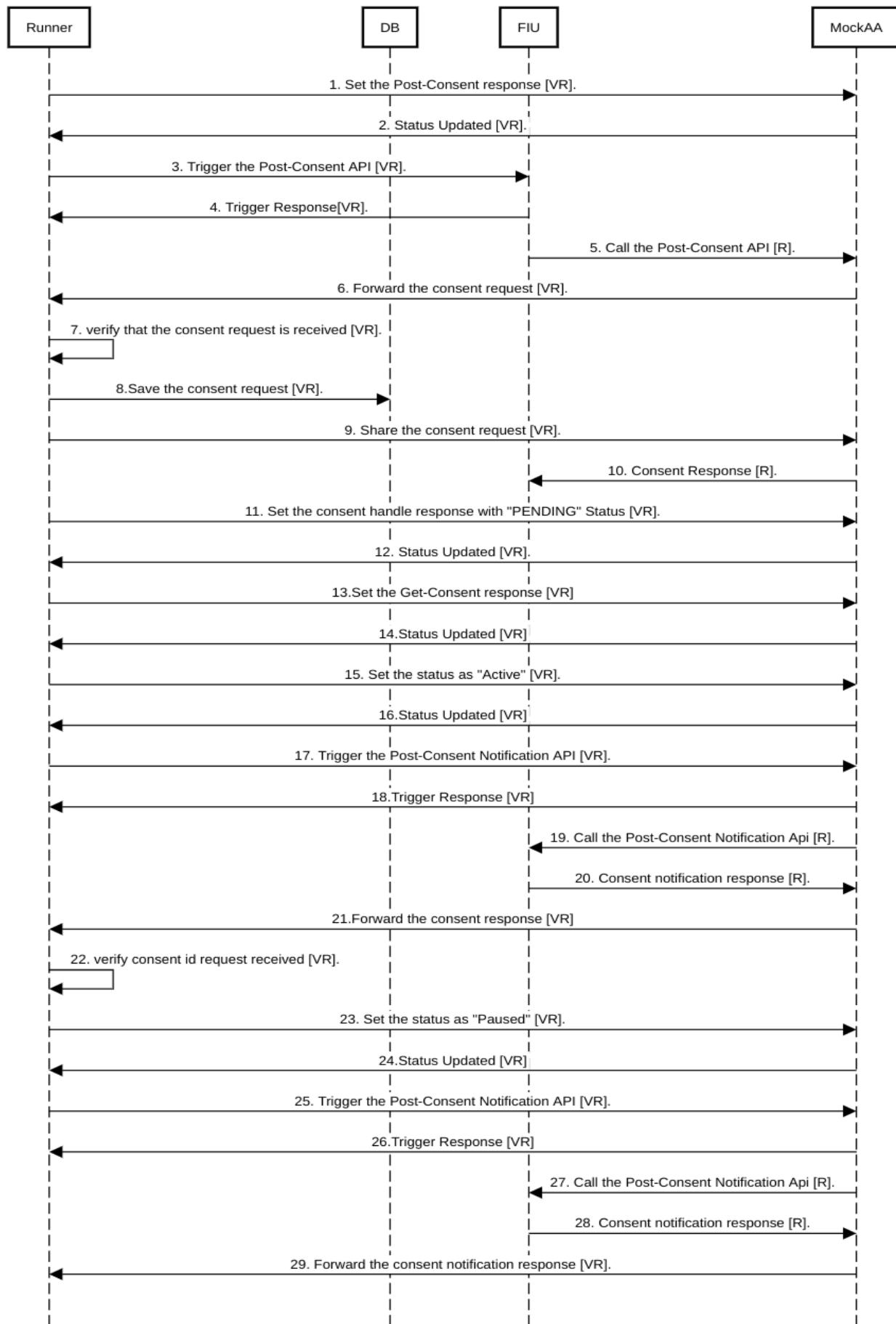
Runner<-MockAA:29. Forward the consent notification response [VR].

Runner ->Runner:28. Verify that the user is not able to do FI Request [VR].

Runner -> DB:29. Store the validation status - success / fail [VR].

Runner->MockAA:30. Response of forwarded consent notification [VR].

FIU_1016__POST _ConsentNotification.feature Scenario 1016_1



FIU_1017_POST_ConsentNotification.feature

Scenario 1017_1

title FIU_1017_POST_ConsentNotification.feature

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Update [VR]

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8.Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13.Set the Get-Consent response [VR]

Runner<-MockAA:14.Status Updated [VR]

Runner->MockAA:15. Set the status as "Active" [VR].

Runner<-MockAA:16.Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18.Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21.Forward the consent response [VR]

Runner->Runner:22. verify consent id request received [VR].

Runner->MockAA:23. Set the status as "EXPIRED" [VR].

Runner<-MockAA:24.Status Updated [VR]

Runner->MockAA:25. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:26.Trigger Response [VR]

FIU<-MockAA:27. Call the Post-Consent Notification Api [R].

FIU->MockAA:28. Consent notification response [R].

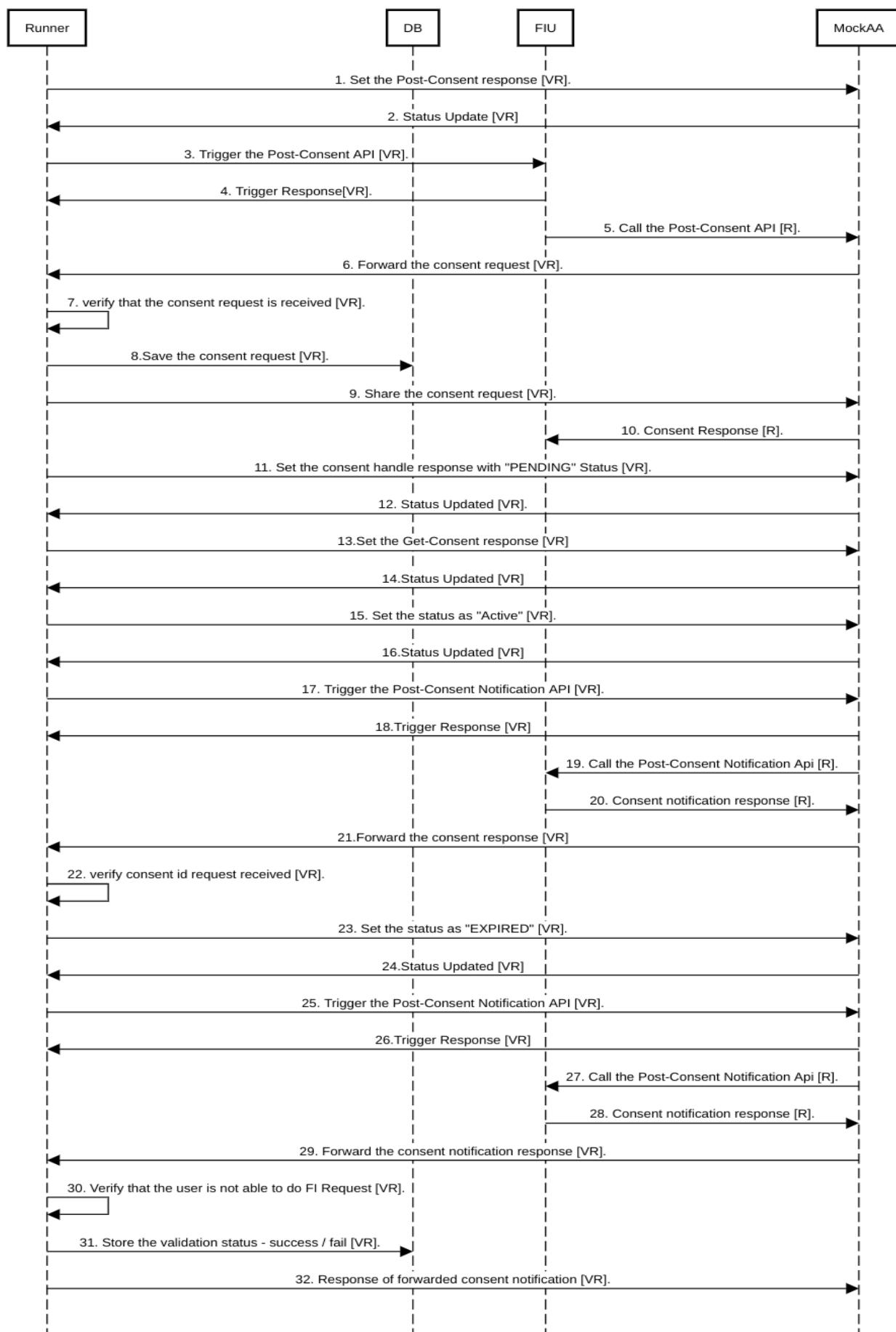
Runner<-MockAA:29. Forward the consent notification response [VR].

Runner ->Runner:30. Verify that the user is not able to do FI Request [VR].

Runner -> DB:31. Store the validation status - success / fail [VR].

Runner->MockAA:32. Response of forwarded consent notification [VR].

FIU_1017__POST__ConsentNotification.feature



FIU_1018__POST__ConsentNotification.feature

Scenario 1018_1

title FIU_1018__POST__ConsentNotification.feature Scenario 1018_1

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Update [VR]

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8.Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent response [VR]

Runner<-MockAA:14. Status Updated [VR]

Runner->MockAA:15. Set the status as "Active" [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner<-MockAA:21. Forward the consent response [VR]

Runner->Runner:22. verify consent id request received [VR].

Runner->MockAA:23. Set the status as "REVOKED" [VR].

Runner<-MockAA:24. Status Updated [VR]

Runner->MockAA:25. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:26. Trigger Response [VR]

FIU<-MockAA:27. Call the Post-Consent Notification Api [R].

FIU->MockAA:28. Consent notification response [R].

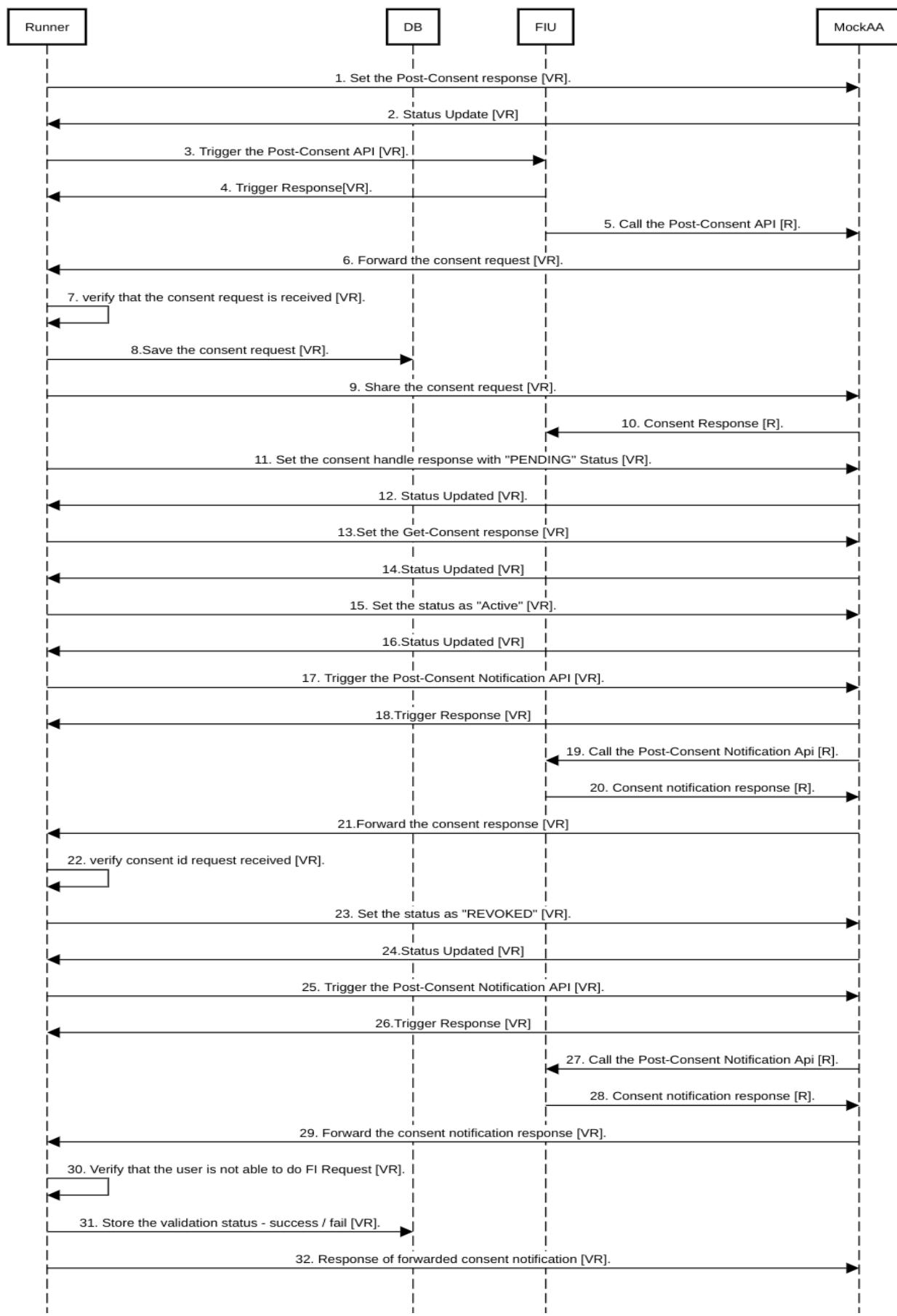
Runner<-MockAA:29. Forward the consent notification response [VR].

Runner ->Runner:30. Verify that the user is not able to do FI Request [VR].

Runner -> DB:31. Store the validation status - success / fail [VR].

Runner->MockAA:32. Response of forwarded consent notification [VR].

FIU_1018__POST _ConsentNotification.feature Scenario 1018_1



FIU_1019_POST_ConsentNotification.feature

Scenario 1019_1

title FIU_1019_POST_ConsentNotification.feature Scenario 1019_1

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response with incorrect version number [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the Get-Consent handle response [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the request with valid values [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-Consent Notification Api [R].

FIU->MockAA:18. Consent notification response [R].

Runner<-MockAA:19. Forward the consent notification response [VR].

Runner ->Runner:20. Verify that the Response code 400 is displayed [VR].

Runner->Runner:21. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

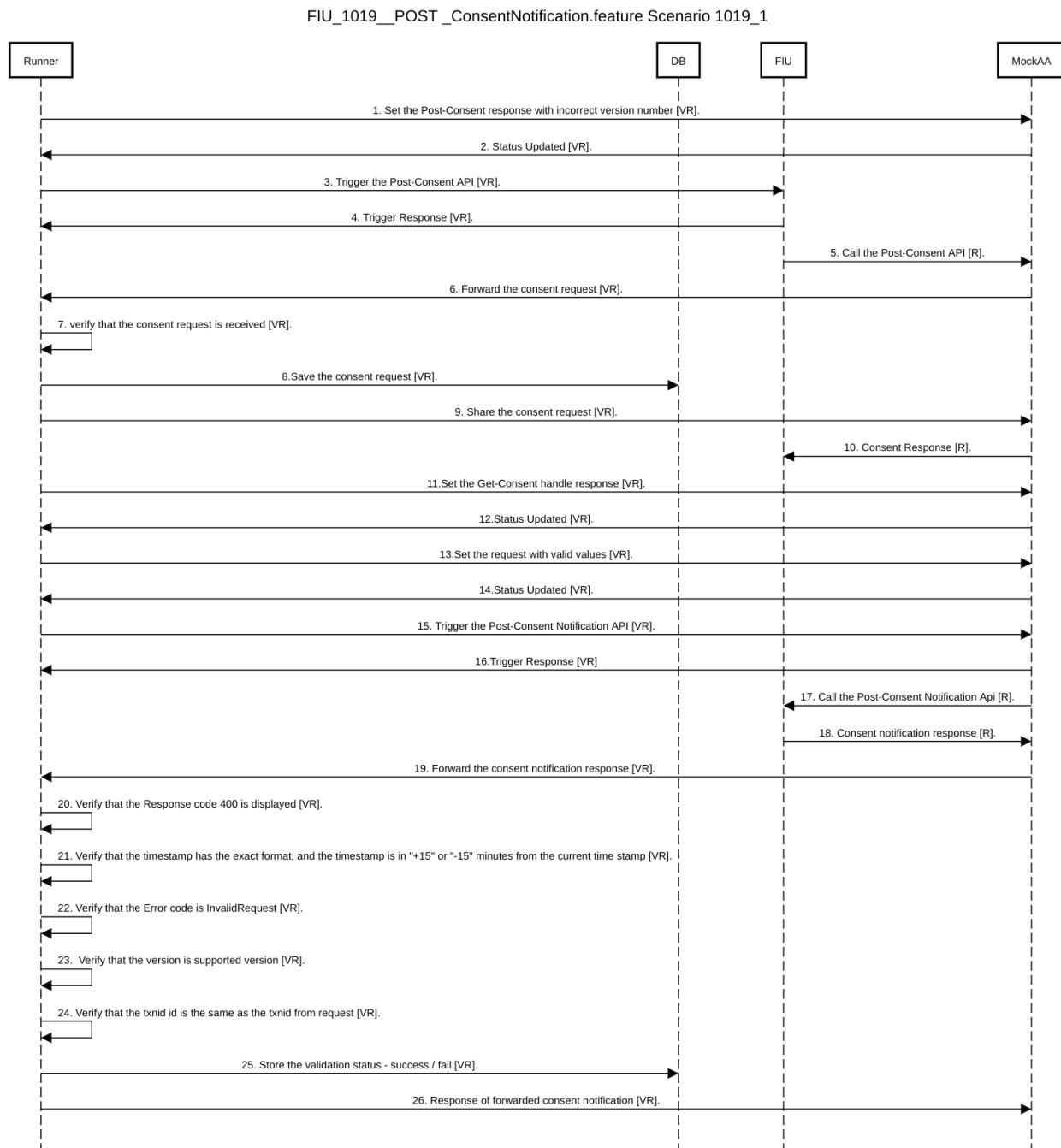
Runner->Runner:22. Verify that the Error code is InvalidRequest [VR].

Runner->Runner:23. Verify that the version is supported version [VR].

Runner->Runner:24. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB:25. Store the validation status – success / fail [VR].

Runner->MockAA:26. Response of forwarded consent notification [VR].



Scenario 1019_2

title FIU_1019__POST _ConsentNotification.feature Scenario 1019_2

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response with an extra digit at the end of the version number [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the Get-Consent handle response [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the request with valid values [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-Consent Notification Api [R].

FIU->MockAA:18. Consent notification response [R].

Runner<-MockAA:19. Forward the consent notification response [VR].

Runner ->Runner:20. Verify that the Response code 400 is displayed [VR].

Runner->Runner:21. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

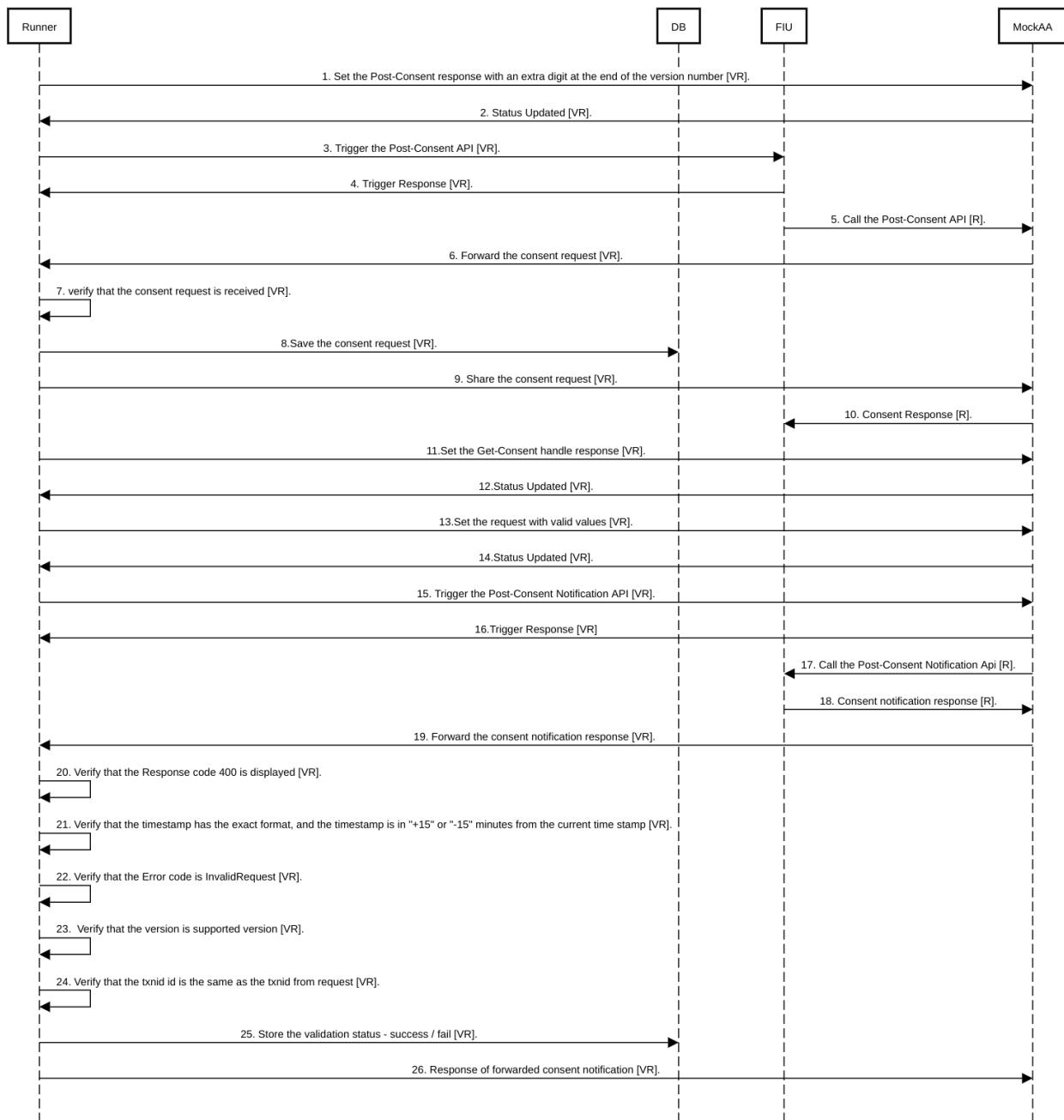
Runner->Runner:22. Verify that the Error code is InvalidRequest [VR].

Runner->Runner:23. Verify that the version is supported version [VR].
 Runner->Runner:24. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB:25. Store the validation status - success / fail [VR].

Runner->MockAA:26. Response of forwarded consent notification [VR].

FIU_1019__POST_ConsentNotification.feature Scenario 1019_2



Scenario 1019_3

title FIU_1019_POST_ConsentNotification.feature Scenario 1019_3

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response by removing the last character or digit at the end of the correct version number [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the Get-Consent handle response [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the request with valid values [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-Consent Notification Api [R].

FIU->MockAA:18. Consent notification response [R].

Runner<-MockAA:19. Forward the consent notification response [VR].

Runner ->Runner:20. Verify that the Response code 400 is displayed [VR].

Runner->Runner:21. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:22. Verify that the Error code is InvalidRequest [VR].

Runner->Runner:23. Verify that the version is supported version [VR].

Runner->Runner:24. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB:25. Store the validation status - success / fail [VR].

Runner->MockAA:26. Response of forwarded consent notification [VR].



FIU_1020_POST_ConsentNotification.feature

Scenario 1020_1

title FIU_1020_POST_ConsentNotification.feature Scenario 1020_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response with future timestamp(current time + 15 minutes) [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the Get-Consent handle response [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the request with valid values [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-Consent Notification Api [R].

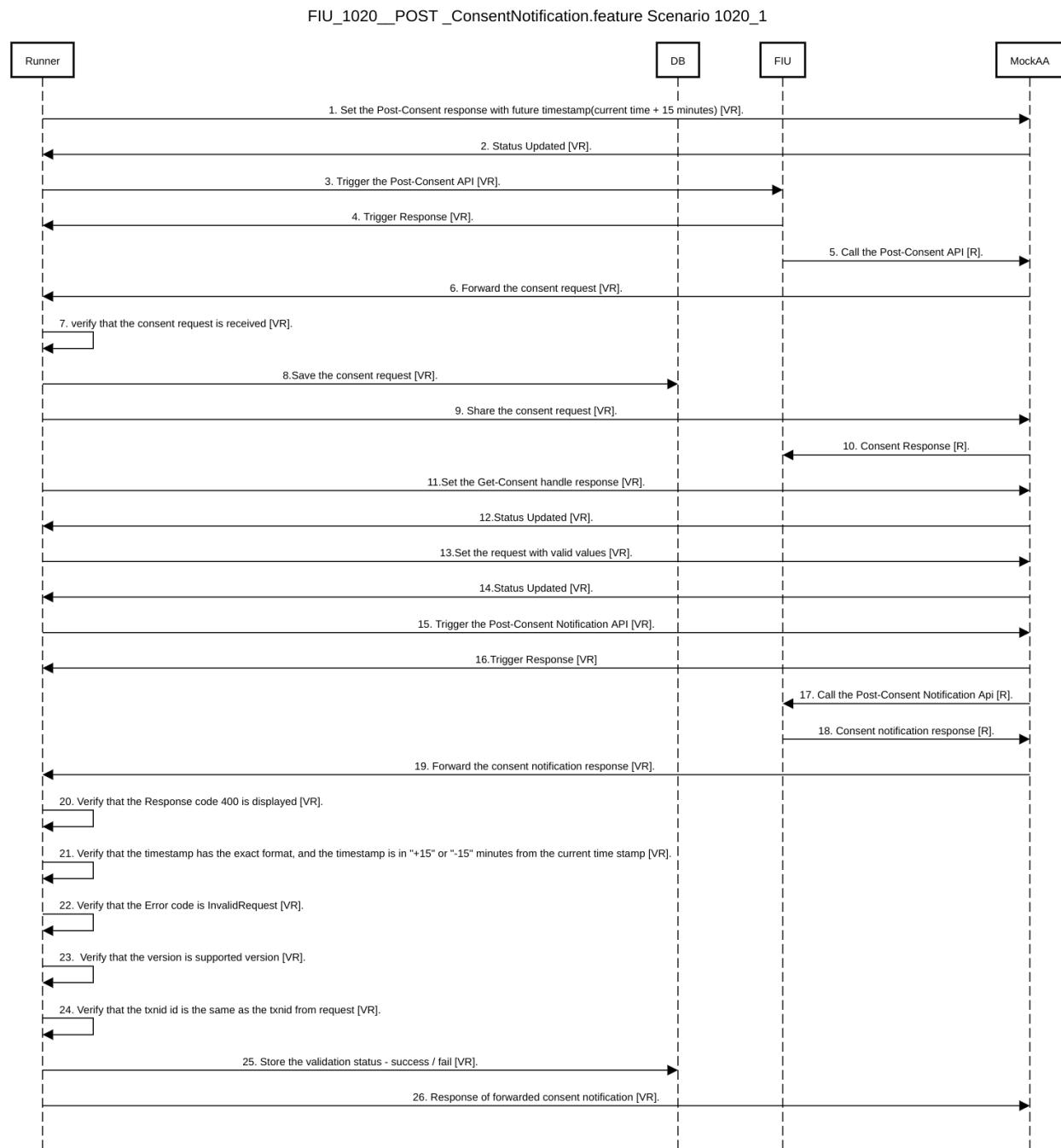
FIU->MockAA:18. Consent notification response [R].

Runner<-MockAA:19. Forward the consent notification response [VR].

Runner ->Runner:20. Verify that the Response code 400 is displayed [VR].

Runner->Runner:21. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

- Runner->Runner:22. Verify that the Error code is InvalidRequest [VR].
 Runner->Runner:23. Verify that the version is supported version [VR].
 Runner->Runner:24. Verify that the txnid id is the same as the txnid from request [VR].
 Runner -> DB:25. Store the validation status - success / fail [VR].
 Runner->MockAA:26. Response of forwarded consent notification [VR].



Scenario 1020_2

title FIU_1020__POST _ConsentNotification.feature Scenario 1020_2

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response with an expired timestamp(current time - 15 minutes) [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the Get-Consent handle response [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the request with valid values [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-Consent Notification Api [R].

FIU->MockAA:18. Consent notification response [R].

Runner<-MockAA:19. Forward the consent notification response [VR].

Runner ->Runner:20. Verify that the Response code 400 is displayed [VR].

Runner->Runner:21. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

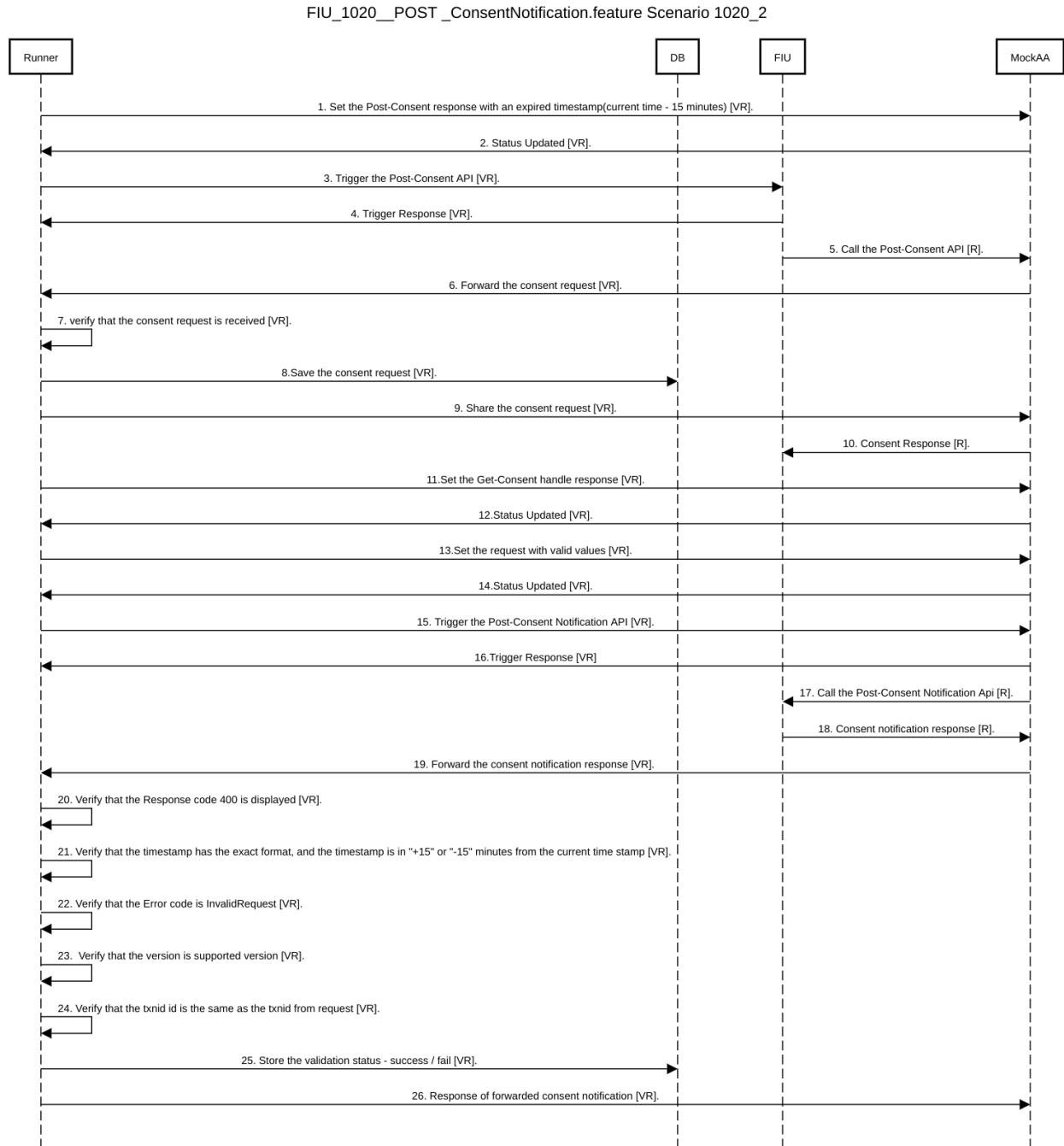
Runner->Runner:22. Verify that the Error code is InvalidRequest [VR].

Runner->Runner:23. Verify that the version is supported version [VR].

Runner->Runner:24. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB:25. Store the validation status - success / fail [VR].

Runner->MockAA:26. Response of forwarded consent notification [VR].



FIU_1021_POST_ConsentNotification.feature

Scenario 1021_1

title FIU_1021_POST_ConsentNotification.feature Scenario 1021_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response with new txnid [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the Get-Consent handle response [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the request with valid values [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-Consent Notification Api [R].

FIU->MockAA:18. Consent notification response [R].

Runner<-MockAA:19. Forward the consent notification response [VR].

Runner -> Runner:20. Verify that the Response code 400 is displayed [VR].

Runner->Runner:21. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:22. Verify that the Error code is InvalidRequest [VR].

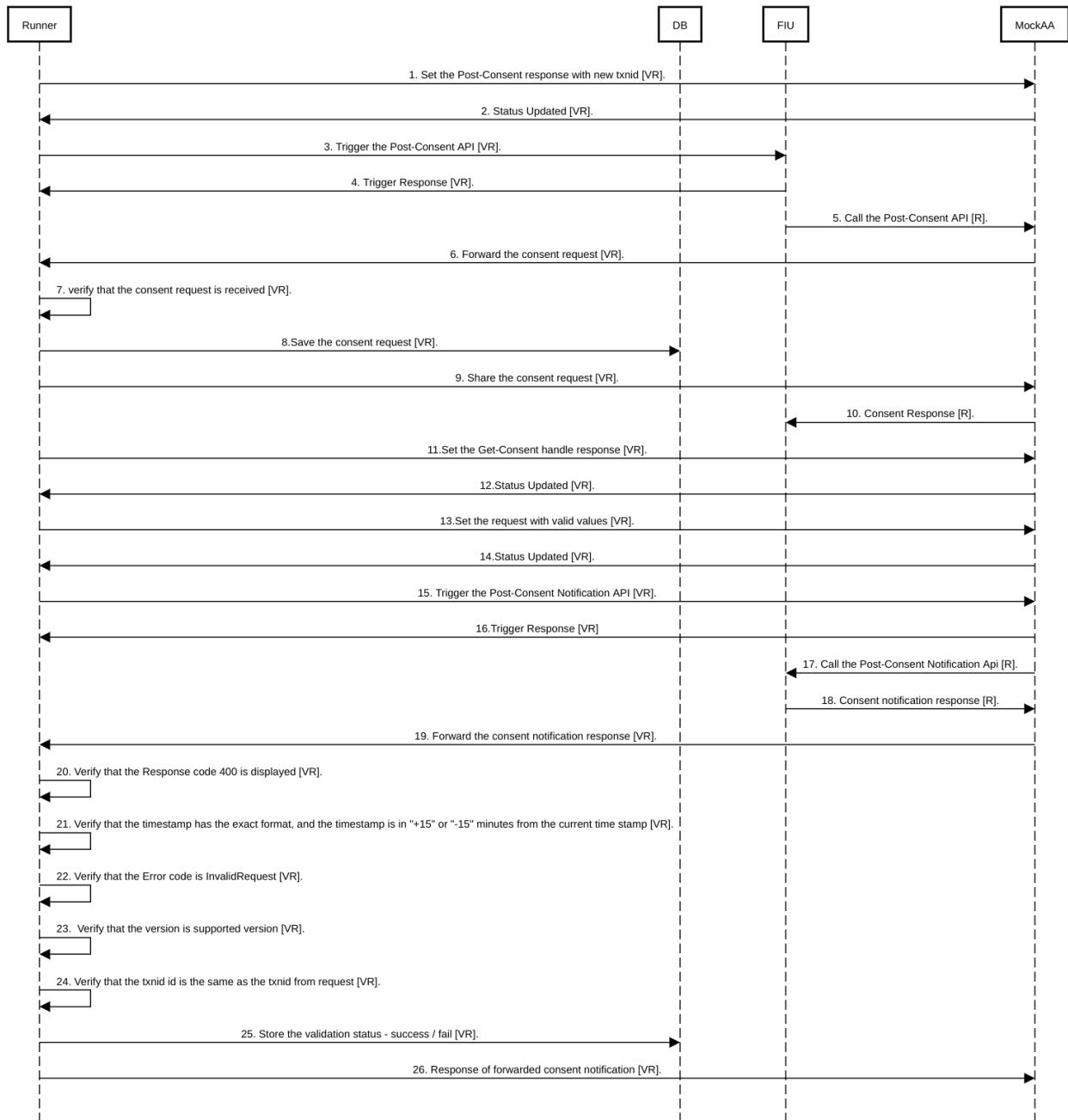
Runner->Runner:23. Verify that the version is supported version [VR].

Runner->Runner:24. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB:25. Store the validation status – success / fail [VR].

Runner->MockAA:26. Response of forwarded consent notification [VR].

FIU_1021__POST _ConsentNotification.feature Scenario 1021_1



FIU_1022_POST_ConsentNotification.feature

Scenario 1022_1

title FIU_1022_POST_ConsentNotification.feature Scenario 1022_1

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response with different CustomerId [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the Get-Consent handle response [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the request with valid values [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-Consent Notification Api [R].

FIU->MockAA:18. Consent notification response [R].

Runner<-MockAA:19. Forward the consent notification response [VR].

Runner ->Runner:20. Verify that the Response code 400 is displayed [VR].

Runner->Runner:21. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:22. Verify that the Error code is InvalidRequest [VR].

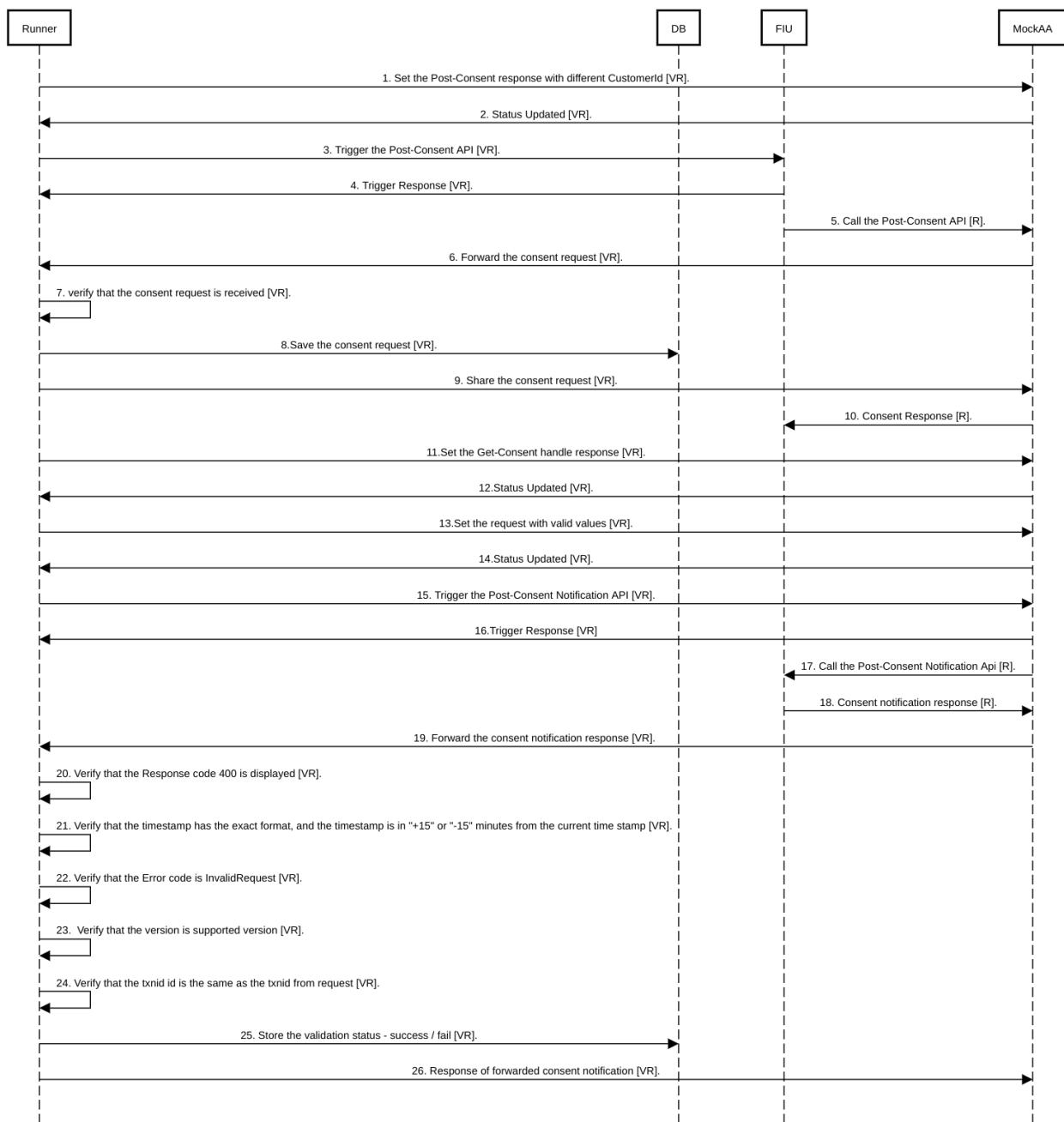
Runner->Runner:23. Verify that the version is supported version [VR].

Runner->Runner:24. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB:25. Store the validation status – success / fail [VR].

Runner->MockAA:26. Response of forwarded consent notification [VR].

FIU_1022__POST _ConsentNotification.feature Scenario 1022_1



FIU_1023_POST_ConsentNotification.feature

Scenario 1023_1

title FIU_1023_POST_ConsentNotification.feature Scenario 1023_1

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response with invalid JWS signature [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the Get-Consent ID response [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:14. Trigger Response [VR]

FIU<-MockAA:15. Call the Post-Consent Notification Api [R].

FIU->MockAA:16. Consent notification response [R].

Runner<-MockAA:17. Forward the consent notification response [VR].

Runner ->Runner:18. Verify that the Response code 400 is displayed [VR].

Runner->Runner:19. Verify that the timestamp has the exact format, and the timestamp is in "+15" minutes from the current time stamp [VR].

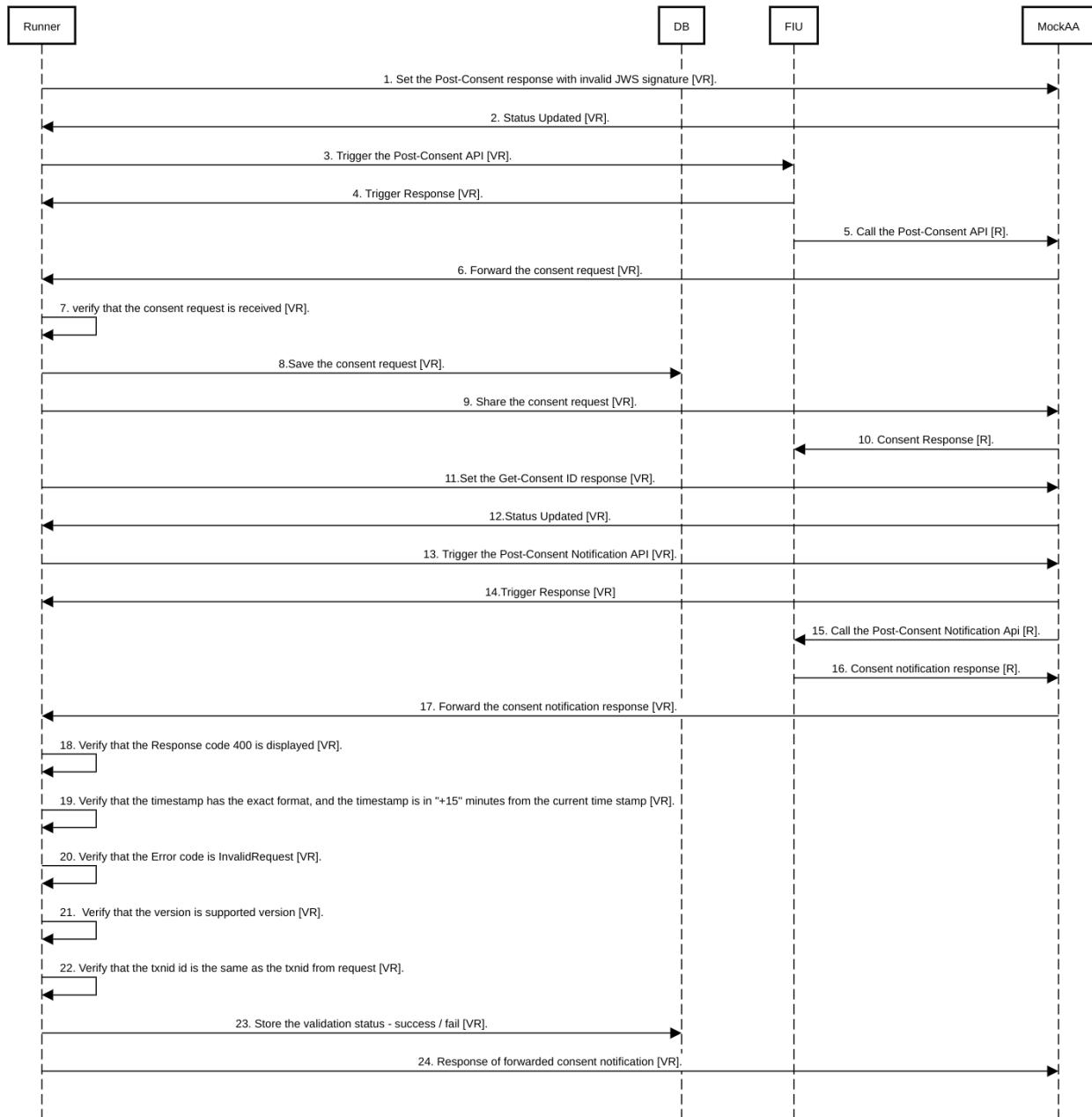
Runner->Runner:20. Verify that the Error code is InvalidRequest [VR].

Runner->Runner:21. Verify that the version is supported version [VR].
 Runner->Runner:22. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB:23. Store the validation status - success / fail [VR].

Runner->MockAA:24. Response of forwarded consent notification [VR].

FIU_1023 __ POST _ConsentNotification.feature Scenario 1023_1



FIU_1024_POST_ConsentNotification.feature

Scenario 1024_1

title FIU_1024_POST_ConsentNotification.feature Scenario 1024_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set consent handle request with incorrect version number [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->FIU:13. Trigger the Get-Consent handle Api [VR].

Runner<-FIU:14. Trigger Response [VR]

FIU->MockAA:15. Call the Get-Consent handle Api [R].

Runner<-MockAA:16. Forward the consent handle request [VR]

Runner->Runner:17. Verify the consent handle request is received [VR].

Runner->MockAA:18. Share the consent handle request [VR]

FIU<-MockAA:19. Consent handle Response[R].

Runner->MockAA:20. Trigger the Post-Consent Notification API [VR].

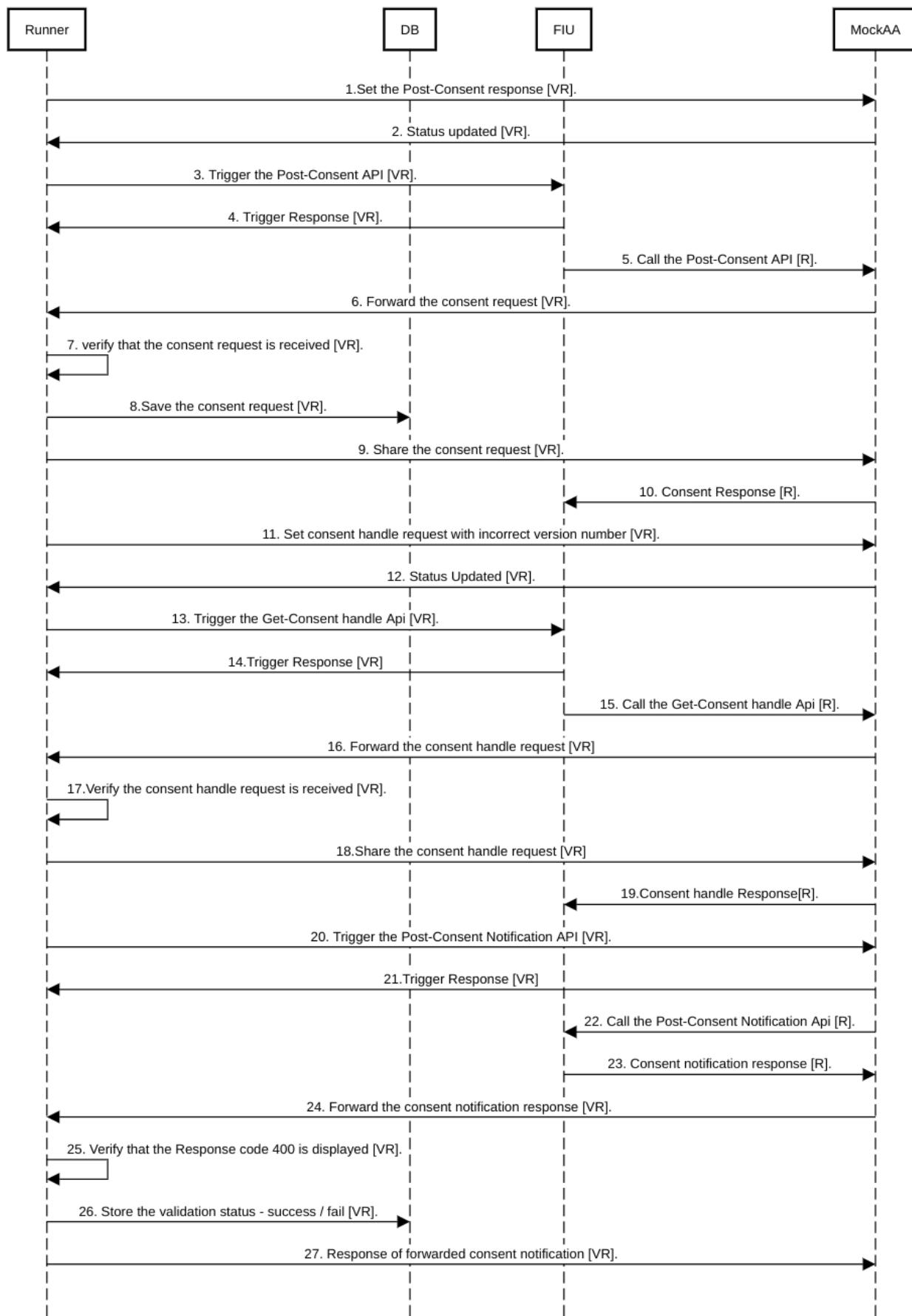
Runner<-MockAA:21. Trigger Response [VR]

FIU<-MockAA:22. Call the Post-Consent Notification Api [R].

FIU->MockAA:23. Consent notification response [R].

Runner<-MockAA:24. Forward the consent notification response [VR].
Runner ->Runner:25. Verify that the Response code 400 is displayed [VR].
Runner -> DB: 26. Store the validation status - success / fail [VR].
Runner->MockAA:27. Response of forwarded consent notification [VR].

FIU_1024__POST _ConsentNotification.feature Scenario 1024_1



Scenario 1024_2

title FIU_1024_POST _ConsentNotification.feature Scenario 1024_2

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set consent handle request with an extra digit at the end of the correct version number [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->FIU:13. Trigger the Get-Consent handle Api [VR].

Runner<-FIU:14. Trigger Response [VR]

FIU->MockAA:15. Call the Get-Consent handle Api [R].

Runner<-MockAA:16. Forward the consent handle request [VR]

Runner->Runner:17. Verify the consent handle request is received [VR].

Runner->MockAA:18. Share the consent handle request [VR]

FIU<-MockAA:19. Consent handle Response[R].

Runner->MockAA:20. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:21. Trigger Response [VR]

FIU<-MockAA:22. Call the Post-Consent Notification Api [R].

FIU->MockAA:23. Consent notification response [R].

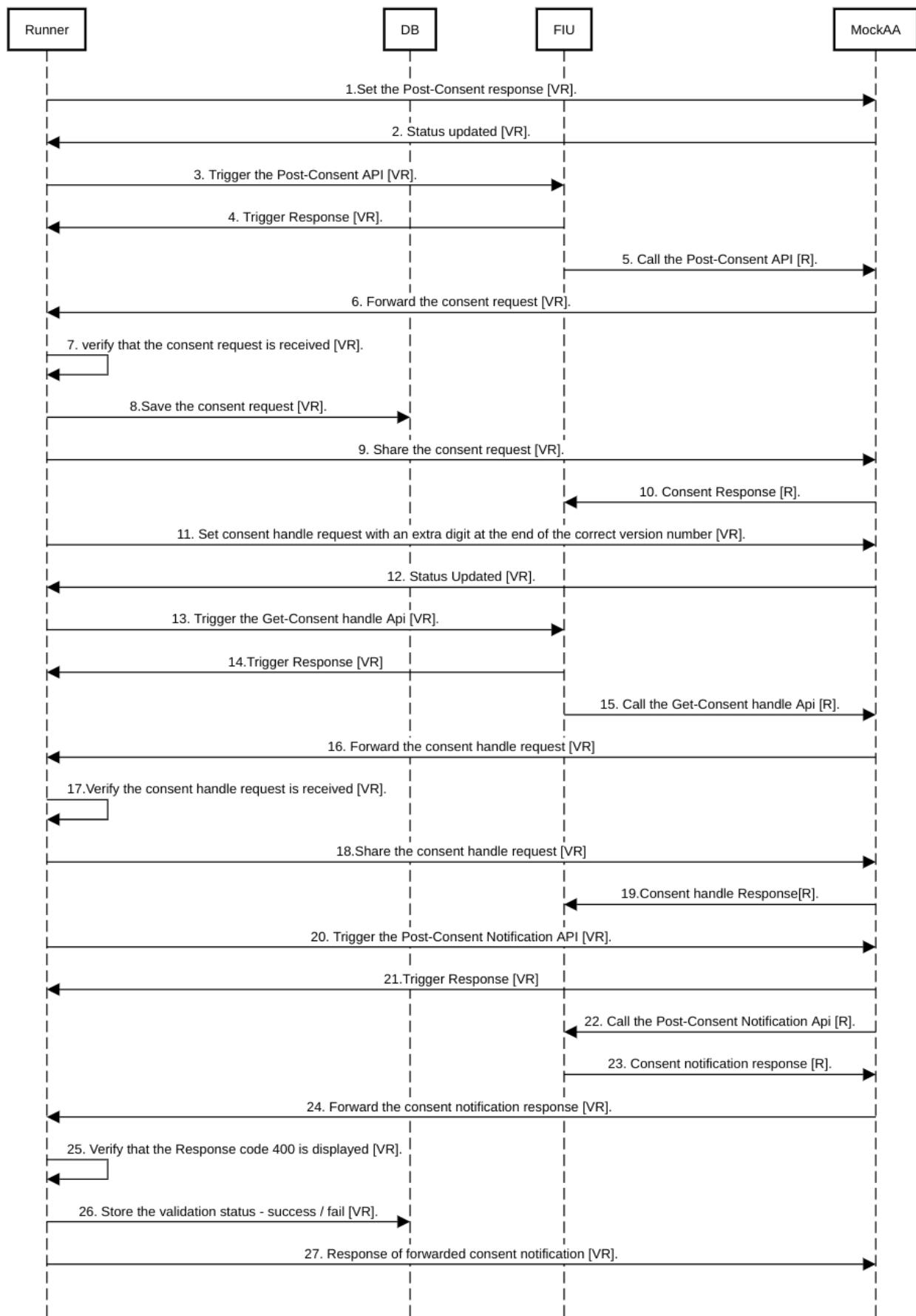
Runner<-MockAA:24. Forward the consent notification response [VR].

Runner ->Runner:25. Verify that the Response code 400 is displayed [VR].

Runner -> DB: 26. Store the validation status - success / fail [VR].

Runner->MockAA:27. Response of forwarded consent notification [VR].

FIU_1024__POST _ConsentNotification.feature Scenario 1024_2



Scenario 1024_3

title FIU_1024_POST _ConsentNotification.feature Scenario 1024_3

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set consent handle request by removing the last character or digit at the end of the correct version number[VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->FIU:13. Trigger the Get-Consent handle Api [VR].

Runner<-FIU:14. Trigger Response [VR]

FIU->MockAA:15. Call the Get-Consent handle Api [R].

Runner<-MockAA:16. Forward the consent handle request [VR]

Runner->Runner:17.Verify the consent handle request is received [VR].

Runner->MockAA:18.Share the consent handle request [VR]

FIU<-MockAA:19.Consent handle Response[R].

Runner->MockAA:20. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:21.Trigger Response [VR]

FIU<-MockAA:22. Call the Post-Consent Notification Api [R].

FIU->MockAA:23. Consent notification response [R].

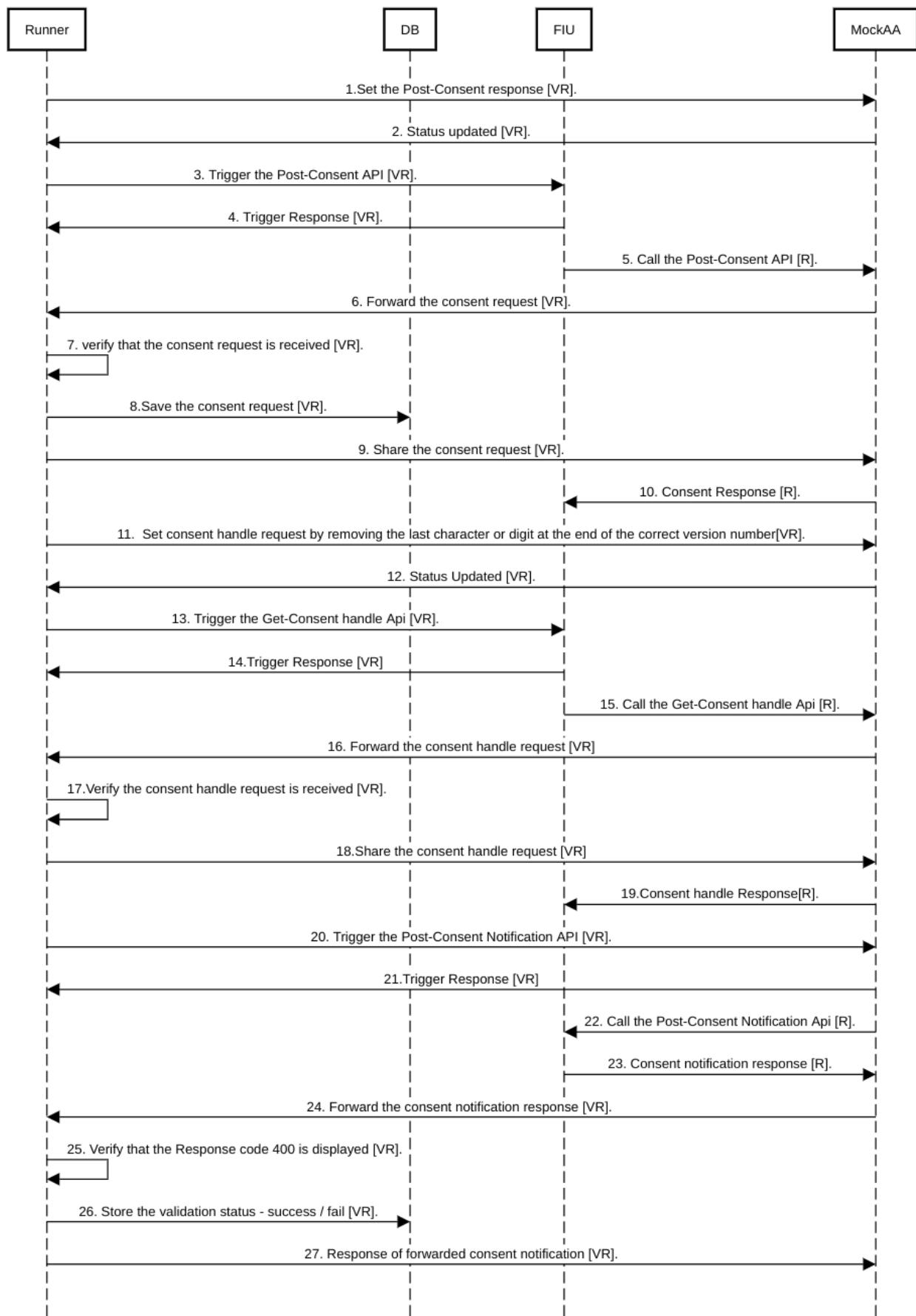
Runner<-MockAA:24. Forward the consent notification response [VR].

Runner ->Runner:25. Verify that the Response code 400 is displayed [VR].

Runner -> DB: 26. Store the validation status - success / fail [VR].

Runner->MockAA:27. Response of forwarded consent notification [VR].

FIU_1024__POST _ConsentNotification.feature Scenario 1024_3



FIU_1025__GET__ConsentHandle.feature

Scenario 1025_1

title FIU_1025__GET__ConsentHandle.feature Scenario 1025_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set consent handle request with future timestamp(current time + 15 minutes) [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->FIU:13. Trigger the Get-Consent handle Api [VR].

Runner<-FIU:14. Trigger Response [VR]

FIU->MockAA:15. Call the Get-Consent handle Api [R].

Runner<-MockAA:16. Forward the consent handle request [VR]

Runner->Runner:17.Verify the consent handle request is received [VR].

Runner->MockAA:18.Share the consent handle request [VR]

FIU<-MockAA:19.Consent handle Response[R].

Runner->MockAA:20. Trigger the Post-Consent Notification API [VR].

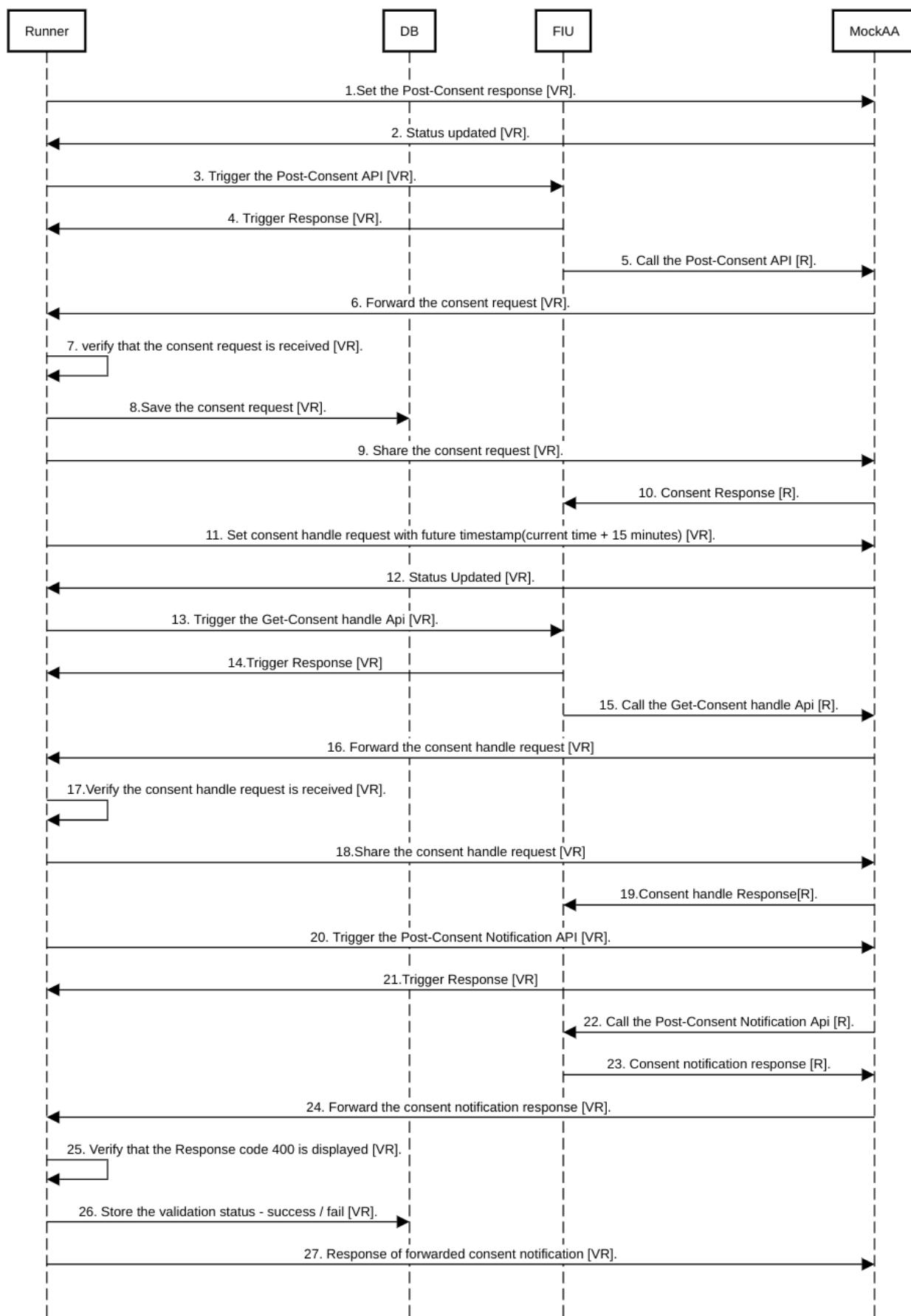
Runner<-MockAA:21.Trigger Response [VR]

FIU<-MockAA:22. Call the Post-Consent Notification Api [R].

FIU->MockAA:23. Consent notification response [R].

Runner<-MockAA:24. Forward the consent notification response [VR].
Runner ->Runner:25. Verify that the Response code 400 is displayed [VR].
Runner -> DB: 26. Store the validation status - success / fail [VR].
Runner->MockAA:27. Response of forwarded consent notification [VR].

FIU_1025__GET _ConsentHandle.feature Scenario 1025_1



Scenario 1025_2

title FIU_1025__GET _ConsentHandle.feature Scenario 1025_2

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set consent handle request with expired timestamp(current time - 15 minutes) [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->FIU:13. Trigger the Get-Consent handle Api [VR].

Runner<-FIU:14. Trigger Response [VR]

FIU->MockAA:15. Call the Get-Consent handle Api [R].

Runner<-MockAA:16. Forward the consent handle request [VR]

Runner->Runner:17. Verify the consent handle request is received [VR].

Runner->MockAA:18. Share the consent handle request [VR]

FIU<-MockAA:19. Consent handle Response[R].

Runner->MockAA:20. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:21. Trigger Response [VR]

FIU<-MockAA:22. Call the Post-Consent Notification Api [R].

FIU->MockAA:23. Consent notification response [R].

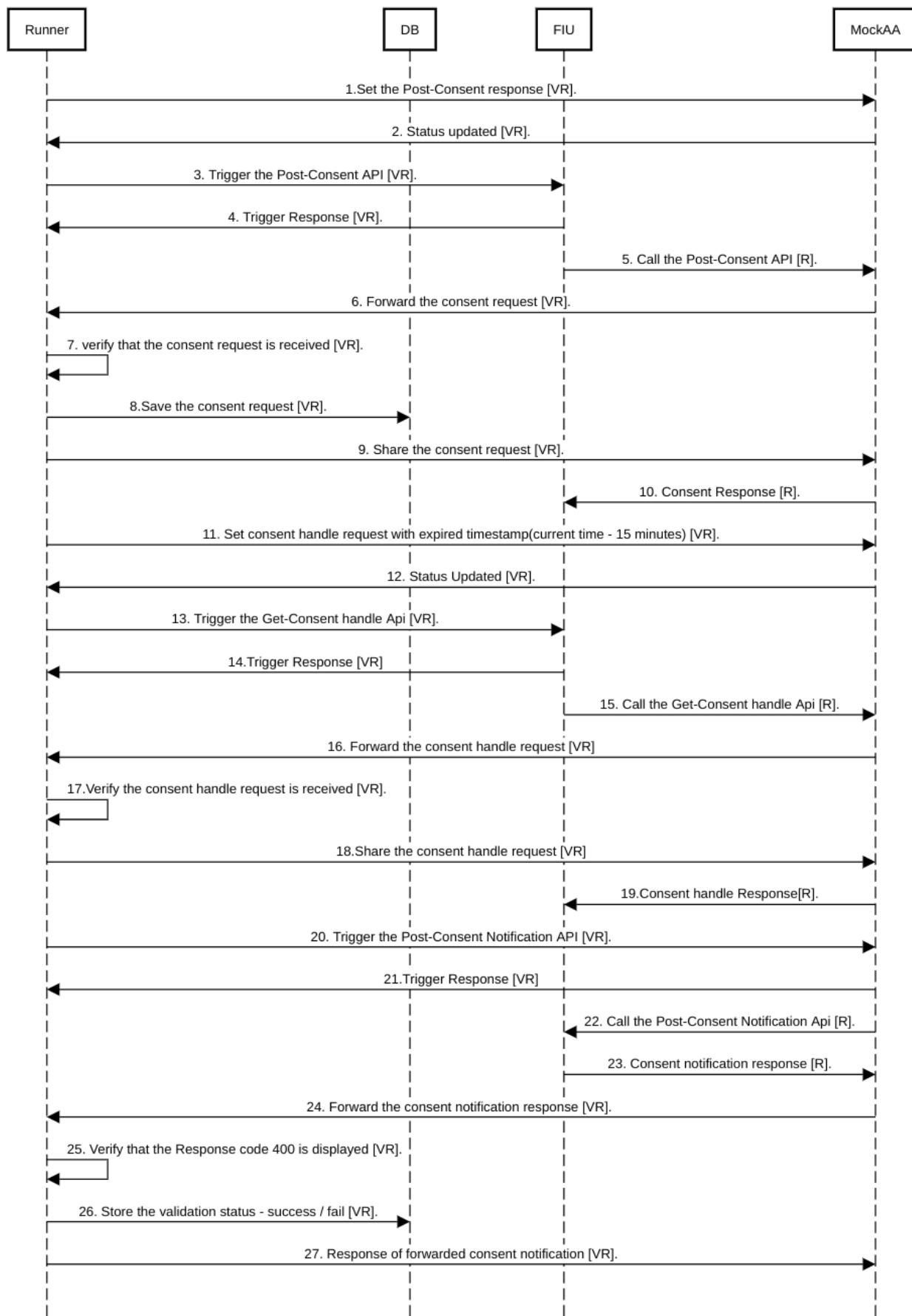
Runner<-MockAA:24. Forward the consent notification response [VR].

Runner ->Runner:25. Verify that the Response code 400 is displayed [VR].

Runner -> DB: 26. Store the validation status - success / fail [VR].

Runner->MockAA:27. Response of forwarded consent notification [VR].

FIU_1025__GET _ConsentHandle.feature Scenario 1025_2



FIU_1026_POST_ConsentNotification.feature

Scenario 1026_1

title FIU_1026_POST_ConsentNotification.feature Scenario 1026_1

participant Runner
participant DB
participant FIU
participant MockAA
participant Alternate MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->Alternate MockAA:3. Set the Post-Consent response [VR].

Runner<-Alternate MockAA:4. Status Updated [VR].

Runner->FIU:5 Trigger the Post-Consent API [VR].

Runner<-FIU:6 Trigger Response[VR].

FIU->MockAA:7 Call the Post-Consent API [R].

Runner<-MockAA:8 Forward the consent request [VR].

Runner->Runner:9 verify that the consent request is received [VR].

Runner -> DB:10. Save the consent request [VR].

Runner->MockAA:11. Share the consent request [VR].

FIU<-MockAA:12.Consent Response [R].

FIU->Alternate MockAA:13. Call the Post-Consent API [VR].

Runner<-Alternate MockAA:14. Forward the alternate consent request [VR]

Runner->Runner:15. verify that the alternate consent request is received [VR].

Runner->DB:16. Save the alternate consent request [VR].

Runner->Alternate MockAA:17. Share the alternate consent request [VR].

FIU<-Alternate MockAA:18. Consent Response [VR]

Runner->MockAA:19. Set the consent handle response with consentHandle of Alternate Mock [VR].

Runner<-MockAA:20. Status Updated [VR].

Runner->MockAA:21. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:22. Trigger Response [VR]

FIU<-MockAA:23. Call the Post-Consent Notification Api [R].

FIU->MockAA:24. Consent notification response [R].

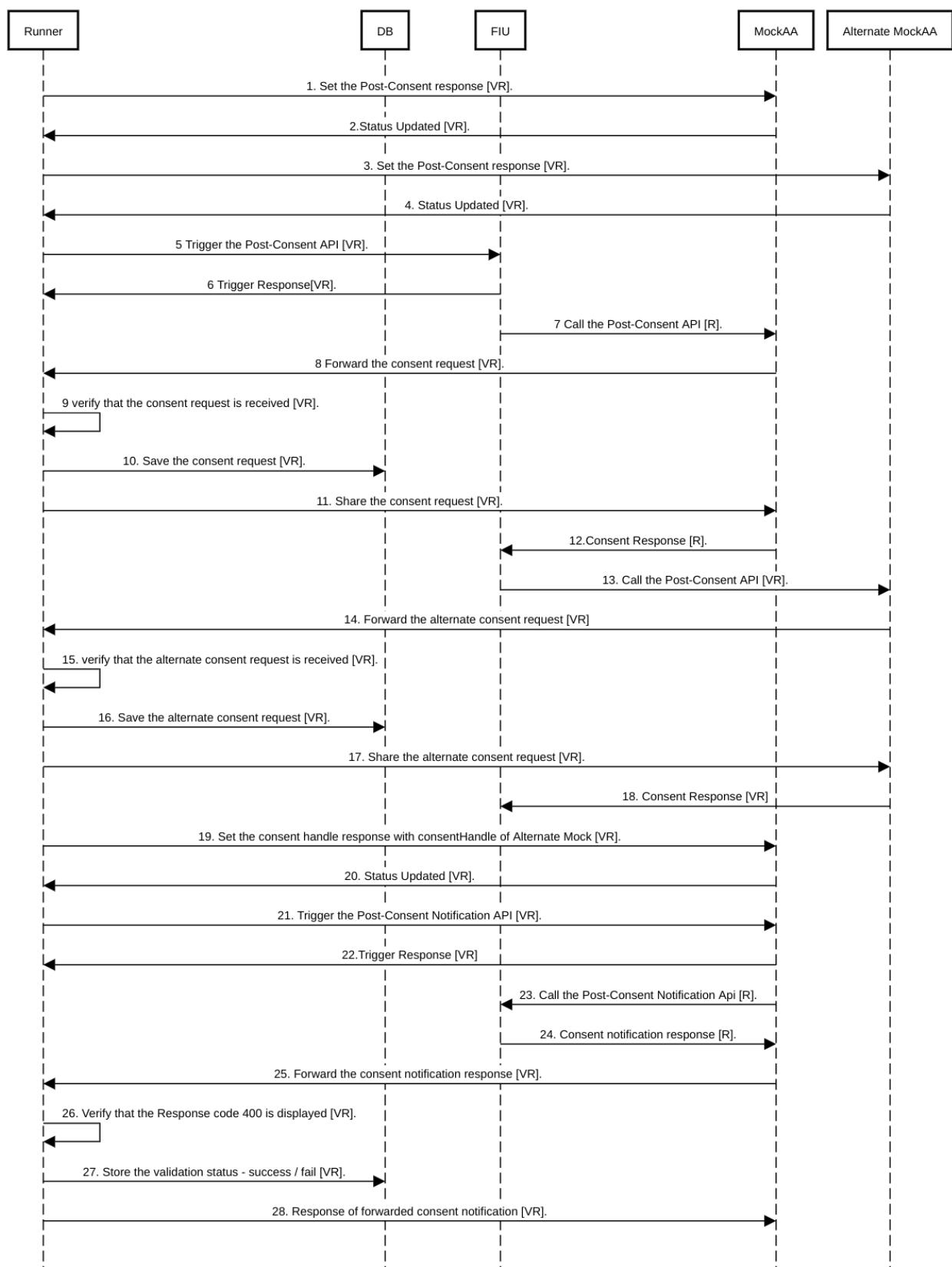
Runner<-MockAA:25. Forward the consent notification response [VR].

Runner ->Runner:26. Verify that the Response code 400 is displayed [VR].

Runner -> DB:27. Store the validation status - success / fail [VR].

Runner->MockAA:28. Response of forwarded consent notification [VR].

FIU_1026_POST_ConsentNotification.feature Scenario 1026_1



FIU_1028_POST_ConsentNotification.feature

Scenario 1028_1

title FIU_1028_POST_ConsentNotification.feature Scenario 1028_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the ConsentStatus as "FAILED" [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->FIU:13. Trigger the Get-Consent handle Api [VR].

Runner<-FIU:14. Trigger Response [VR]

FIU->MockAA:15. Call the Get-Consent handle Api [R].

Runner<-MockAA:16. Forward the consent handle request [VR].

Runner->Runner:17. verify that the consent handle request is received [VR].

Runner -> DB:18. Save the consent handle request [VR].

Runner->MockAA:19. Share the consent handle request [VR].

FIU<-MockAA:20. Consent handle Response[R].

Runner->MockAA:21. Trigger the Post-Consent Notification API [VR].

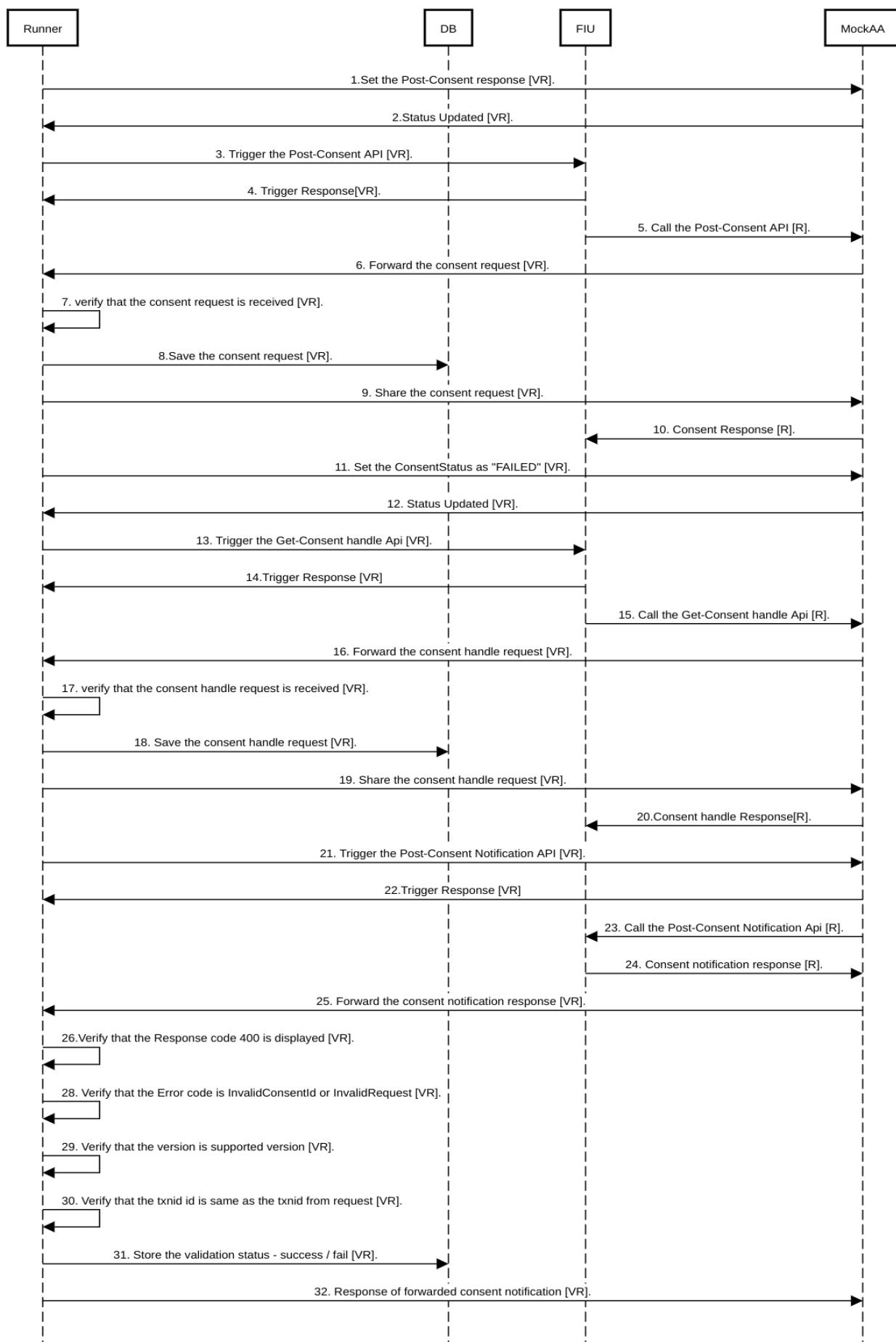
Runner<-MockAA:22. Trigger Response [VR]

FIU<-MockAA:23. Call the Post-Consent Notification Api [R].

FIU->MockAA:24. Consent notification response [R].

Runner<-MockAA:25. Forward the consent notification response [VR].
Runner ->Runner :26.Verify that the Response code 400 is displayed [VR].
Runner ->Runner 27.Verify that the timestamp has the exact format, and the timestamp is in "+15" minutes from the current time stamp [VR].
Runner ->Runner : 28. Verify that the Error code is InvalidConsentId or InvalidRequest [VR].
Runner ->Runner : 29. Verify that the version is supported version [VR].
Runner ->Runner :30. Verify that the txnid id is same as the txnid from request [VR].
Runner ->DB : 31. Store the validation status - success / fail [VR].
Runner->MockAA:32. Response of forwarded consent notification [VR].

FIU_1028__POST__ConsentNotification.feature Scenario 1028_1



FIU_1029_POST_ConsentNotification.feature

Scenario 1029_1

title FIU_1029_POST_ConsentNotification.feature Scenario 1029_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the ConsentStatus as "PENDING" [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->FIU:13. Trigger the Get-Consent handle Api [VR].

Runner<-FIU:14. Trigger Response [VR]

FIU->MockAA:15. Call the Get-Consent handle Api [R].

Runner<-MockAA:16. Forward the consent handle request [VR].

Runner->Runner:17. verify that the consent handle request is received [VR].

Runner -> DB:18. Save the consent handle request [VR].

Runner->MockAA:19. Share the consent handle request [VR].

FIU<-MockAA:20. Consent handle Response[R].

Runner->MockAA:21. Trigger the Post-Consent Notification API [VR].

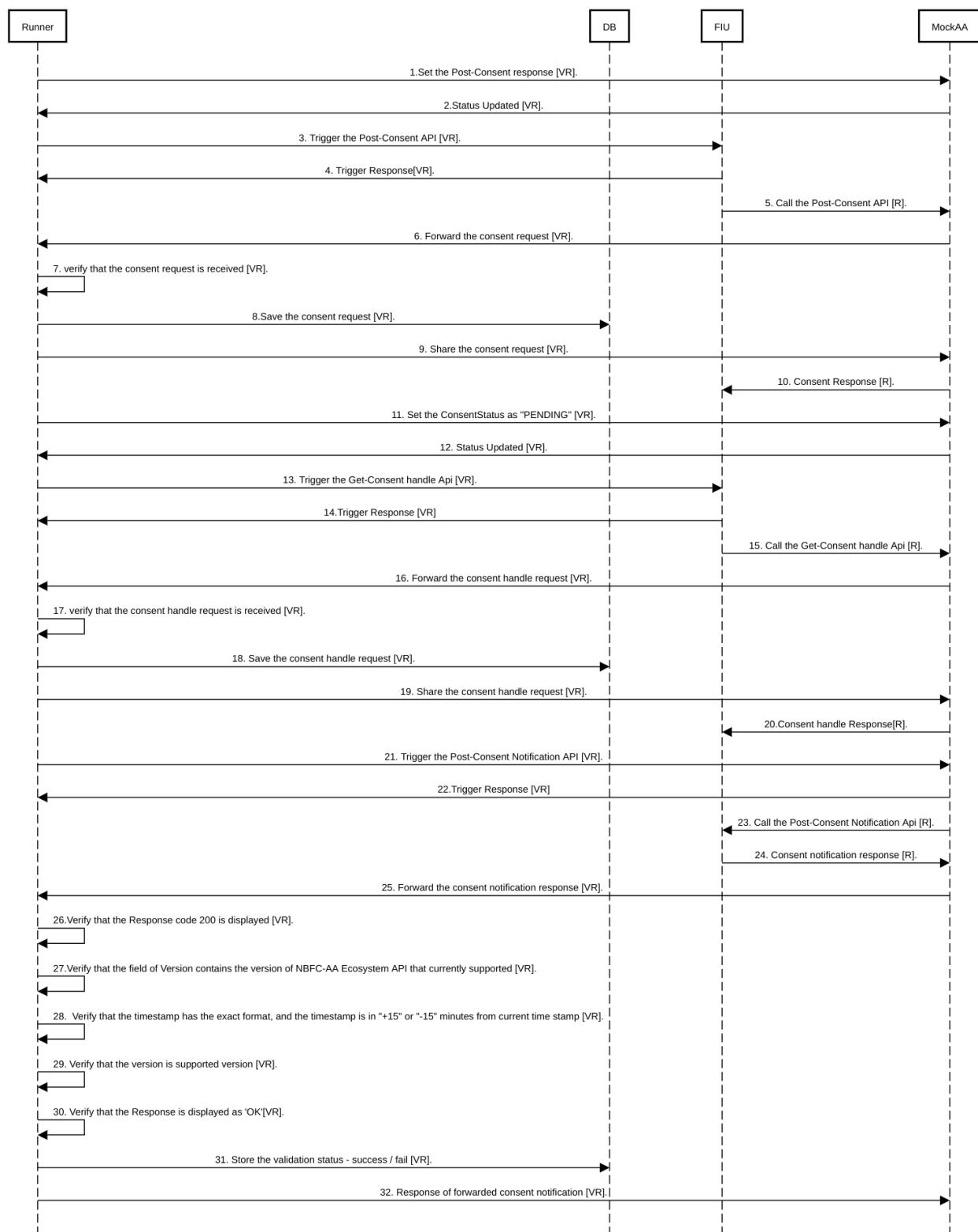
Runner<-MockAA:22. Trigger Response [VR]

FIU<-MockAA:23. Call the Post-Consent Notification Api [R].

FIU->MockAA:24. Consent notification response [R].

Runner<-MockAA:25. Forward the consent notification response [VR].
Runner ->Runner : 26.Verify that the Response code 200 is displayed [VR].
Runner ->Runner : 27.Verify that the field of Version contains the version of NBFC-AA Ecosystem API that currently supported [VR].
Runner ->Runner : 28. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from current time stamp [VR].
Runner ->Runner: 29. Verify that the version is supported version [VR].
Runner ->Runner : 30. Verify that the Response is displayed as 'OK'[VR].
Runner -> DB : 31. Store the validation status - success / fail [VR].
Runner->MockAA:32. Response of forwarded consent notification [VR].

FIU_1029__POST _ConsentNotification.feature Scenario 1029_1



FIU_1030_POST_ConsentNotification.feature

Scenario 1030_1

title FIU_1030_POST_ConsentNotification.feature Scenario 1030_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle with invalid JWS signature [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->FIU:13. Trigger the Get-Consent handle Api [VR].

Runner<-FIU:14. Trigger Response [VR]

FIU->MockAA:15. Call the Get-Consent handle Api [R].

Runner<-MockAA:16. Forward the consent handle request [VR].

Runner->Runner:17. verify that the consent handle request is received [VR].

Runner -> DB:18. Save the consent handle request [VR].

Runner->MockAA:19. Share the consent handle request [VR].

FIU<-MockAA:20. Consent handle Response[R].

Runner->MockAA:21. Trigger the Post-Consent Notification API [VR].

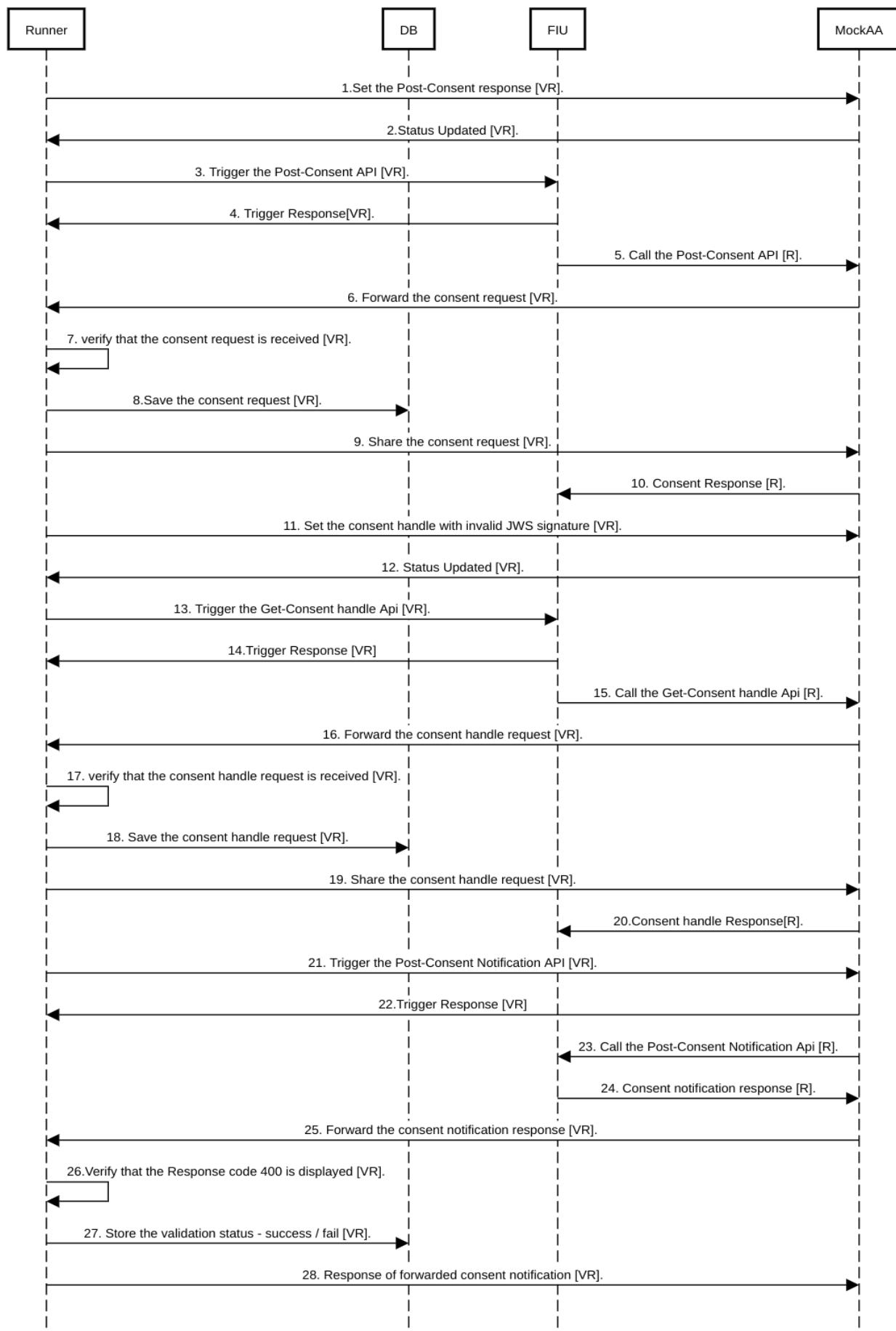
Runner<-MockAA:22. Trigger Response [VR]

FIU<-MockAA:23. Call the Post-Consent Notification Api [R].

FIU->MockAA:24. Consent notification response [R].

Runner<-MockAA:25. Forward the consent notification response [VR].
Runner ->Runner : 26.Verify that the Response code 400 is displayed [VR].
Runner -> DB : 27. Store the validation status - success / fail [VR].
Runner->MockAA:28. Response of forwarded consent notification [VR].

FIU_1030__POST _ConsentNotification.feature Scenario 1030_1



FIU_1031__GET _ConsentNotification.feature

Scenario 1031_1

title FIU_1031__GET _ConsentNotification.feature Scenario 1031_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR]

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent response with incorrect version number [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set consent notification request status as "ACTIVE" [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner->FIU:21. Trigger The Get-Consent Handle API [VR]

Runner<-FIU:22. Trigger Response [VR].

FIU->MockAA:23. Call the Get-Consent Handle API [R].

FIU<-MockAA:24. Consent Handle Response [R]

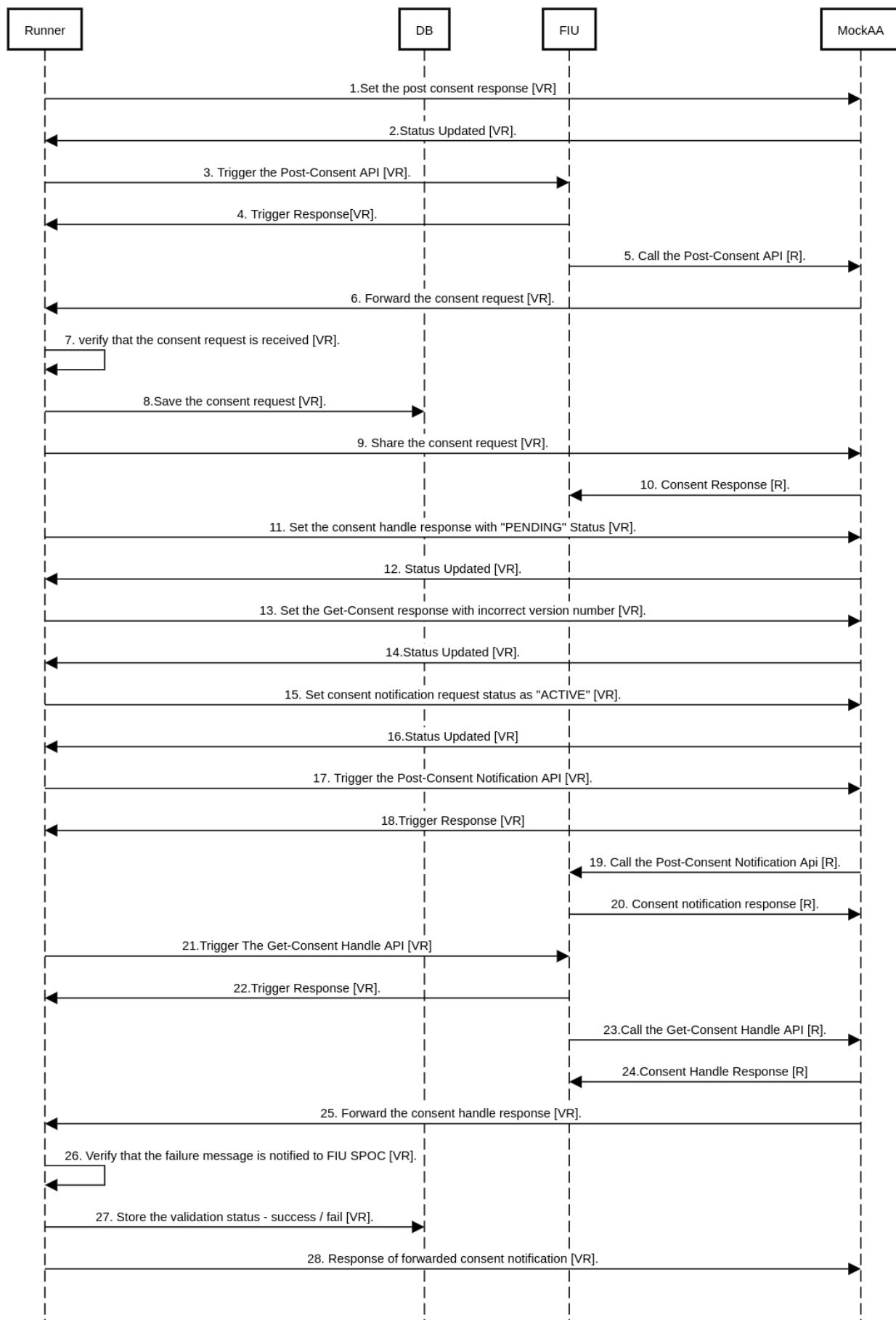
Runner<-MockAA:25. Forward the consent handle response [VR].

Runner ->Runner:26. Verify that the failure message is notified to FIU SPOC [VR].

Runner -> DB:27. Store the validation status - success / fail [VR].

Runner->MockAA:28. Response of forwarded consent notification [VR].

FIU_1031__GET _ConsentNotification.feature Scenario 1031_1



Scenario 1031_2

title FIU_1031__GET _ConsentNotification.feature Scenario 1031_2

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call
R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR]
Runner<-MockAA:2. Status Updated [VR].
Runner->FIU:3. Trigger the Post-Consent API [VR].
Runner<-FIU:4. Trigger Response[VR].
FIU->MockAA:5. Call the Post-Consent API [R].
Runner<-MockAA:6. Forward the consent request [VR].
Runner->Runner:7. verify that the consent request is received [VR].
Runner -> DB:8. Save the consent request [VR].
Runner->MockAA:9. Share the consent request [VR].
FIU<-MockAA:10. Consent Response [R].
Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].
Runner<-MockAA:12. Status Updated [VR].
Runner->MockAA:13. Set the Get-Consent response by adding an extra digit at the end of the correct version number [VR].
Runner<-MockAA:14. Status Updated [VR].
Runner->MockAA:15. Set consent notification request status as "ACTIVE" [VR].
Runner<-MockAA:16. Status Updated [VR]
Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].
Runner<-MockAA:18. Trigger Response [VR]
FIU<-MockAA:19. Call the Post-Consent Notification Api [R].
FIU->MockAA:20. Consent notification response [R].
Runner->FIU:21. Trigger The Get-Consent Handle API [VR]

Runner<-FIU:22.Trigger Response [VR].

FIU->MockAA:23.Call the Get-Consent Handle API [R].

FIU<-MockAA:24.Consent Handle Response [R]

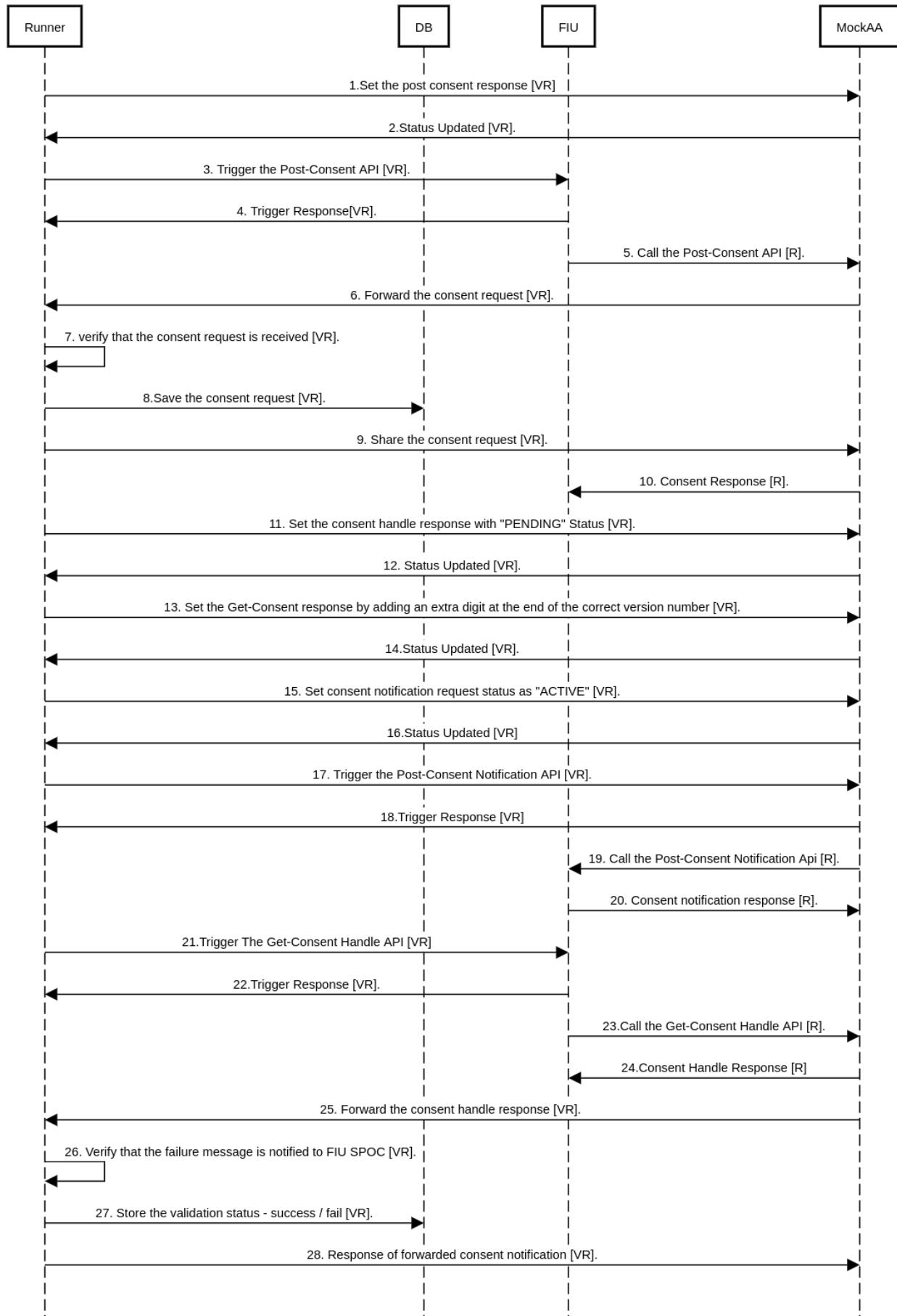
Runner<-MockAA:25. Forward the consent handle response [VR].

Runner ->Runner:26. Verify that the failure message is notified to FIU SPOC [VR].

Runner -> DB:27. Store the validation status – success / fail [VR].

Runner->MockAA:28. Response of forwarded consent notification [VR].

FIU_1031__GET _ConsentNotification.feature Scenario 1031_2



Scenario 1031_3

title FIU_1031__GET _ConsentNotification.feature Scenario 1031_3

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR]

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent response by removing the last character or digit from valid version number [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set consent notification request status as "ACTIVE" [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner->FIU:21. Trigger The Get-Consent Handle API [VR]

Runner<-FIU:22. Trigger Response [VR].

FIU->MockAA:23. Call the Get-Consent Handle API [R].

FIU<-MockAA:24. Consent Handle Response [R]

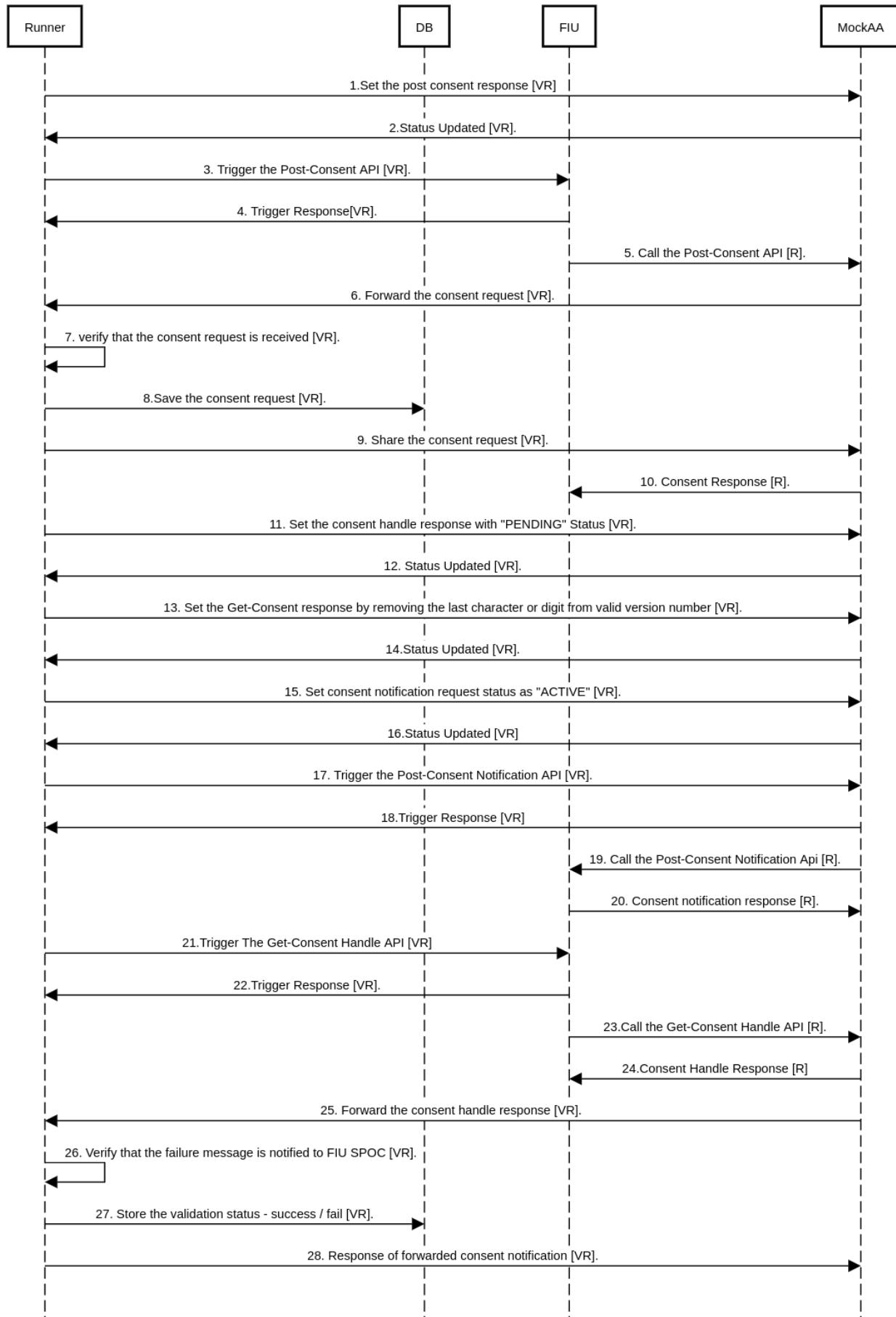
Runner<-MockAA:25. Forward the consent handle response [VR].

Runner ->Runner:26. Verify that the failure message is notified to FIU SPOC [VR].

Runner -> DB:27. Store the validation status – success / fail [VR].

Runner->MockAA:28. Response of forwarded consent notification [VR].

FIU_1031__GET _ConsentNotification.feature Scenario 1031_3



FIU_1032_POST_ConsentNotification.feature

Scenario 1032_1

title FIU_1032_POST_ConsentNotification.feature Scenario 1032_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR]

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent response with future timestamp(current time + 15 minutes) [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set consent notification request status as "ACTIVE" [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner->FIU:21.Trigger The Get-Consent Handle API [VR]

Runner<-FIU:22.Trigger Response [VR].

FIU->MockAA:23.Call the Get-Consent Handle API [R].

FIU<-MockAA:24.Consent Handle Response [R]

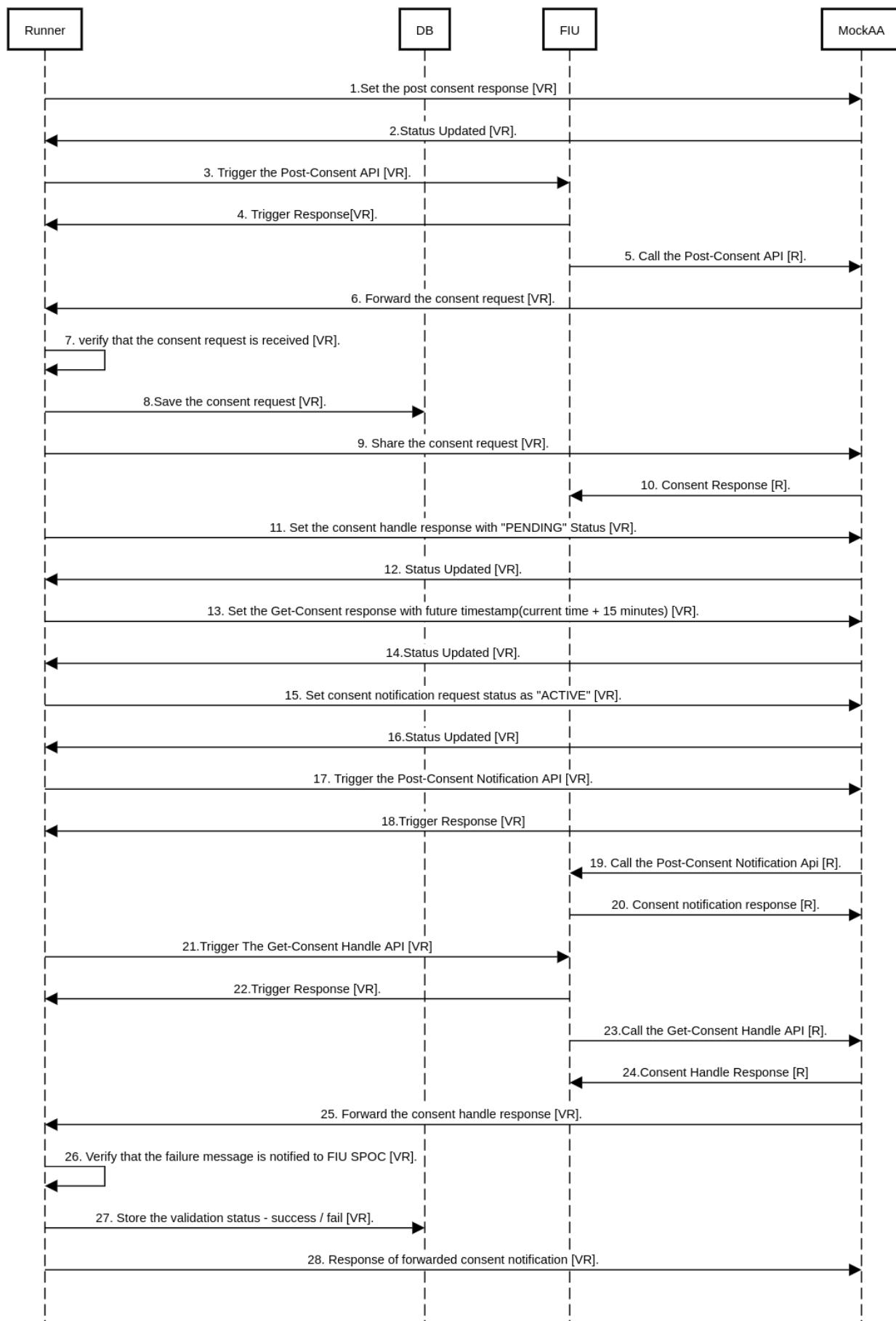
Runner<-MockAA:25. Forward the consent handle response [VR].

Runner ->Runner:26. Verify that the failure message is notified to FIU SPOC [VR].

Runner -> DB:27. Store the validation status - success / fail [VR].

Runner->MockAA:28. Response of forwarded consent notification [VR].

FIU_1032__POST_ConsentNotification.feature Scenario 1032_1



Scenario 1032_2

title FIU_1032__POST _ConsentNotification.feature Scenario 1032_2

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR]

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent response with expired timestamp(current time - 15 minutes) [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set consent notification request status as "ACTIVE" [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner->FIU:21. Trigger The Get-Consent Handle API [VR]

Runner<-FIU:22. Trigger Response [VR].

FIU->MockAA:23. Call the Get-Consent Handle API [R].

FIU<-MockAA:24. Consent Handle Response [R]

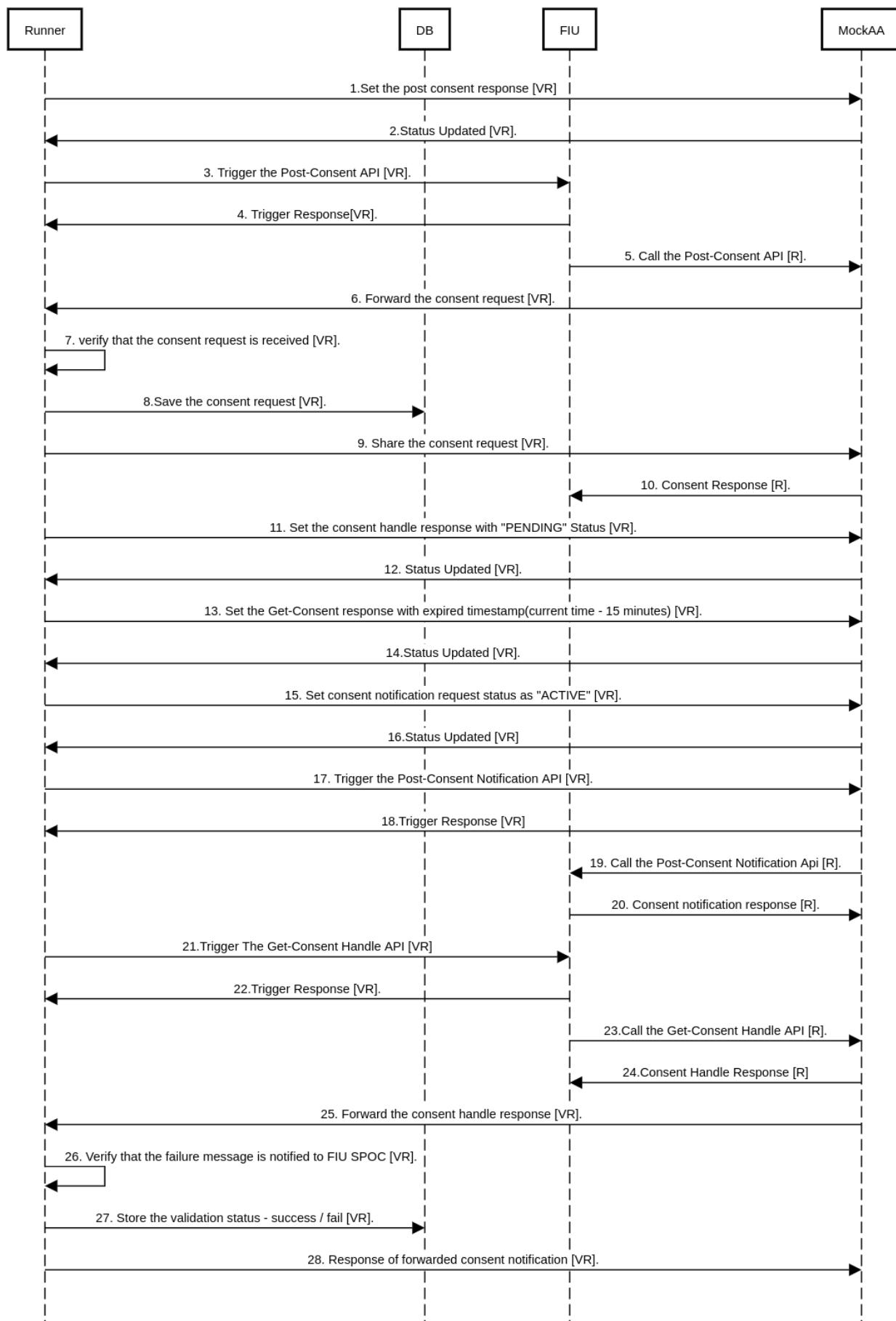
Runner<-MockAA:25. Forward the consent handle response [VR].

Runner ->Runner:26. Verify that the failure message is notified to FIU SPOC [VR].

Runner -> DB:27. Store the validation status – success / fail [VR].

Runner->MockAA:28. Response of forwarded consent notification [VR].

FIU_1032__POST_ConsentNotification.feature Scenario 1032_2



FIU_1033__GET _ConsentId.feature

Scenario 1033_1

title FIU_1033__GET _ConsentId.feature Scenario 1033_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR]

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent response by removing the first character from consent Id [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set consent notification request status as "ACTIVE" [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner->FIU:21.Trigger The Get-Consent API [VR]

Runner<-FIU:22.Trigger Response [VR].

FIU->MockAA:23.Call the Get-Consent API [R].

FIU<-MockAA:24.Consent Handle Response [R]

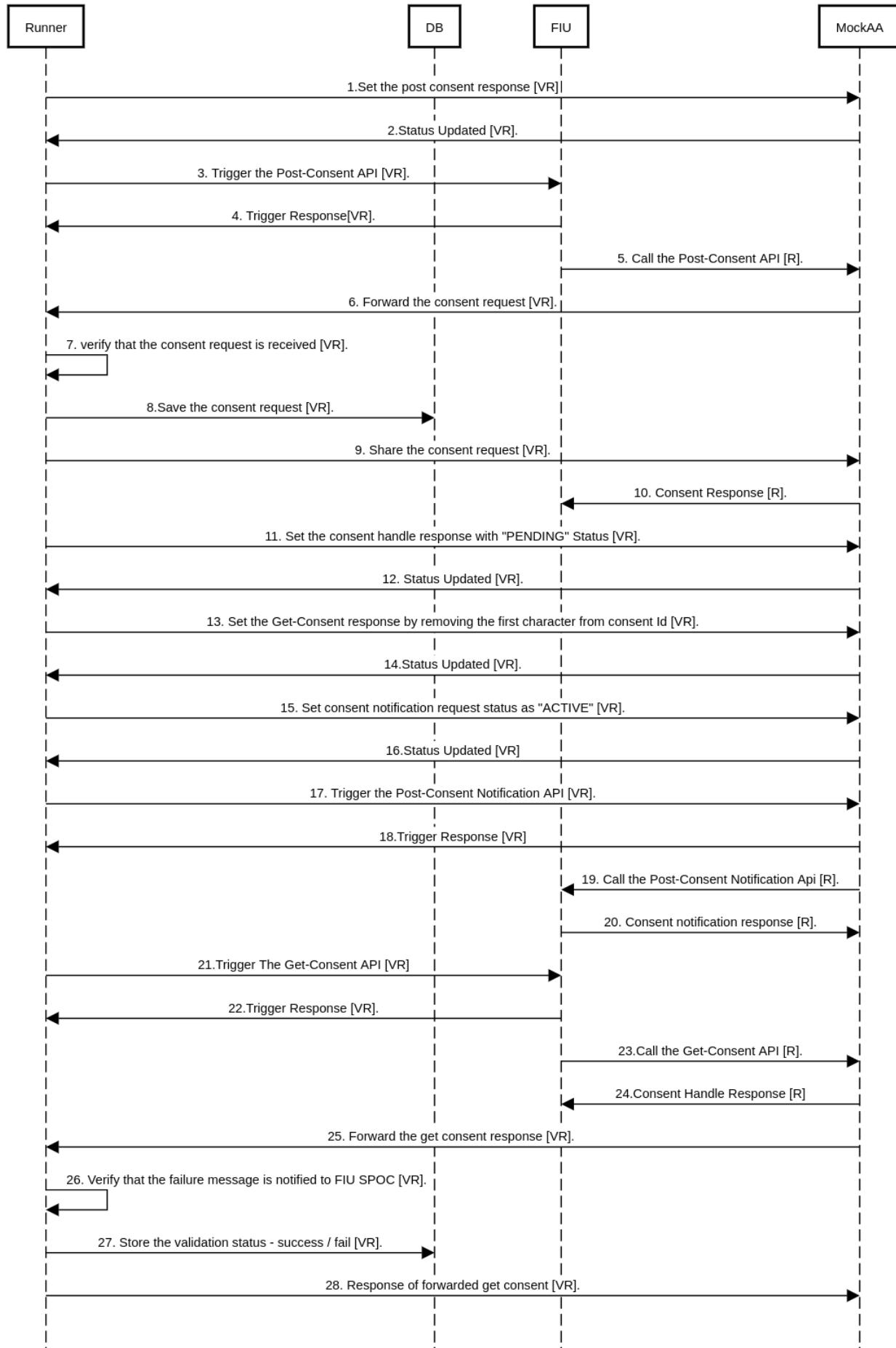
Runner<-MockAA:25. Forward the get consent response [VR].

Runner ->Runner:26. Verify that the failure message is notified to FIU SPOC [VR].

Runner -> DB:27. Store the validation status - success / fail [VR].

Runner->MockAA:28. Response of forwarded get consent [VR].

FIU_1033__GET_ConsentId.feature Scenario 1033_1



Scenario 1033_2

title FIU_1033__GET _ConsentId.feature Scenario 1033_2

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call
R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR]
Runner<-MockAA:2. Status Updated [VR].
Runner->FIU:3. Trigger the Post-Consent API [VR].
Runner<-FIU:4. Trigger Response[VR].
FIU->MockAA:5. Call the Post-Consent API [R].
Runner<-MockAA:6. Forward the consent request [VR].
Runner->Runner:7. verify that the consent request is received [VR].
Runner -> DB:8. Save the consent request [VR].
Runner->MockAA:9. Share the consent request [VR].
FIU<-MockAA:10. Consent Response [R].
Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].
Runner<-MockAA:12. Status Updated [VR].
Runner->MockAA:13. Set the Get-Consent response by removing the last character from consent Id [VR].
Runner<-MockAA:14. Status Updated [VR].
Runner->MockAA:15. Set consent notification request status as "ACTIVE" [VR].
Runner<-MockAA:16. Status Updated [VR]
Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].
Runner<-MockAA:18. Trigger Response [VR]
FIU<-MockAA:19. Call the Post-Consent Notification Api [R].
FIU->MockAA:20. Consent notification response [R].
Runner->FIU:21. Trigger The Get-Consent API [VR]

Runner<-FIU:22.Trigger Response [VR].

FIU->MockAA:23.Call the Get-Consent API [R].

FIU<-MockAA:24.Consent Handle Response [R]

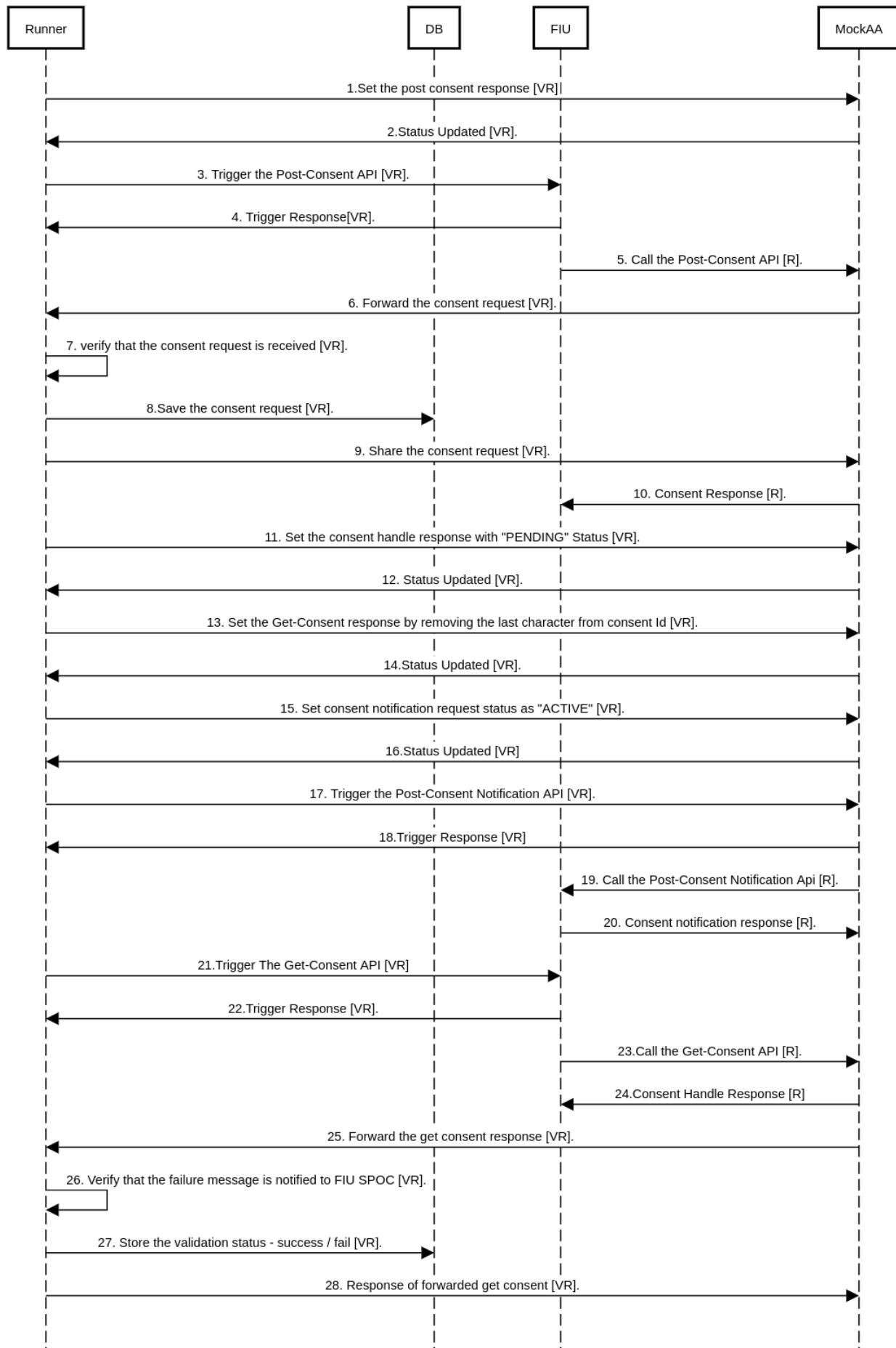
Runner<-MockAA:25. Forward the get consent response [VR].

Runner ->Runner:26. Verify that the failure message is notified to FIU SPOC [VR].

Runner -> DB:27. Store the validation status – success / fail [VR].

Runner->MockAA:28. Response of forwarded get consent [VR].

FIU_1033__GET_ConsentId.feature Scenario 1033_2



FIU_1034__GET __ConsentId.feature

Scenario 1034_1

title FIU_1034__GET __ConsentId.feature Scenario 1034_1

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR]

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent response with consent status as "PAUSED" [VR].

Runner<-MockAA:14.Status Updated [VR].

Runner->MockAA:15. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:16.Trigger Response [VR]

FIU<-MockAA:17. Call the Post-Consent Notification Api [R].

FIU->MockAA:18. Consent notification response [R].

Runner->FIU:19.Trigger The Get-Consent API [VR]

Runner<-FIU:20.Trigger Response [VR].

FIU->MockAA:22. Call the Get-Consent API [R].

FIU<-MockAA:22. Consent Handle Response [R]

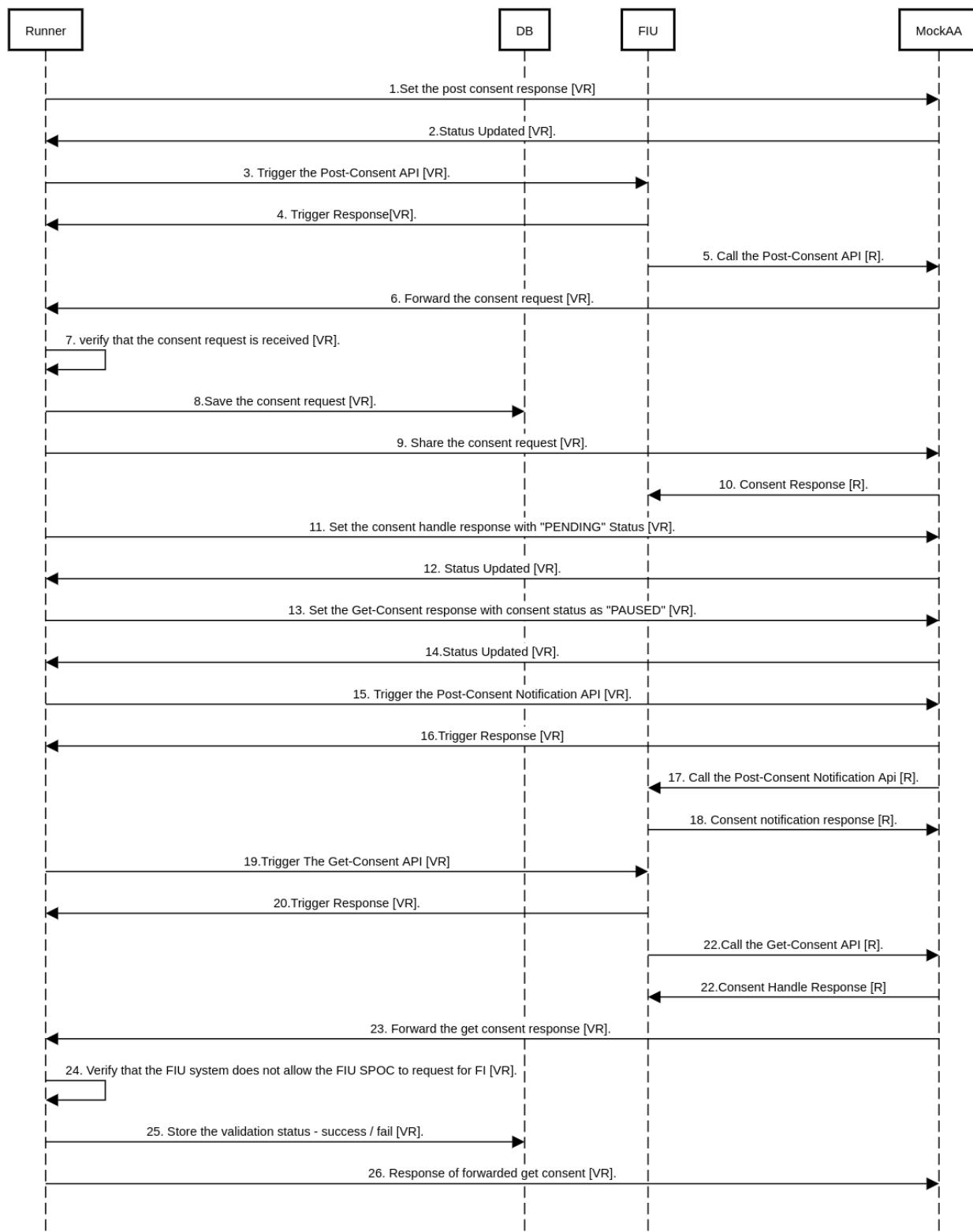
Runner<-MockAA:23. Forward the get consent response [VR].

Runner ->Runner:24. Verify that the FIU system does not allow the FIU SPOC to request for FI [VR].

Runner -> DB:25. Store the validation status – success / fail [VR].

Runner->MockAA:26. Response of forwarded get consent [VR].

FIU_1034__GET_ConsentId.feature Scenario 1034_1



FIU_1035__GET _ConsentId.feature:

Scenario 1035_1

title FIU_1035__GET _ConsentId.feature Scenario 1035_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR]

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent response with consent status as "REVOKE" [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-Consent Notification Api [R].

FIU->MockAA:18. Consent notification response [R].

Runner->FIU:19. Trigger The Get-Consent API [VR]

Runner<-FIU:20. Trigger Response [VR].

FIU->MockAA:22. Call the Get-Consent API [R].

FIU<-MockAA:22. Consent Handle Response [R]

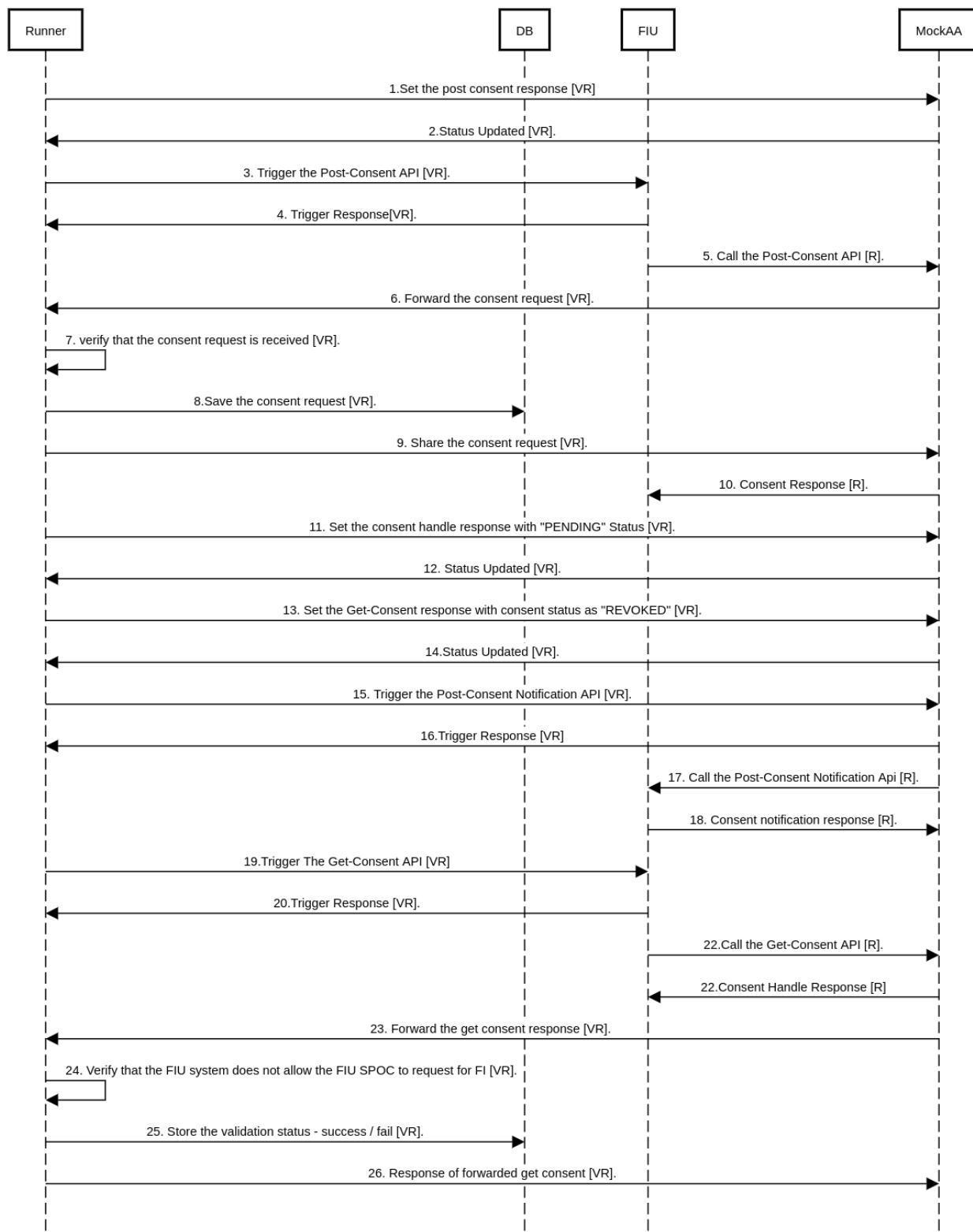
Runner<-MockAA:23. Forward the get consent response [VR].

Runner ->Runner:24. Verify that the FIU system does not allow the FIU SPOC to request for FI [VR].

Runner -> DB:25. Store the validation status - success / fail [VR].

Runner->MockAA:26. Response of forwarded get consent [VR].

FIU_1035__GET_ConsentId.feature Scenario 1035_1



FIU_1036_POST_ConsentNotification.feature

Scenario 1036_1

title FIU_1036_POST_ConsentNotification.feature Scenario 1036_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the Status as "Active", with invalid JWS signature [VR].

Runner<-MockAA:12. Status Updated [VR]

Runner->MockAA:13. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:14. Trigger Response [VR]

FIU<-MockAA:15. Call the Post-Consent Notification Api [R].

FIU->MockAA:16. Consent notification response [R].

Runner<-MockAA:17. Forward the consent notification response [VR].

Runner ->Runner:18. Verify that the Response code 400 is displayed [VR].

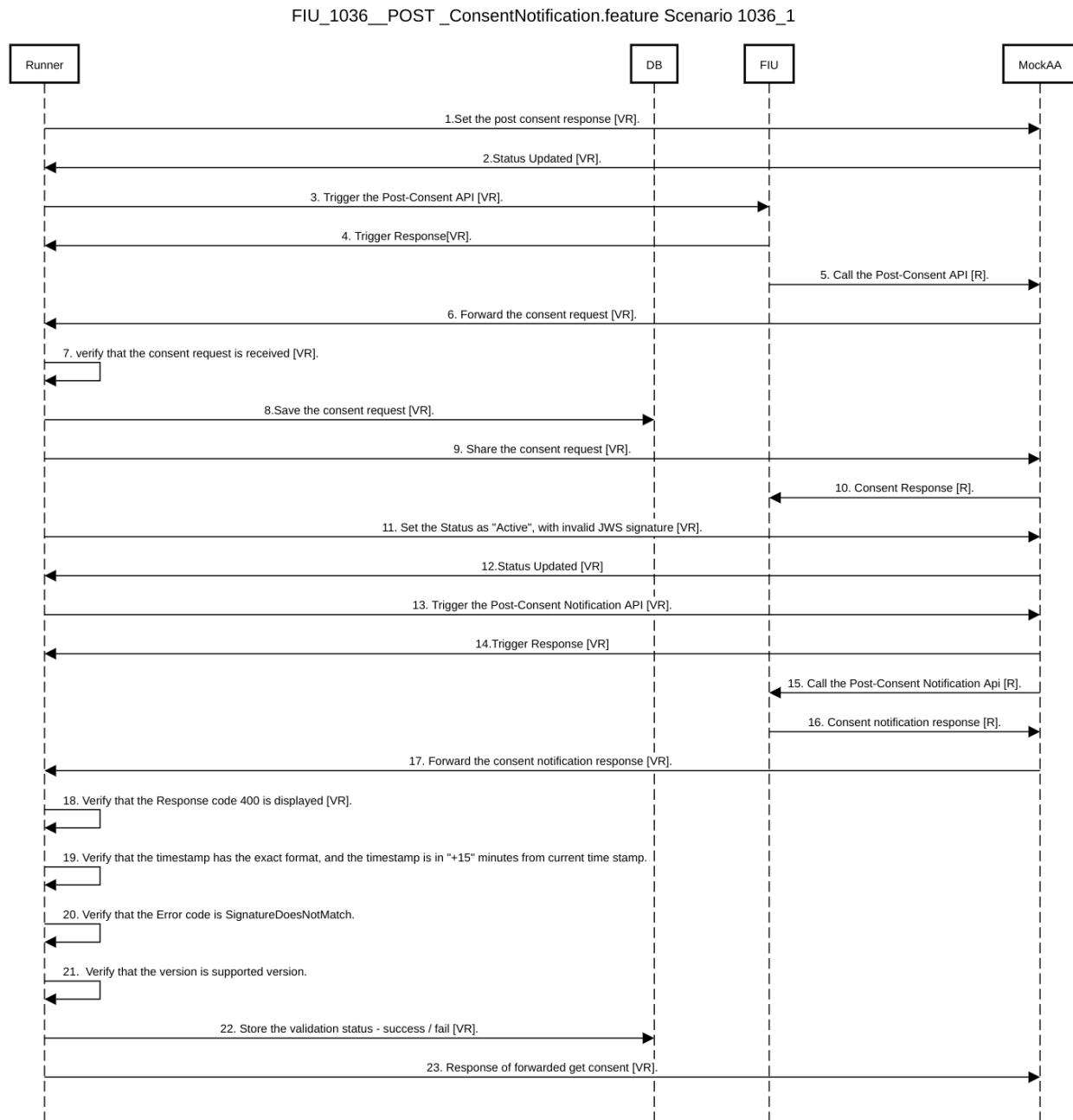
Runner->Runner:19. Verify that the timestamp has the exact format, and the timestamp is in "+15" minutes from current time stamp.

Runner->Runner:20. Verify that the Error code is SignatureDoesNotMatch.

Runner->Runner:21. Verify that the version is supported version.

Runner -> DB : 22. Store the validation status - success / fail [VR].

Runner->MockAA:23. Response of forwarded get consent [VR].



FIU_1037__GET _ConsentId.feature

Scenario 1037_1

title FIU_1037__GET _ConsentId.feature Scenario 1037_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR]

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent response by creating Timestamp with invalid date format [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set consent notification request status as "ACTIVE" [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner->FIU:21.Trigger The Get-Consent API [VR]

Runner<-FIU:22.Trigger Response [VR].

FIU->MockAA:23.Call the Get-Consent API [R].

FIU<-MockAA:24.Consent Handle Response [R]

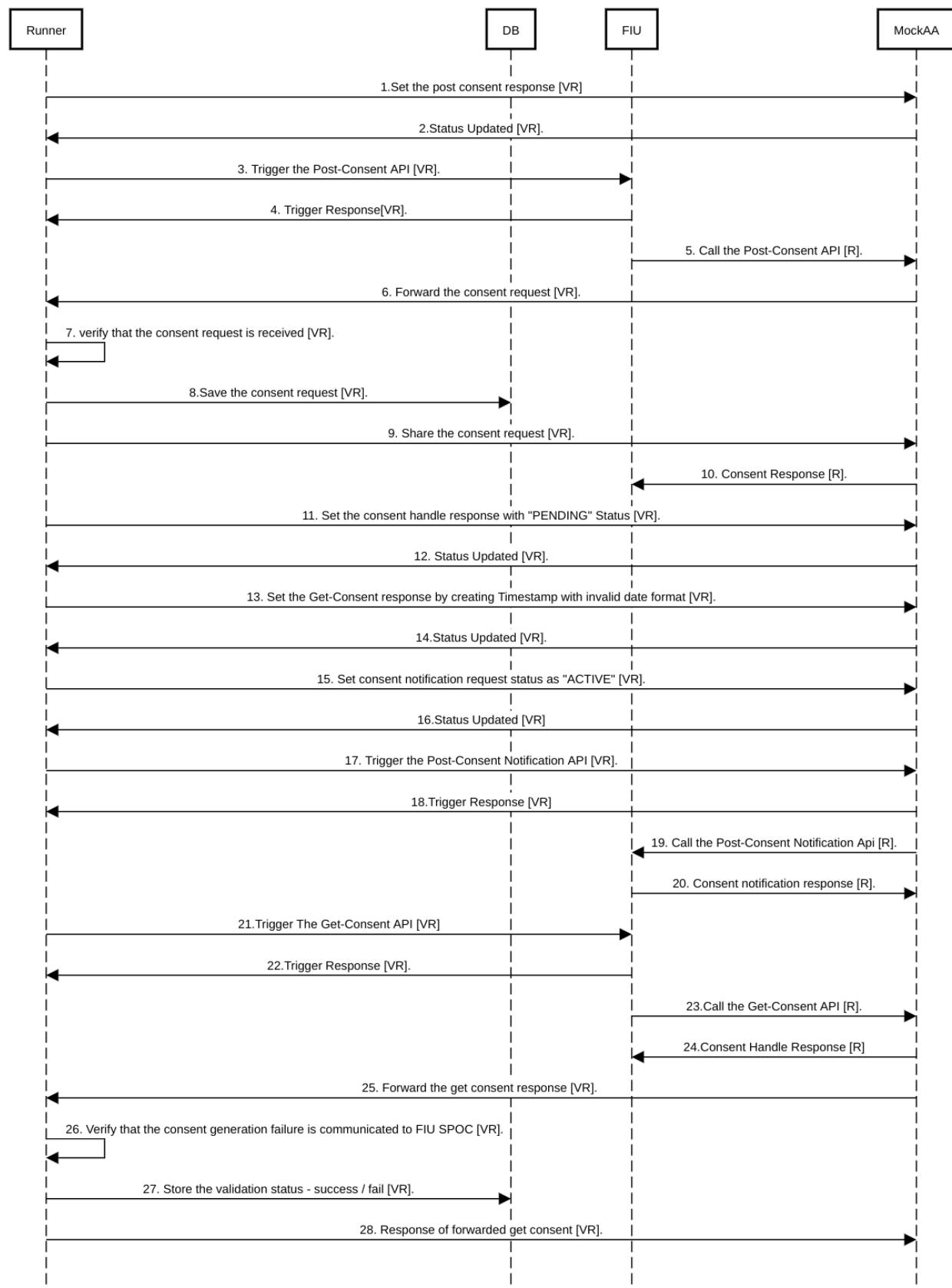
Runner<-MockAA:25. Forward the get consent response [VR].

Runner ->Runner:26. Verify that the consent generation failure is communicated to FIU SPOC [VR].

Runner -> DB:27. Store the validation status – success / fail [VR].

Runner->MockAA:28. Response of forwarded get consent [VR].

FIU_1037__GET_ConsentId.feature Scenario 1037_1



Scenario 1037_2

title FIU_1037__GET __ConsentId.feature Scenario 1037_2

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR]

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent response by creating Timestamp with +1 day variation [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set consent notification request status as "ACTIVE" [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner->FIU:21. Trigger The Get-Consent API [VR]

Runner<-FIU:22. Trigger Response [VR].

FIU->MockAA:23. Call the Get-Consent API [R].

FIU<-MockAA:24.Consent Handle Response [R]

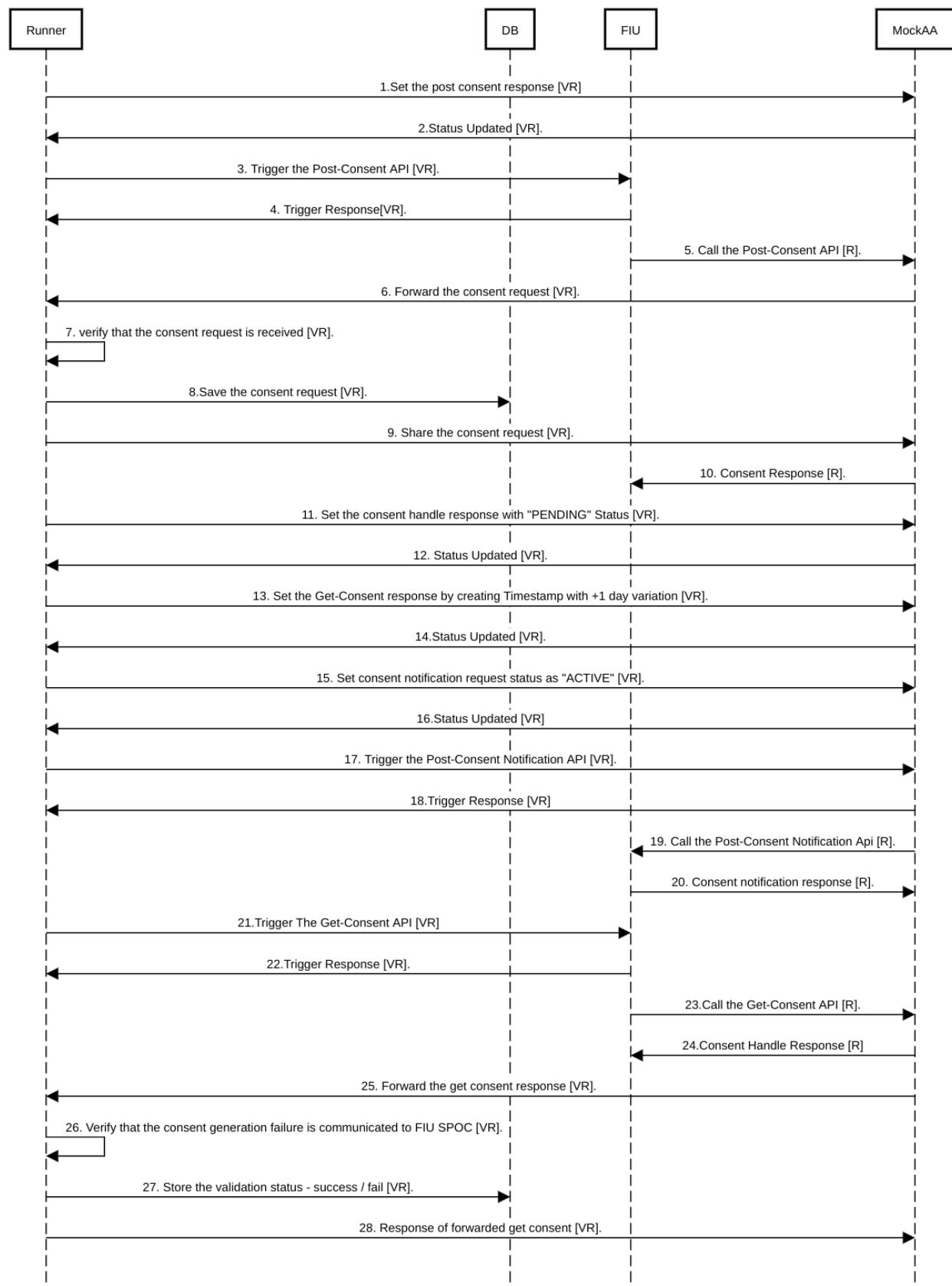
Runner<-MockAA:25. Forward the get consent response [VR].

Runner ->Runner:26. Verify that the consent generation failure is communicated to FIU SPOC [VR].

Runner -> DB:27. Store the validation status - success / fail [VR].

Runner->MockAA:28. Response of forwarded get consent [VR].

FIU_1037__GET_ConsentId.feature Scenario 1037_2



FIU_1038__GET__ConsentId.feature

Scenario 1038_1

title FIU_1038__GET__ConsentId.feature Scenario 1038_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR]

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent with unsigned consentDetails in signedconsent [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set consent notification request status as "ACTIVE" [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

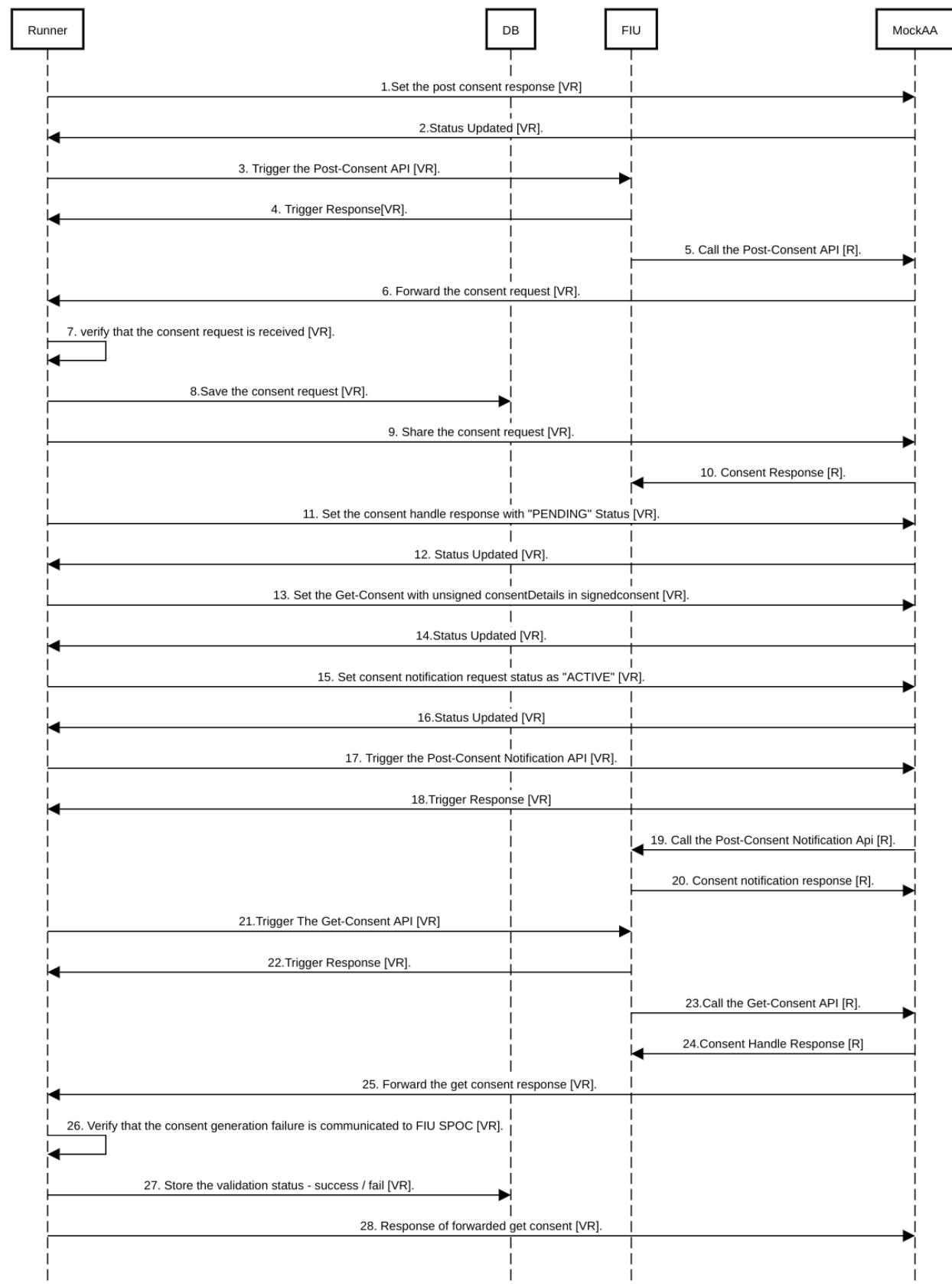
FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner->FIU:21. Trigger The Get-Consent API [VR]

Runner<-FIU:22.Trigger Response [VR].
FIU->MockAA:23.Call the Get-Consent API [R].
FIU<-MockAA:24.Consent Handle Response [R]
Runner<-MockAA:25. Forward the get consent response [VR].
Runner ->Runner:26. Verify that the consent generation failure is communicated to FIU SPOC [VR].
Runner -> DB:27. Store the validation status – success / fail [VR].
Runner->MockAA:28. Response of forwarded get consent [VR].

FIU_1038__GET_ConsentId.feature Scenario 1038_1



Scenario 1038_2

title FIU_1038__GET _ConsentId.feature Scenario 1038_2

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR]

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent by adding an extra key to the consent detail, and generate the signeConsent [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set consent notification request status as "ACTIVE" [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner->FIU:21. Trigger The Get-Consent API [VR]

Runner<-FIU:22. Trigger Response [VR].

FIU->MockAA:23. Call the Get-Consent API [R].

FIU<-MockAA:24.Consent Handle Response [R]

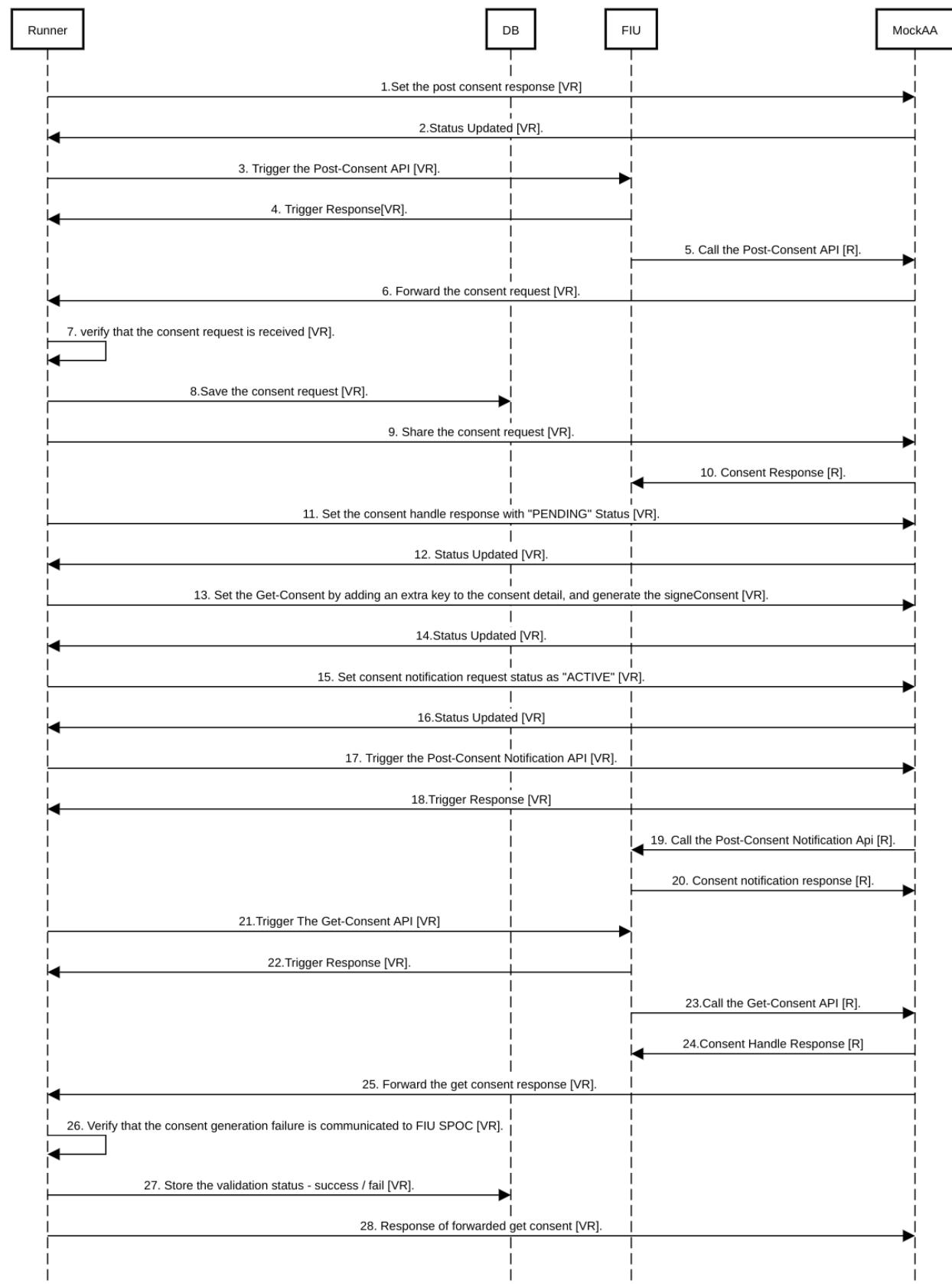
Runner<-MockAA:25. Forward the get consent response [VR].

Runner ->Runner:26. Verify that the consent generation failure is communicated to FIU SPOC [VR].

Runner -> DB:27. Store the validation status - success / fail [VR].

Runner->MockAA:28. Response of forwarded get consent [VR].

FIU_1038__GET_ConsentId.feature Scenario 1038_2



Scenario 1038_3

title FIU_1038__GET __ConsentId.feature Scenario 1038_3

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR]

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent by removing one mandatory field, and generate the signedconsent [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set consent notification request status as "ACTIVE" [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner->FIU:21. Trigger The Get-Consent API [VR]

Runner<-FIU:22. Trigger Response [VR].

FIU->MockAA:23. Call the Get-Consent API [R].

FIU<-MockAA:24.Consent Handle Response [R]

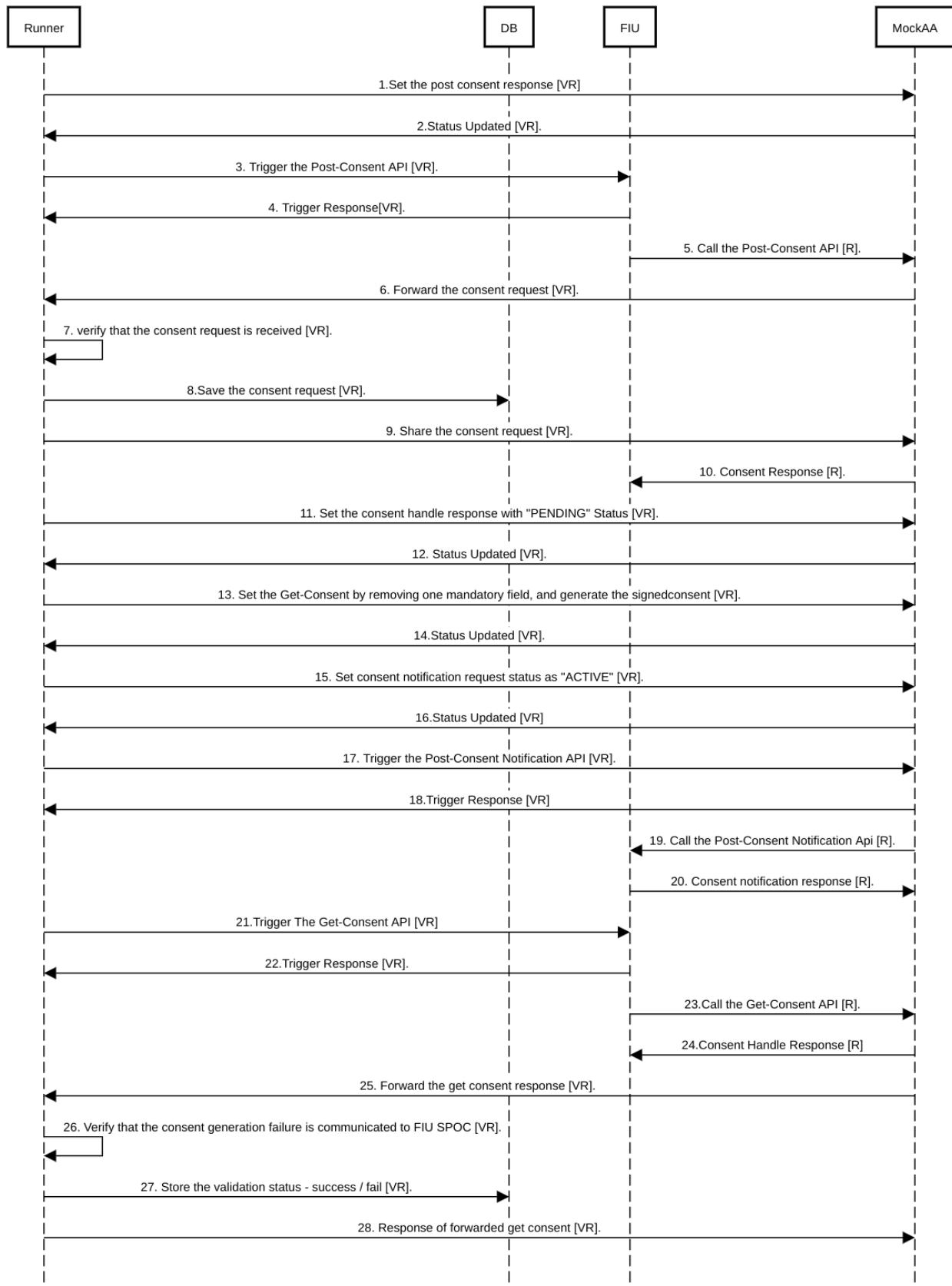
Runner<-MockAA:25. Forward the get consent response [VR].

Runner ->Runner:26. Verify that the consent generation failure is communicated to FIU SPOC [VR].

Runner -> DB:27. Store the validation status - success / fail [VR].

Runner->MockAA:28. Response of forwarded get consent [VR].

FIU_1038__GET__ConsentId.feature Scenario 1038_3



FIU_1039__GET__ConsentId.feature

Scenario 1039_1

title FIU_1039__GET__ConsentId.feature Set1 Scenario 1039_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR]

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent by setting the count as an String value [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set consent notification request status as "ACTIVE" [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

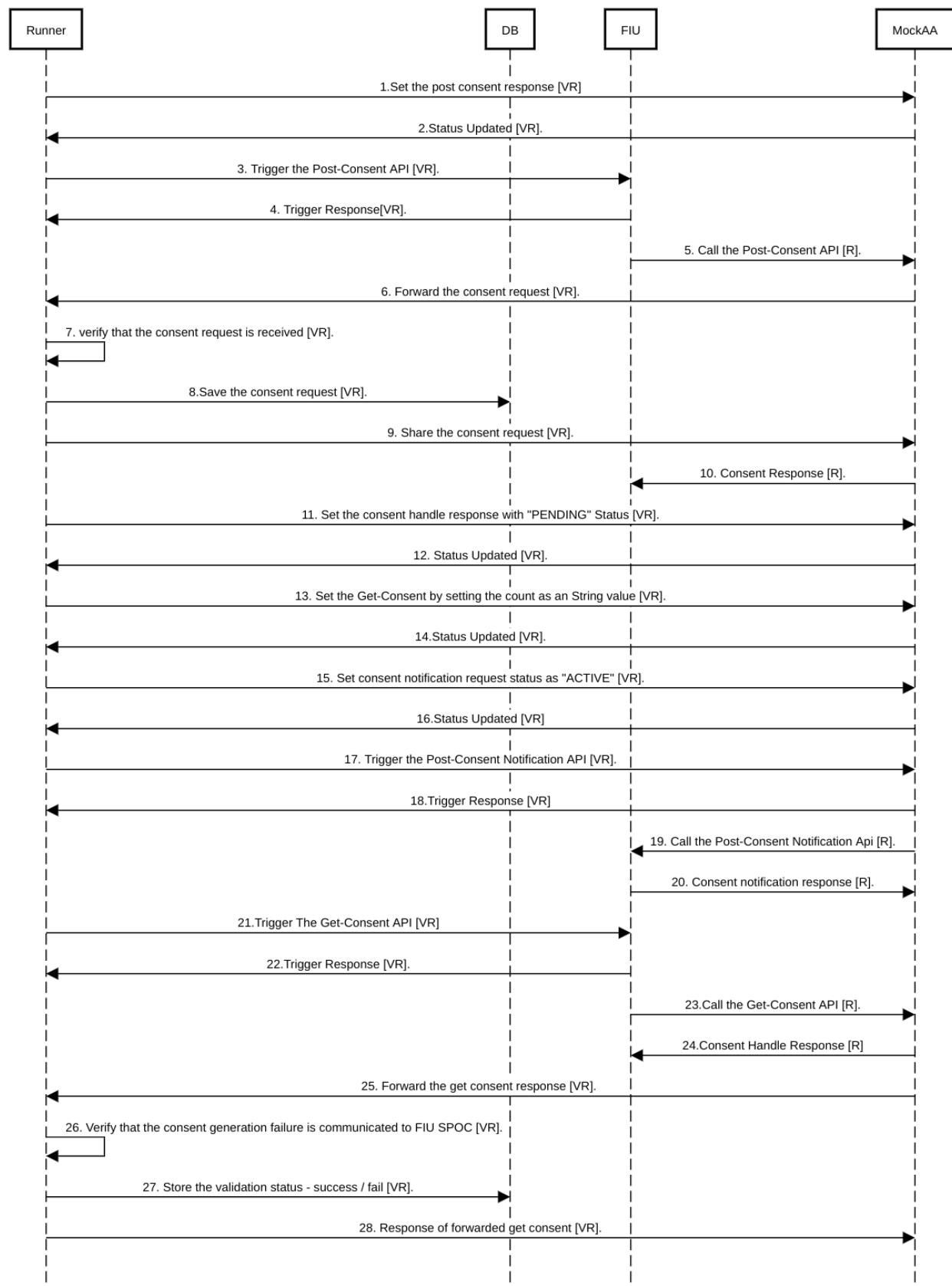
FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner->FIU:21. Trigger The Get-Consent API [VR]

Runner<-FIU:22.Trigger Response [VR].
FIU->MockAA:23.Call the Get-Consent API [R].
FIU<-MockAA:24.Consent Handle Response [R]
Runner<-MockAA:25. Forward the get consent response [VR].
Runner ->Runner:26. Verify that the consent generation failure is communicated to FIU SPOC [VR].
Runner -> DB:27. Store the validation status – success / fail [VR].
Runner->MockAA:28. Response of forwarded get consent [VR].

FIU_1039__GET_ConsentId.feature Set1 Scenario 1039_1



Scenario 1039_2

title FIU_1039__GET __ConsentId.feature Set1 Scenario 1039_2

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR]

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent by setting the count as alpha numerical value [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set consent notification request status as "ACTIVE" [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner->FIU:21. Trigger The Get-Consent API [VR]

Runner<-FIU:22. Trigger Response [VR].

FIU->MockAA:23. Call the Get-Consent API [R].

FIU<-MockAA:24.Consent Handle Response [R]

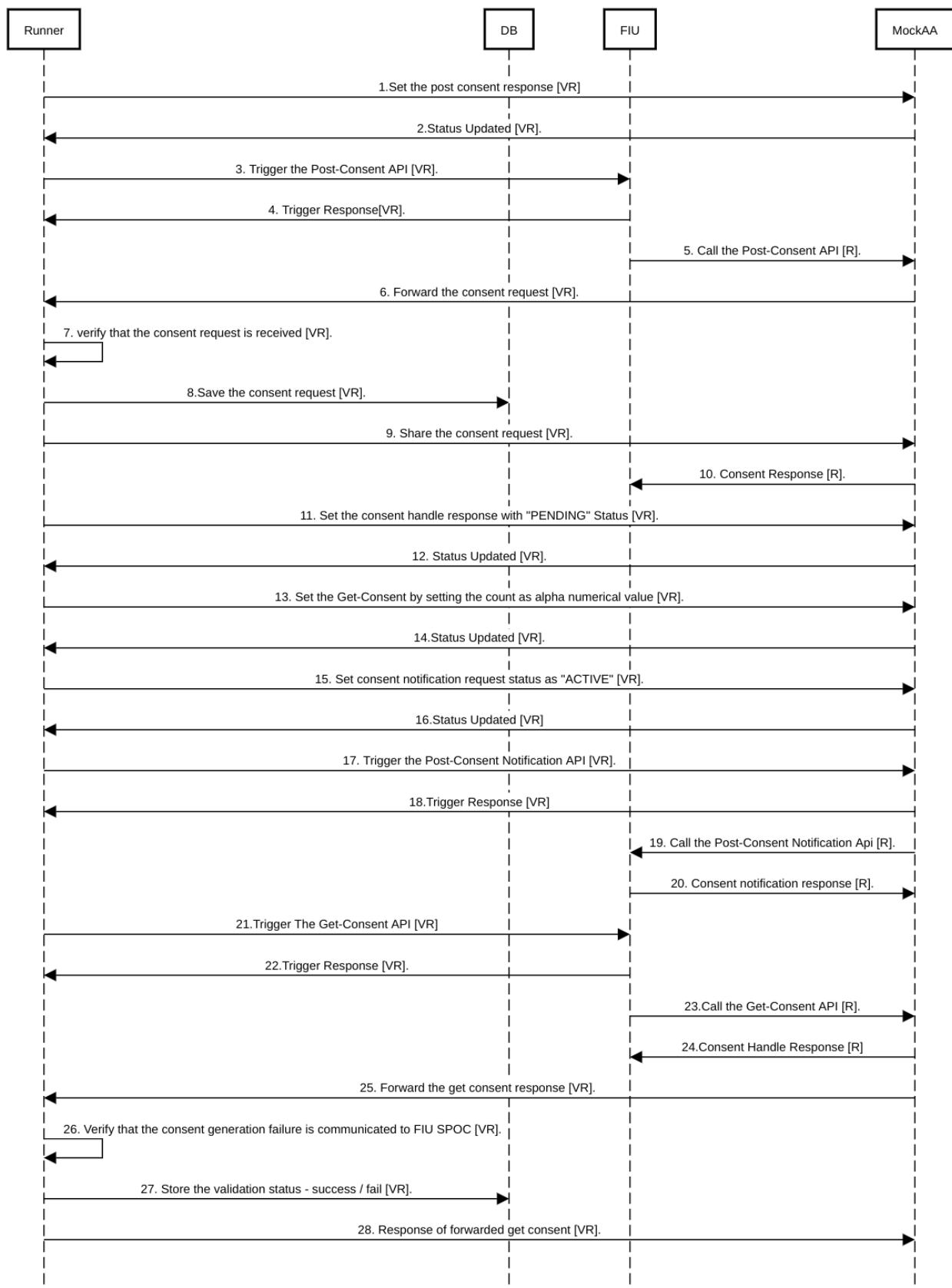
Runner<-MockAA:25. Forward the get consent response [VR].

Runner ->Runner:26. Verify that the consent generation failure is communicated to FIU SPOC [VR].

Runner -> DB:27. Store the validation status - success / fail [VR].

Runner->MockAA:28. Response of forwarded get consent [VR].

FIU_1039__GET_ConsentId.feature Set1 Scenario 1039_2



Scenario 1039_3

title FIU_1039__GET __ConsentId.feature Set1 Scenario 1039_3

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR]

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent by setting the lastUseDate Time date time with invalid date format [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set consent notification request status as "ACTIVE" [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner->FIU:21. Trigger The Get-Consent API [VR]

Runner<-FIU:22. Trigger Response [VR].

FIU->MockAA:23. Call the Get-Consent API [R].

FIU<-MockAA:24.Consent Handle Response [R]

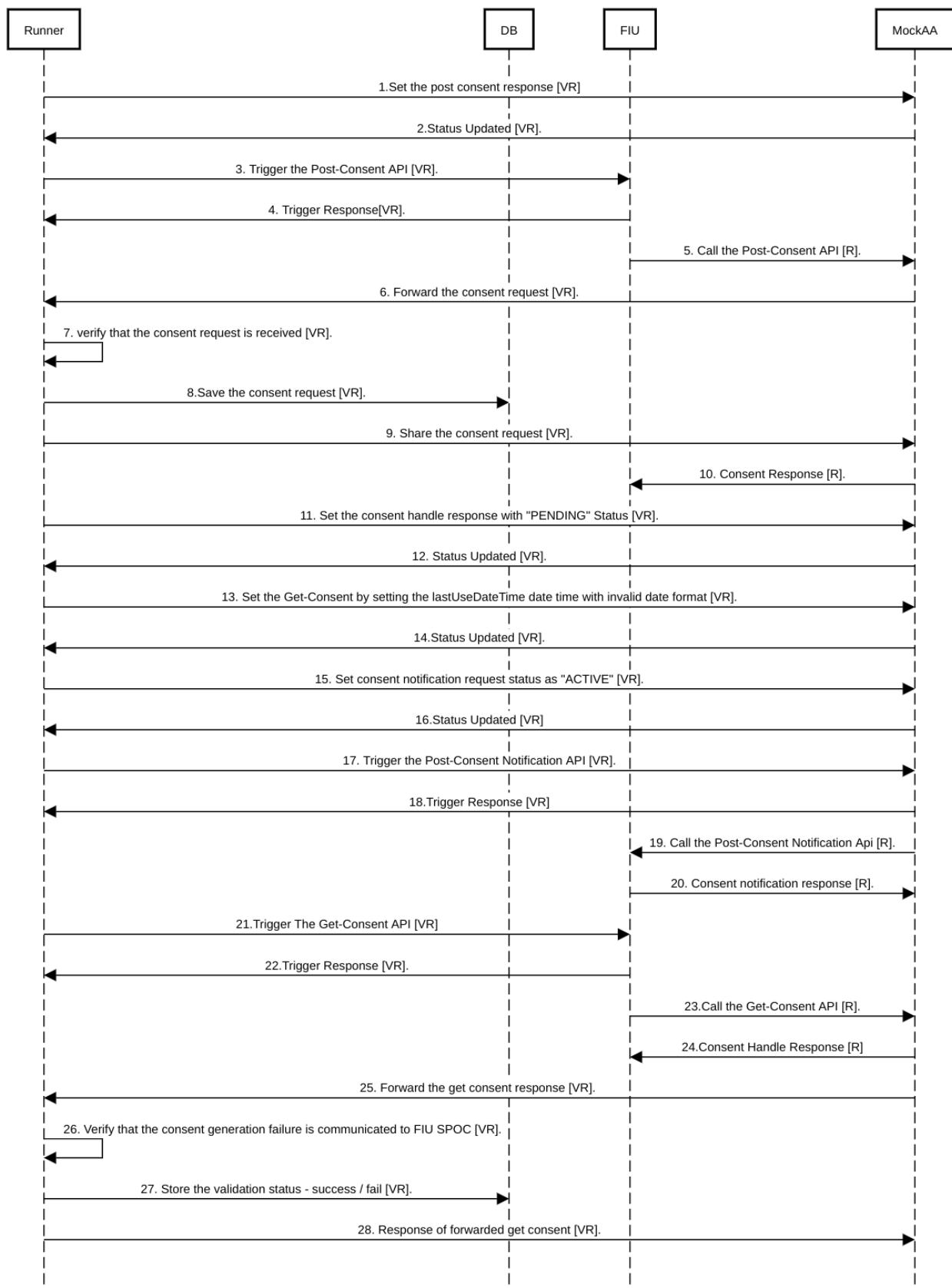
Runner<-MockAA:25. Forward the get consent response [VR].

Runner ->Runner:26. Verify that the consent generation failure is communicated to FIU SPOC [VR].

Runner -> DB:27. Store the validation status - success / fail [VR].

Runner->MockAA:28. Response of forwarded get consent [VR].

FIU_1039__GET_ConsentId.feature Set1 Scenario 1039_3



Scenario 1039_4

title FIU_1039__GET __ConsentId.feature Set1 Scenario 1039_4

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR]

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent by setting the lastUseDate Time greater than current time [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set consent notification request status as "ACTIVE" [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner->FIU:21. Trigger The Get-Consent API [VR]

Runner<-FIU:22. Trigger Response [VR].

FIU->MockAA:23. Call the Get-Consent API [R].

FIU<-MockAA:24.Consent Handle Response [R]

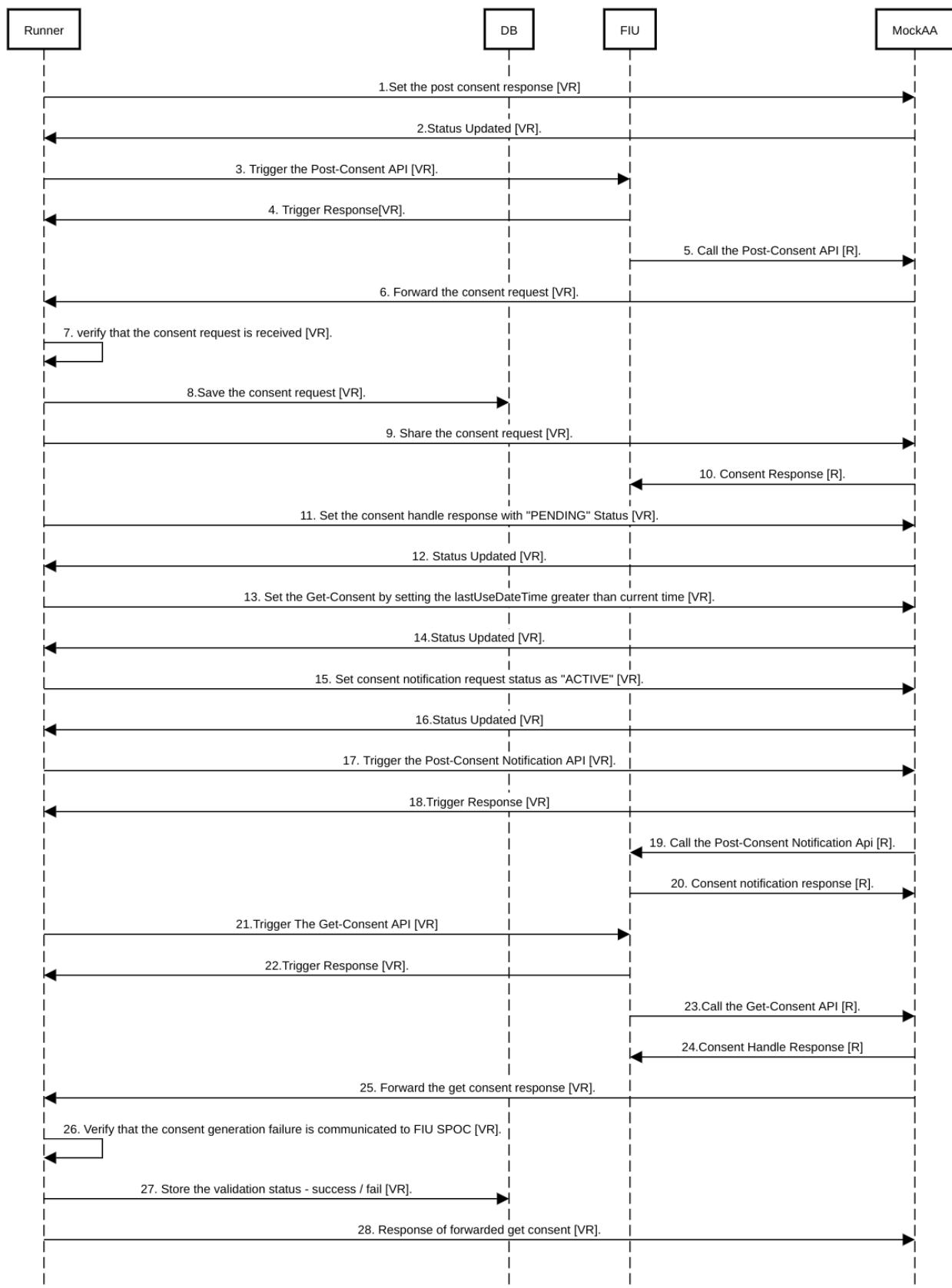
Runner<-MockAA:25. Forward the get consent response [VR].

Runner ->Runner:26. Verify that the consent generation failure is communicated to FIU SPOC [VR].

Runner -> DB:27. Store the validation status - success / fail [VR].

Runner->MockAA:28. Response of forwarded get consent [VR].

FIU_1039__GET__ConsentId.feature Set1 Scenario 1039_4



FIU_1040_POST_ConsentNotification.feature

Scenario 1040_1

title FIU_1040_POST_ConsentNotification.feature Scenario 1040_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8.Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the get consent response with signedConsent generated as alternate AA [VR].

Runner<-MockAA:12.Status Updated [VR].

Runner->MockAA:13. Set the consent notification request with "ACTIVE" Status [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:16.Trigger Response [VR]

FIU<-MockAA:17. Call the Post-Consent Notification Api [R].

FIU->MockAA:18. Consent notification response [R].

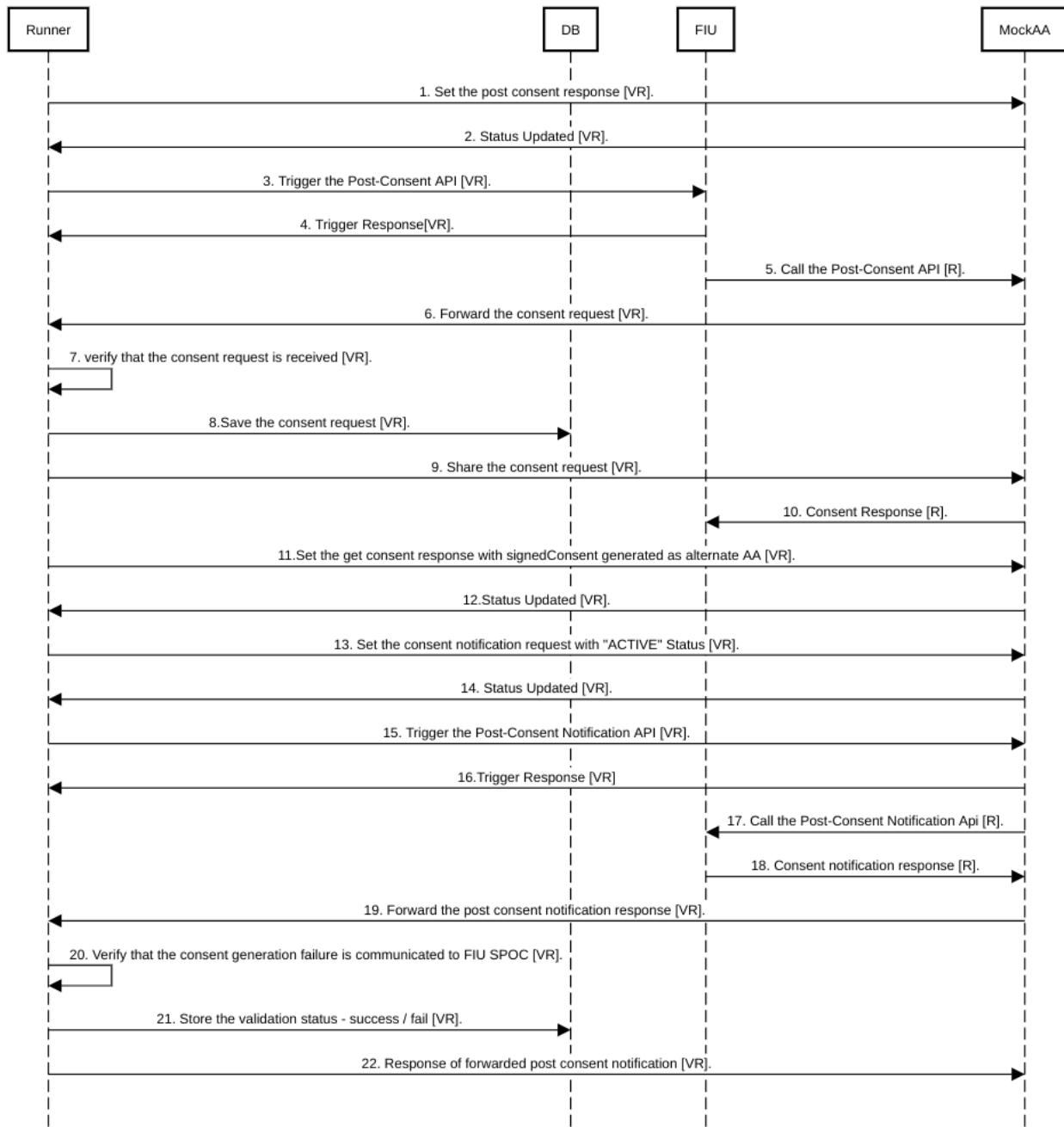
Runner<-MockAA:19. Forward the post consent notification response [VR].

Runner ->Runner:20. Verify that the consent generation failure is communicated to FIU SPOC [VR].

Runner -> DB : 21. Store the validation status - success / fail [VR].

Runner->MockAA:22. Response of forwarded post consent notification [VR].

FIU_1040__POST _ConsentNotification.feature Scenario 1040_1



FIU_1041__GET _ConsentId.feature

Scenario 1041_1

title FIU_1041__GET _ConsentId.feature Scenario 1041_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR]

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the Get-Consent with invalid JWS signature on mock server [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Set consent notification request status as "ACTIVE" [VR].

Runner<-MockAA:16. Status Updated [VR]

Runner->MockAA:17. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:18. Trigger Response [VR]

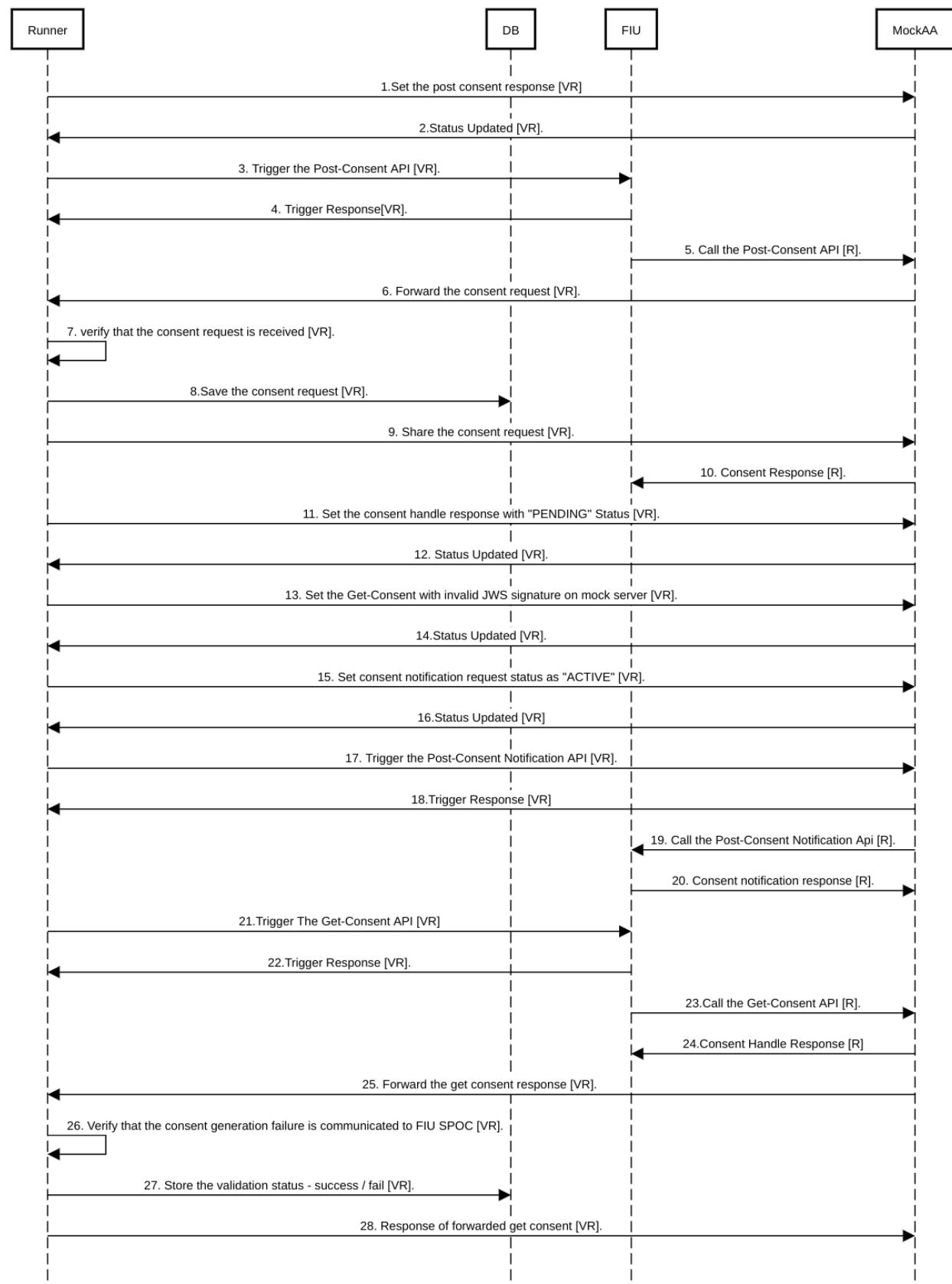
FIU<-MockAA:19. Call the Post-Consent Notification Api [R].

FIU->MockAA:20. Consent notification response [R].

Runner->FIU:21. Trigger The Get-Consent API [VR]

Runner<-FIU:22.Trigger Response [VR].
FIU->MockAA:23.Call the Get-Consent API [R].
FIU<-MockAA:24.Consent Handle Response [R]
Runner<-MockAA:25. Forward the get consent response [VR].
Runner ->Runner:26. Verify that the consent generation failure is communicated to FIU SPOC [VR].
Runner -> DB:27. Store the validation status – success / fail [VR].
Runner->MockAA:28. Response of forwarded get consent [VR].

FIU_1041__GET__ConsentId.feature Scenario 1041_1



FIU_1042_POST_ConsentNotification.feature

Scenario 1042_1

title FIU_1042_POST_ConsentNotification.feature Scenario 1042_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8.Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the Status as "Active", and without the aa_api_key header [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:14.Trigger Response [VR]

FIU<-MockAA:15. Call the Post-Consent Notification Api [R].

FIU->MockAA:16. Consent notification response [R].

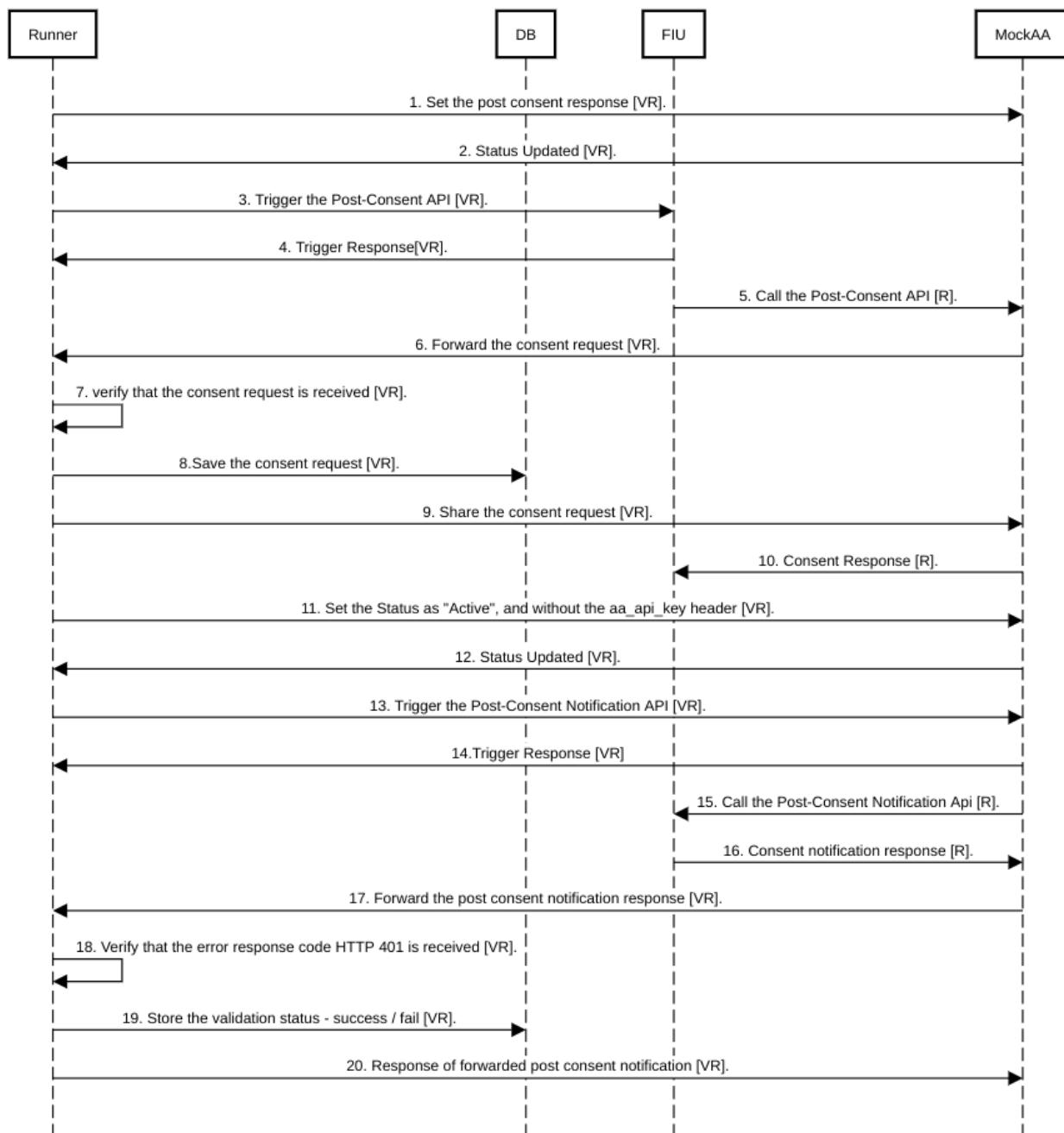
Runner<-MockAA:17. Forward the post consent notification response [VR].

Runner ->Runner:18. Verify that the error response code HTTP 401 is received [VR].

Runner -> DB : 19. Store the validation status - success / fail [VR].

Runner->MockAA:20. Response of forwarded post consent notification [VR].

FIU_1042__POST _ConsentNotification.feature Scenario 1042_1



Scenario 1042_2

title FIU_1042__POST _ConsentNotification.feature Scenario 1042_2

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the Status as "Active", and with aa_api_key as empty string [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:14. Trigger Response [VR]

FIU<-MockAA:15. Call the Post-Consent Notification Api [R].

FIU->MockAA:16. Consent notification response [R].

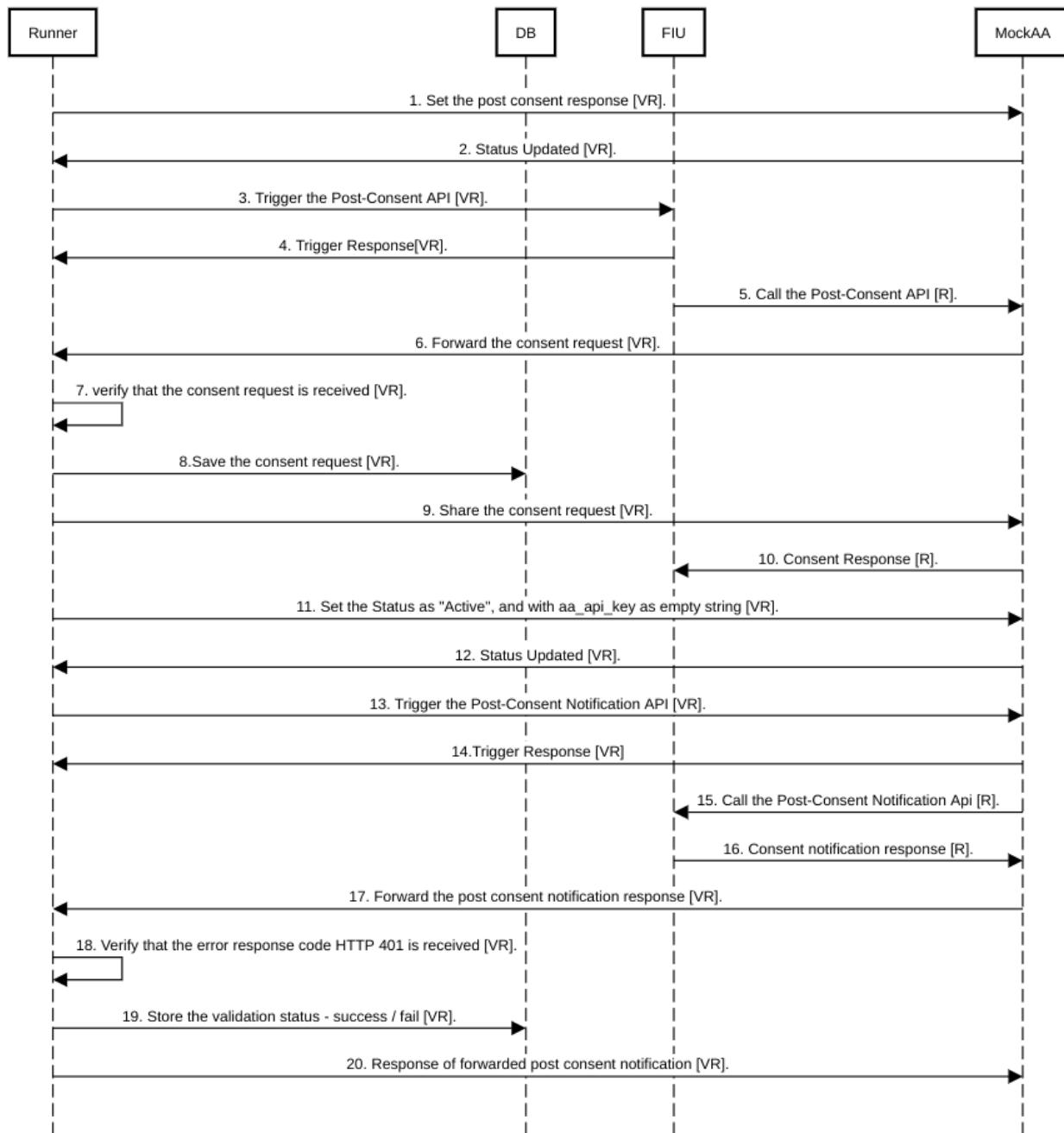
Runner<-MockAA:17. Forward the post consent notification response [VR].

Runner ->Runner:18. Verify that the error response code HTTP 401 is received [VR].

Runner -> DB : 19. Store the validation status - success / fail [VR].

Runner->MockAA:20. Response of forwarded post consent notification [VR].

FIU_1042__POST _ConsentNotification.feature Scenario 1042_2



Scenario 1042_3

title FIU_1042_POST_ConsentNotification.feature Scenario 1042_3

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the Status as "Active", and with aa_api_key as 0(zero) [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:14. Trigger Response [VR]

FIU<-MockAA:15. Call the Post-Consent Notification Api [R].

FIU->MockAA:16. Consent notification response [R].

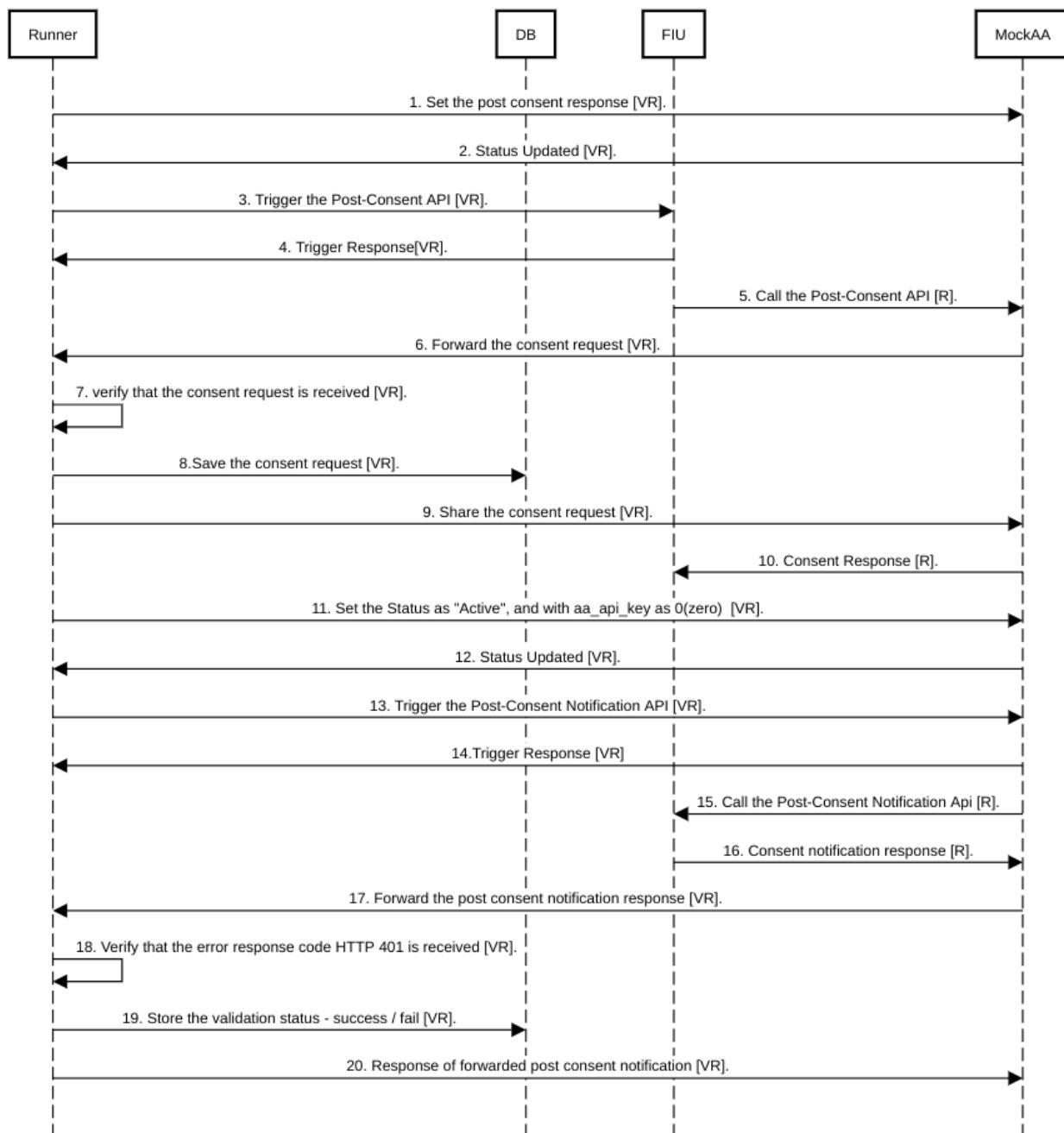
Runner<-MockAA:17. Forward the post consent notification response [VR].

Runner ->Runner:18. Verify that the error response code HTTP 401 is received [VR].

Runner -> DB : 19. Store the validation status - success / fail [VR].

Runner->MockAA:20. Response of forwarded post consent notification [VR].

FIU_1042__POST _ConsentNotification.feature Scenario 1042_3



Scenario 1042_4.1

title FIU_1042_POST_ConsentNotification.feature Scenario 1042_4.1

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the Status as "Active", split the api key with .(dot), and send the request with one char removed for that component [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:14. Trigger Response [VR]

FIU<-MockAA:15. Call the Post-Consent Notification Api [R].

FIU->MockAA:16. Consent notification response [R].

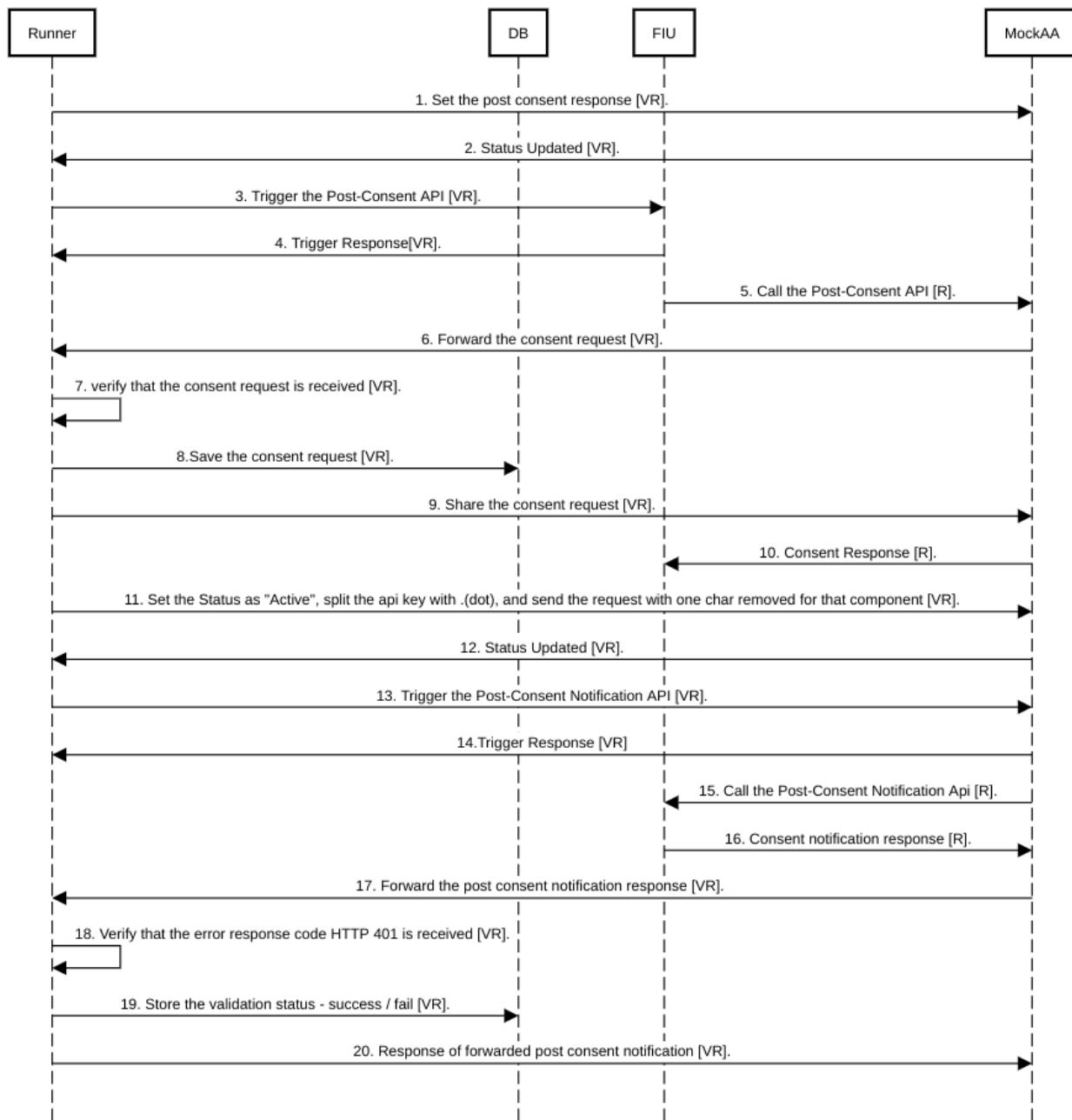
Runner<-MockAA:17. Forward the post consent notification response [VR].

Runner ->Runner:18. Verify that the error response code HTTP 401 is received [VR].

Runner -> DB : 19. Store the validation status - success / fail [VR].

Runner->MockAA:20. Response of forwarded post consent notification [VR].

FIU_1042__POST _ConsentNotification.feature Scenario 1042_4.1



Scenario 1042_4.2

title FIU_1042__POST _ConsentNotification.feature Scenario 1042_4.2

participant Runner
participant DB
participant FIU
participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8.Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the Status as "Active", split the api key with .(dot), and send the request with one character added for that component [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:14.Trigger Response [VR]

FIU<-MockAA:15. Call the Post-Consent Notification Api [R].

FIU->MockAA:16. Consent notification response [R].

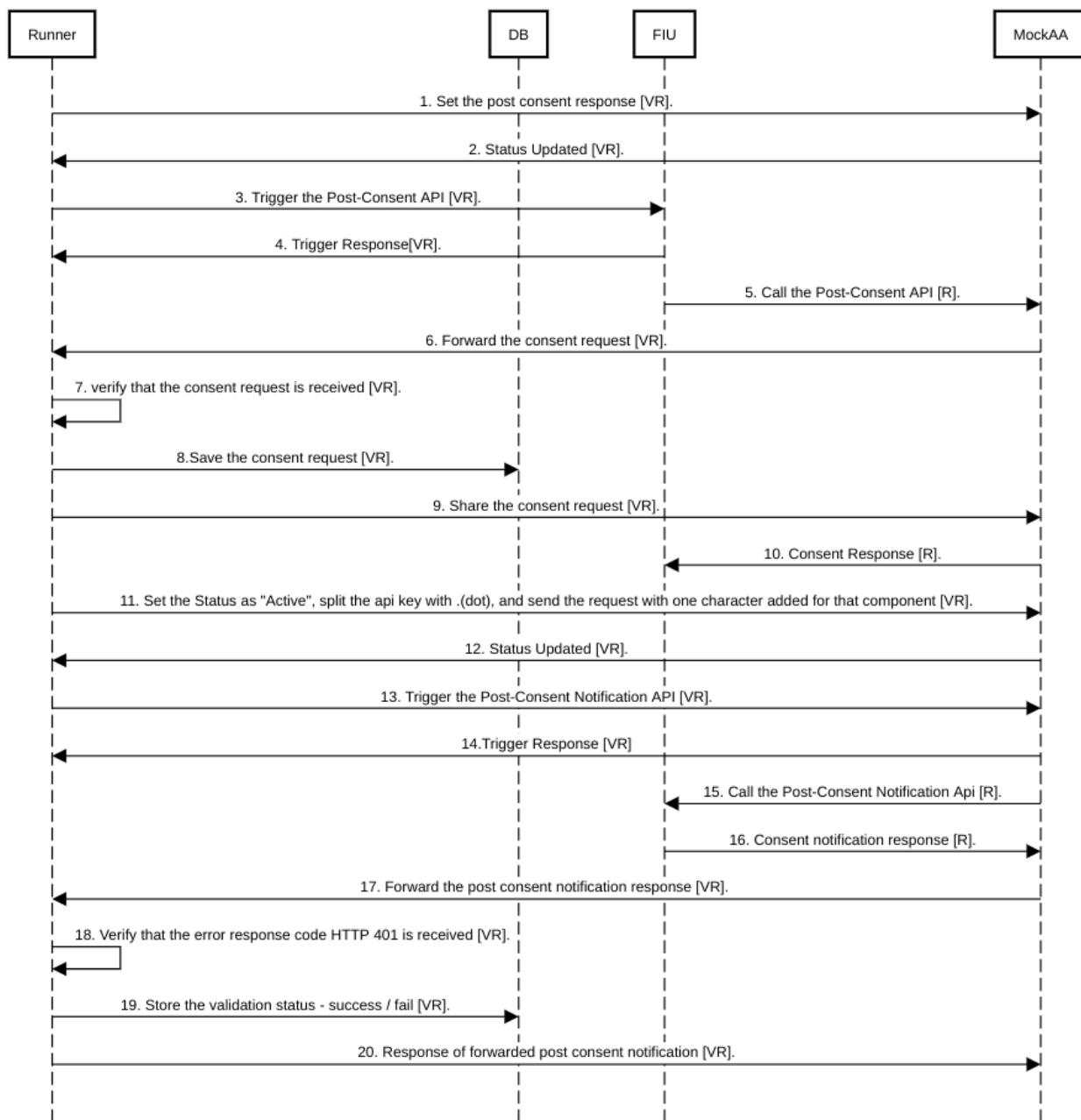
Runner<-MockAA:17. Forward the post consent notification response [VR].

Runner ->Runner:18. Verify that the error response code HTTP 401 is received [VR].

Runner -> DB : 19. Store the validation status - success / fail [VR].

Runner->MockAA:20. Response of forwarded post consent notification [VR].

FIU_1042__POST__ConsentNotification.feature Scenario 1042_4.2



FIU_1043_POST_ConsentNotification.feature

Scenario 1043_1

title FIU_1043_POST_ConsentNotification.feature Scenario 1043_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8.Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the Status as "Active" with Alternate AA api key [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:14.Trigger Response [VR]

FIU<-MockAA:15. Call the Post-Consent Notification Api [R].

FIU->MockAA:16. Consent notification response [R].

Runner<-MockAA:17. Forward the post consent notification response [VR].

Runner ->Runner:18. Verify that the Response code 400 is displayed [VR].

Runner->Runner:19.Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

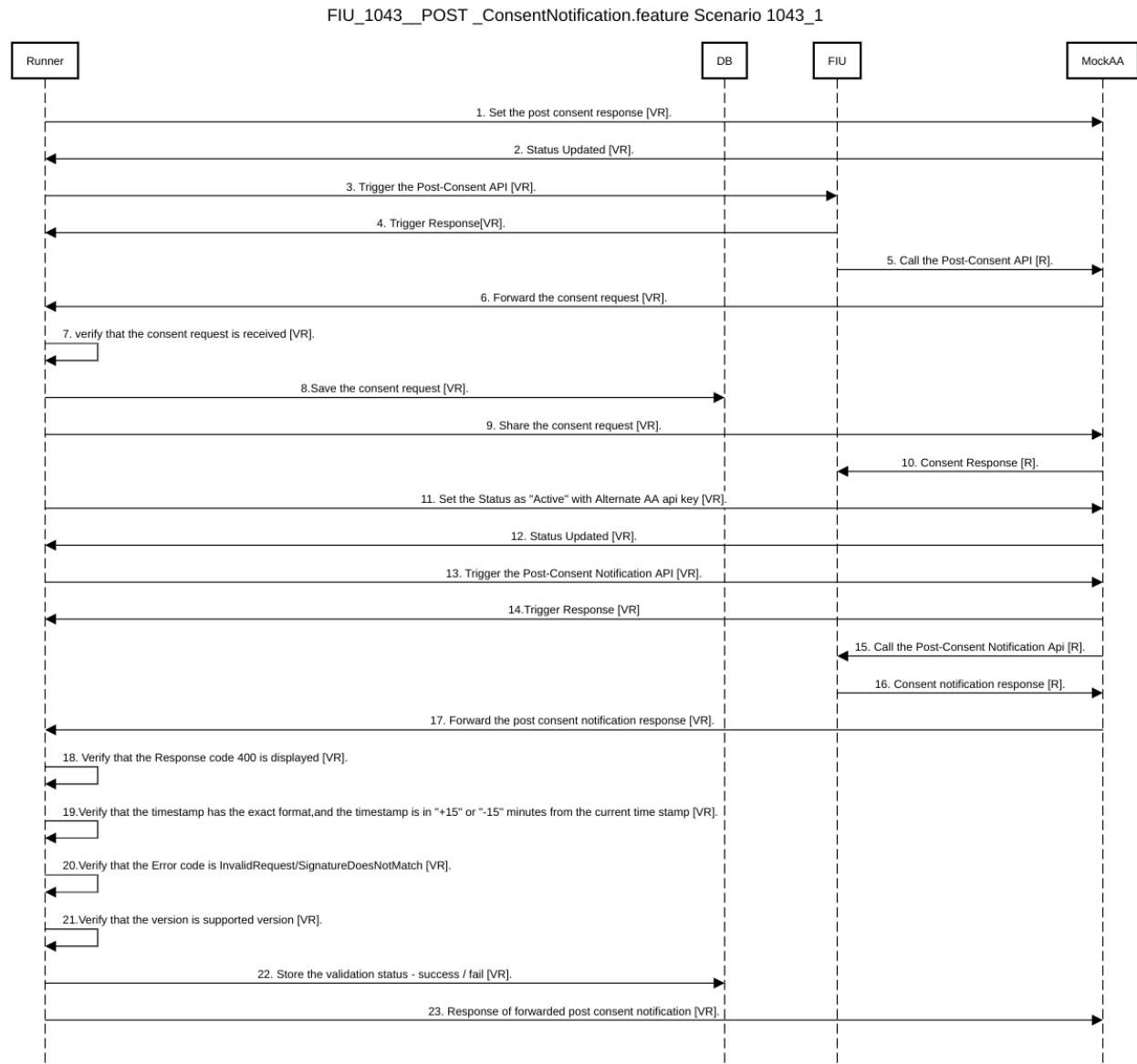
Runner->Runner:20.Verify that the Error code is

InvalidRequest/SignatureDoesNotMatch [VR].

Runner->Runner:21.Verify that the version is supported version [VR].

Runner -> DB : 22. Store the validation status - success / fail [VR].

Runner->MockAA:23. Response of forwarded post consent notification [VR].



FIU_2001_POST_FIRequest.feature

Scenario 2001_1

title FIU_2001_POST_FIRequest.feature Scenario 2001_1

participant Runner
participant DB
participant FIU
participant MockAA

Runner->FIU:1. Trigger the Post-FI request [VR].

Runner<-FIU:2. Trigger Response [VR].

FIU->MockAA:3. Call the Post-FI request [R].

MockAA ->Runner:4. Forward the FI request [VR].

Runner->Runner:5. Verify that the field of Version contains the version of NBFC-MockAA Ecosystem API that currently supported [VR].

Runner->Runner:6. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from current time stamp [VR].

Runner->Runner:7. Verify that the FROM FI Date Range is smaller than TO FI Date Range and both the date formats are in correct date format [VR].

Runner->Runner:8. Verify that the consent id is the valid consent id [VR].

Runner->Runner:9. Verify that the digitalSignature is the Valid signature [VR].

Runner->Runner:10. Verify that the Key Material is not expired [VR].

Runner->Runner:11. Verify that the Key material curve and Algorithm is valid [VR].

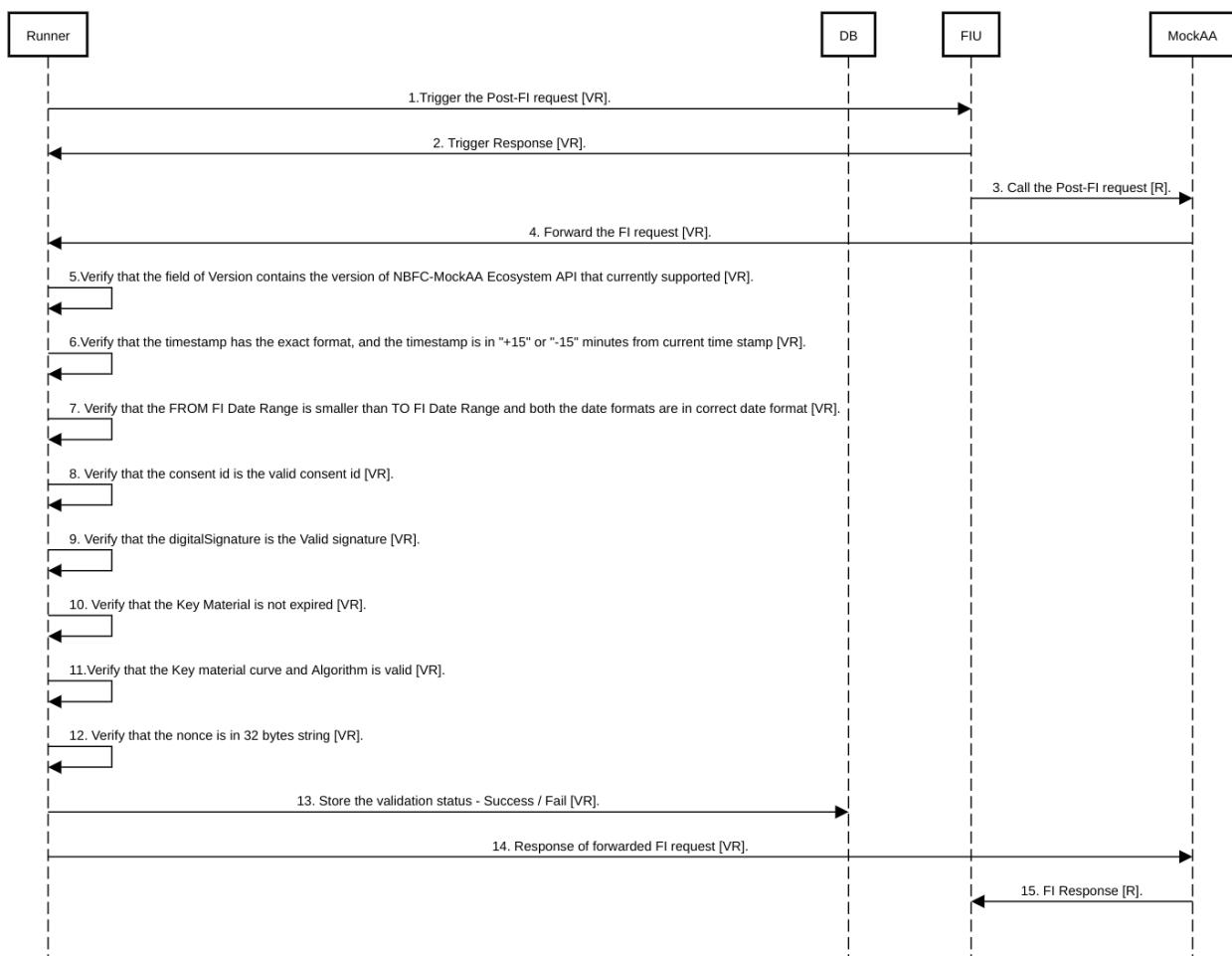
Runner->Runner:12. Verify that the nonce is in 32 bytes string [VR].

Runner -> DB: 13. Store the validation status - Success / Fail [VR].

Runner->MockAA:14. Response of forwarded FI request [VR].

FIU<-MockAA:15. FI Response [R].

FIU_2001_POST _FIRequest.feature Scenario 2001_1



FIU_2002_POST_FINotification.feature

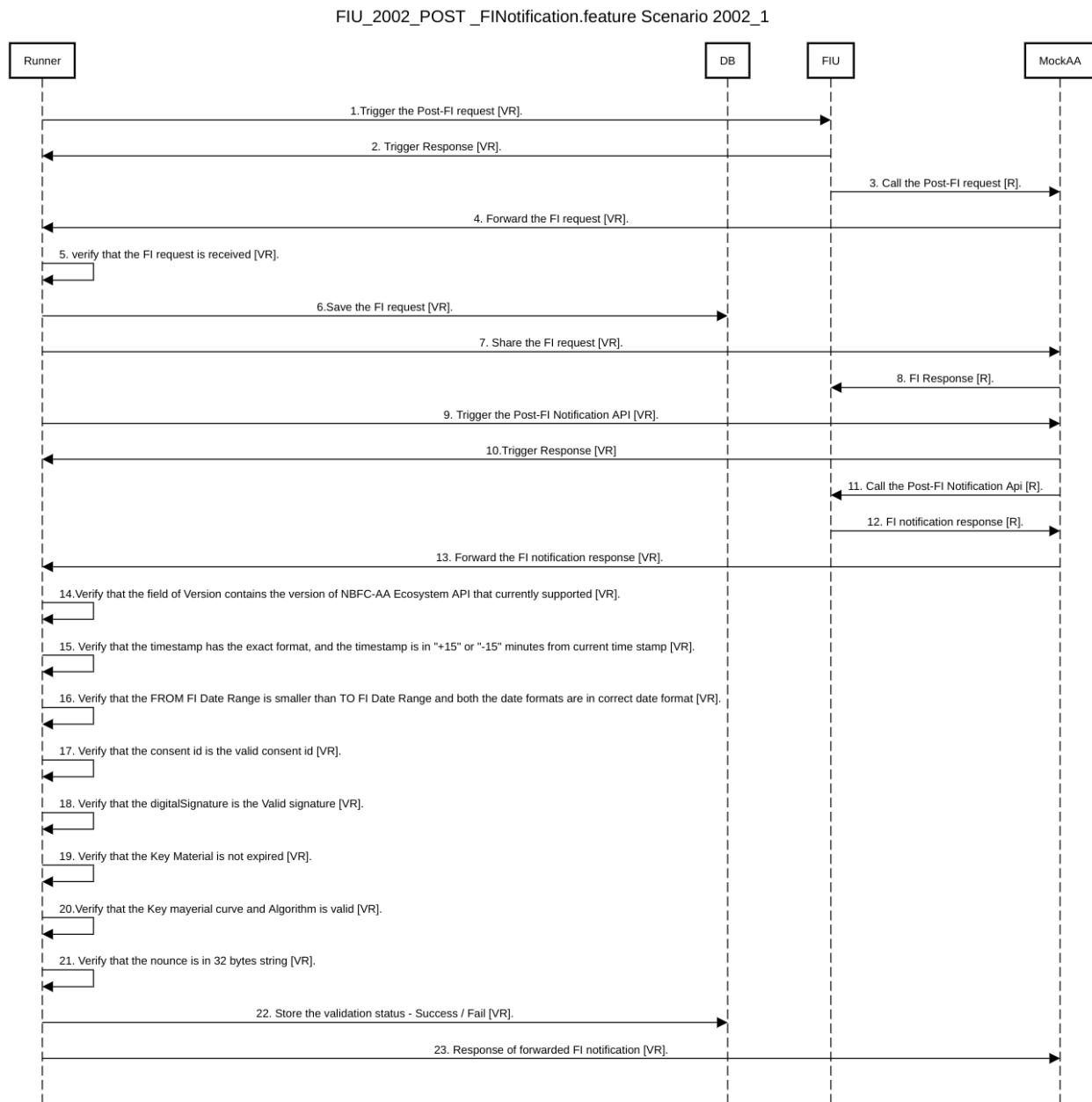
Scenario 2002_1

title FIU_2002_POST_FINotification.feature Scenario 2002_1

participant Runner
participant DB
participant FIU
participant MockAA

Runner->FIU:1. Trigger the Post-FI request [VR].
Runner<-FIU:2. Trigger Response [VR].
FIU->MockAA:3. Call the Post-FI request [R].
MockAA ->Runner:4. Forward the FI request [VR].
Runner->Runner:5. verify that the FI request is received [VR].
Runner -> DB:6. Save the FI request [VR].
Runner->MockAA:7. Share the FI request [VR].
FIU<-MockAA:8. FI Response [R].
Runner->MockAA:9. Trigger the Post-FI Notification API [VR].
Runner<-MockAA:10. Trigger Response [VR]
FIU<-MockAA:11. Call the Post-FI Notification Api [R].
FIU->MockAA:12. FI notification response [R].
Runner<-MockAA:13. Forward the FI notification response [VR].
Runner->Runner:14. Verify that the field of Version contains the version of NBFC-AA Ecosystem API that currently supported [VR].
Runner->Runner:15. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from current time stamp [VR].
Runner->Runner:16. Verify that the FROM FI Date Range is smaller than TO FI Date Range and both the date formats are in correct date format [VR].
Runner->Runner:17. Verify that the consent id is the valid consent id [VR].
Runner->Runner:18. Verify that the digitalSignature is the Valid signature [VR].
Runner->Runner:19. Verify that the Key Material is not expired [VR].
Runner->Runner:20. Verify that the Key material curve and Algorithm is valid [VR].
Runner->Runner:21. Verify that the nonce is in 32 bytes string [VR].

Runner -> DB: 22. Store the validation status - Success / Fail [VR].
 Runner->MockAA:23. Response of forwarded FI notification [VR].



FIU_2003_POST _FINotification.feature

Scenario 2003_1

title FIU_2003_POST _FINotification.feature Scenario 2003_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->FIU:1.Trigger the Post-FI request [VR].

Runner<-FIU:2. Trigger Response [VR].

FIU->MockAA:3. Call the Post-FI request [R].

MockAA ->Runner:4. Forward the FI request [VR].

Runner->Runner:5. verify that the FI request is received [VR].

Runner -> DB:6.Save the FI request [VR].

Runner->MockAA:7. Share the FI request [VR].

FIU<-MockAA:8. FI Response [R].

Runner->MockAA:9. Set FI notification request with incorrect version number [VR].

Runner<-MockAA:10.Status Updated [VR]

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12.Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR].

Runner->Runner:16.Verify that the response code 404 is displayed [VR].

Runner->Runner:17. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from current time stamp.[VR].

Runner->Runner:18. Verify that the FROM FI Date Range is smaller than TO FI Date Range and both the date formats are in correct date format [VR].

Runner->Runner:19. Verify that the Error code is NoSuchVersion [VR].

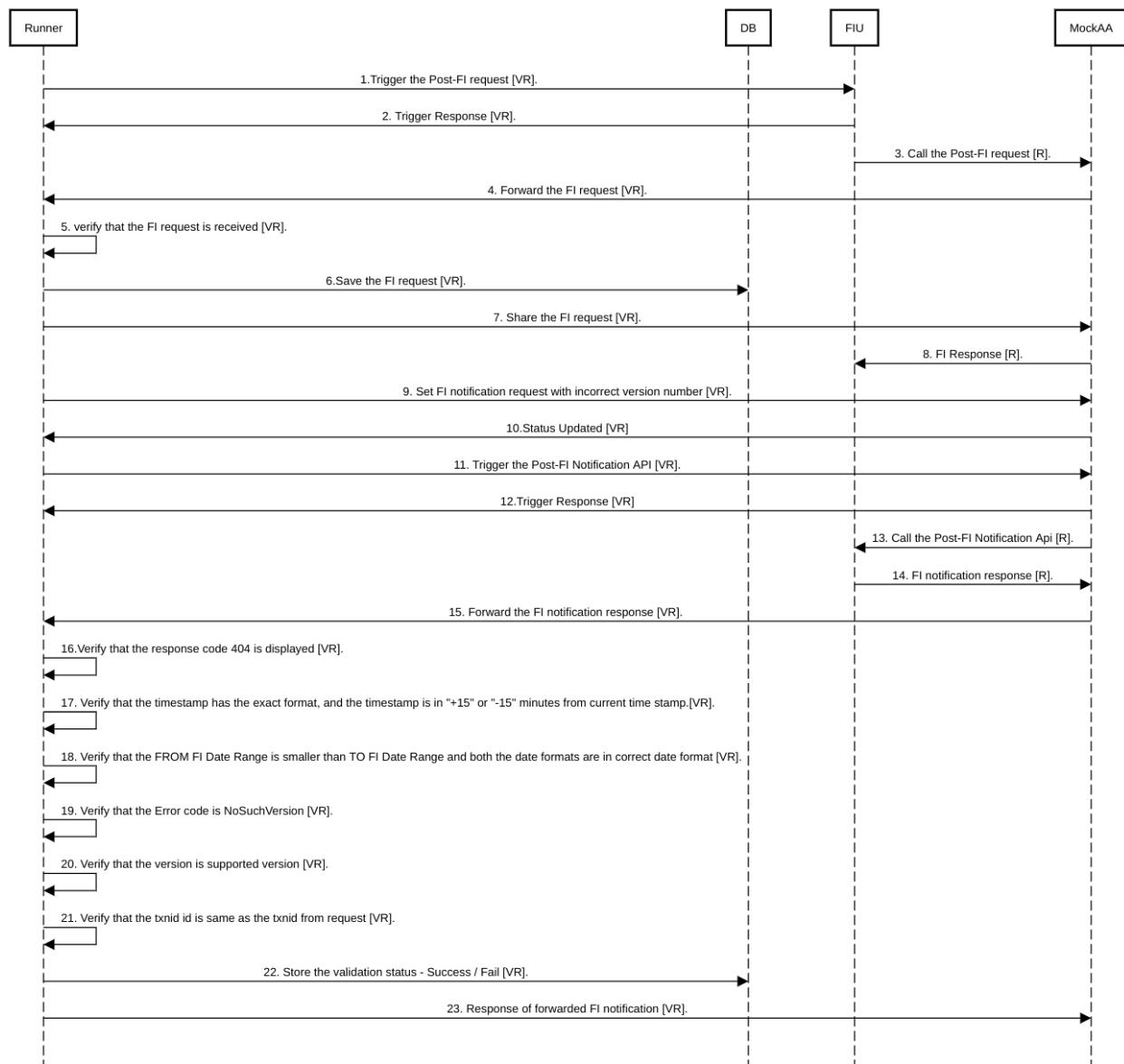
Runner->Runner:20. Verify that the version is supported version [VR].

Runner->Runner:21. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB: 22. Store the validation status - Success / Fail [VR].

Runner->MockAA:23. Response of forwarded FI notification [VR].

FIU_2003_POST _FINotification.feature Scenario 2003_1



Scenario 2003_2

title FIU_2003_POST_FINotification.feature Scenario 2003_2

participant Runner

participant DB

participant FIU

participant MockAA

Runner->FIU:1.Trigger the Post-FI request [VR].

Runner<-FIU:2. Trigger Response [VR].

FIU->MockAA:3. Call the Post-FI request [R].

MockAA ->Runner:4. Forward the FI request [VR].

Runner->Runner:5. verify that the FI request is received [VR].

Runner -> DB:6.Save the FI request [VR].

Runner->MockAA:7. Share the FI request [VR].

FIU<-MockAA:8. FI Response [R].

Runner->MockAA:9. Set FI notification request by adding an extra digit at the end of correct version number [VR].

Runner<-MockAA:10.Status Updated [VR]

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12.Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR].

Runner->Runner:16.Verify that the response code 404 is displayed [VR].

Runner->Runner:17. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp.[VR].

Runner->Runner:18. Verify that the FROM FI Date Range is smaller than TO FI Date Range and both the date formats are in correct date format [VR].

Runner->Runner:19. Verify that the Error code is NoSuchVersion [VR].

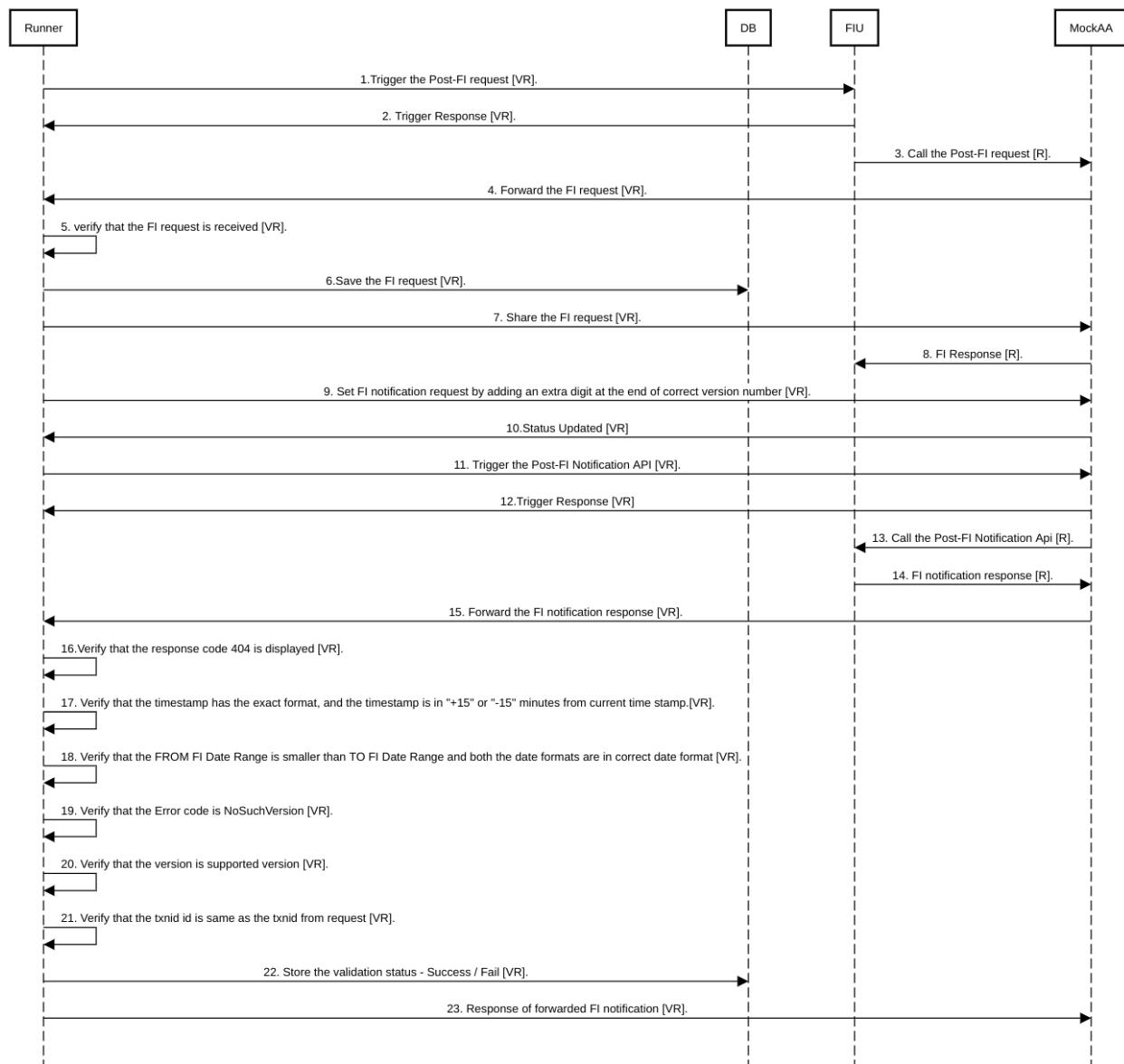
Runner->Runner:20. Verify that the version is supported version [VR].

Runner->Runner:21. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB: 22. Store the validation status - Success / Fail [VR].

Runner->MockAA:23. Response of forwarded FI notification [VR].

FIU_2003_POST _FINotification.feature Scenario 2003_2



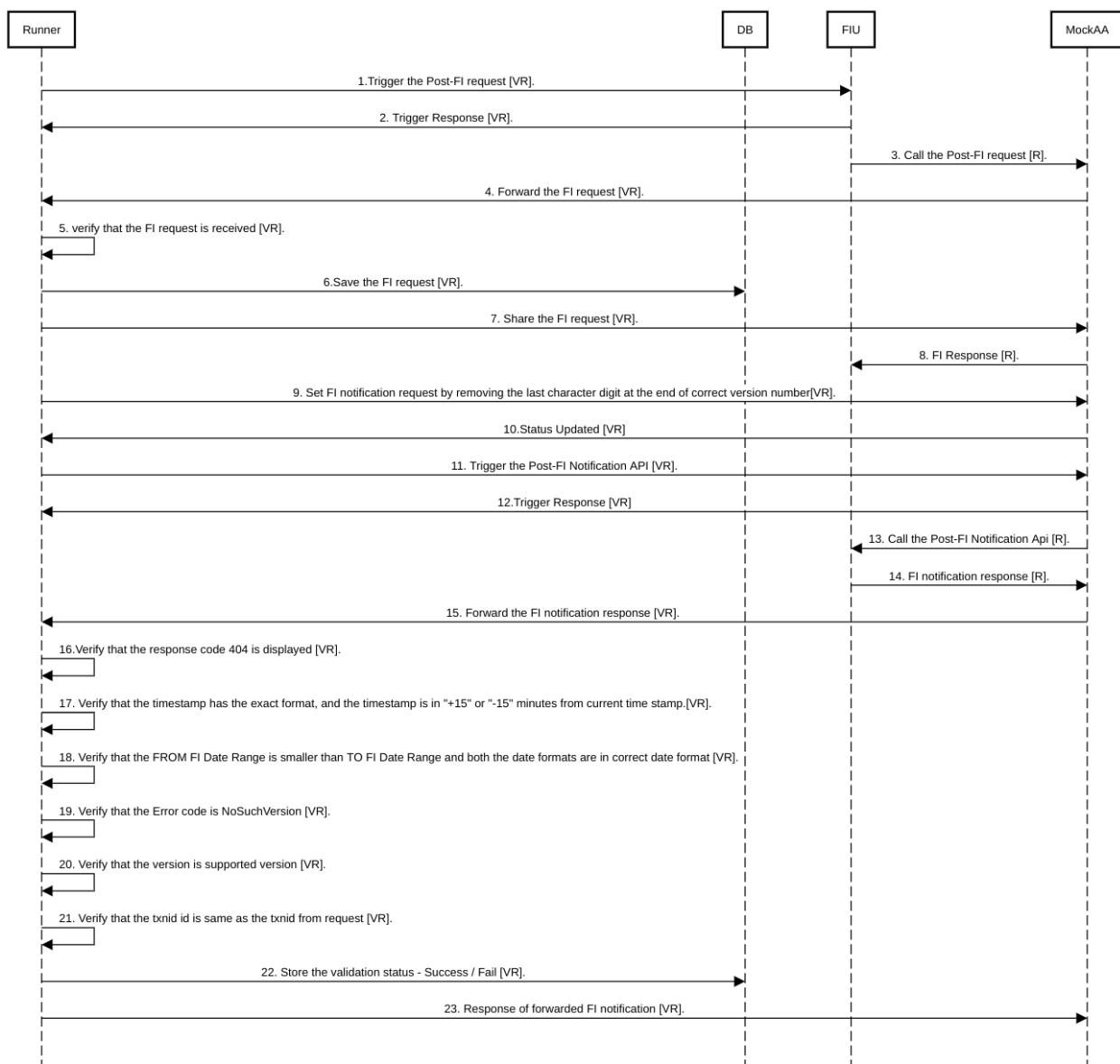
Scenario 2003_3

title FIU_2003_POST_FINotification.feature Scenario 2003_3

participant Runner
participant DB
participant FIU
participant MockAA

Runner->FIU:1. Trigger the Post-FI request [VR].
Runner<-FIU:2. Trigger Response [VR].
FIU->MockAA:3. Call the Post-FI request [R].
MockAA ->Runner:4. Forward the FI request [VR].
Runner->Runner:5. verify that the FI request is received [VR].
Runner -> DB:6. Save the FI request [VR].
Runner->MockAA:7. Share the FI request [VR].
FIU<-MockAA:8. FI Response [R].
Runner->MockAA:9. Set FI notification request by removing the last character digit at the end of the correct version number[VR].
Runner<-MockAA:10. Status Updated [VR]
Runner->MockAA:11. Trigger the Post-FI Notification API [VR].
Runner<-MockAA:12. Trigger Response [VR]
FIU<-MockAA:13. Call the Post-FI Notification Api [R].
FIU->MockAA:14. FI notification response [R].
Runner<-MockAA:15. Forward the FI notification response [VR].
Runner->Runner:16. Verify that the response code 404 is displayed [VR].
Runner->Runner:17. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from current time stamp.[VR].
Runner->Runner:18. Verify that the FROM FI Date Range is smaller than TO FI Date Range and both the date formats are in correct date format [VR].
Runner->Runner:19. Verify that the Error code is NoSuchVersion [VR].
Runner->Runner:20. Verify that the version is supported version [VR].
Runner->Runner:21. Verify that the txnid id is same as the txnid from request [VR].
Runner -> DB: 22. Store the validation status - Success / Fail [VR].
Runner->MockAA:23. Response of forwarded FI notification [VR].

FIU_2003_POST _FINotification.feature Scenario 2003_3



FIU_2004_POST_FINotification.feature

Scenario 2004_1

title FIU_2004_POST_FINotification.feature Scenario 2004_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->FIU:1.Trigger the Post-FI request [VR].

Runner<-FIU:2. Trigger Response [VR].

FIU->MockAA:3. Call the Post-FI request [R].

MockAA ->Runner:4. Forward the FI request [VR].

Runner->Runner:5. verify that the FI request is received [VR].

Runner -> DB:6.Save the FI request [VR].

Runner->MockAA:7. Share the FI request [VR].

FIU<-MockAA:8. FI Response [R].

Runner->MockAA:9. Set FI notification request with random session Id [VR].

Runner<-MockAA:10.Status Updated [VR]

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12.Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR].

Runner->Runner:16.Verify that the response code 400 is displayed [VR].

Runner->Runner:17. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from current time stamp.[VR].

Runner->Runner:18. Verify that the Error code is InvalidSessionId [VR].

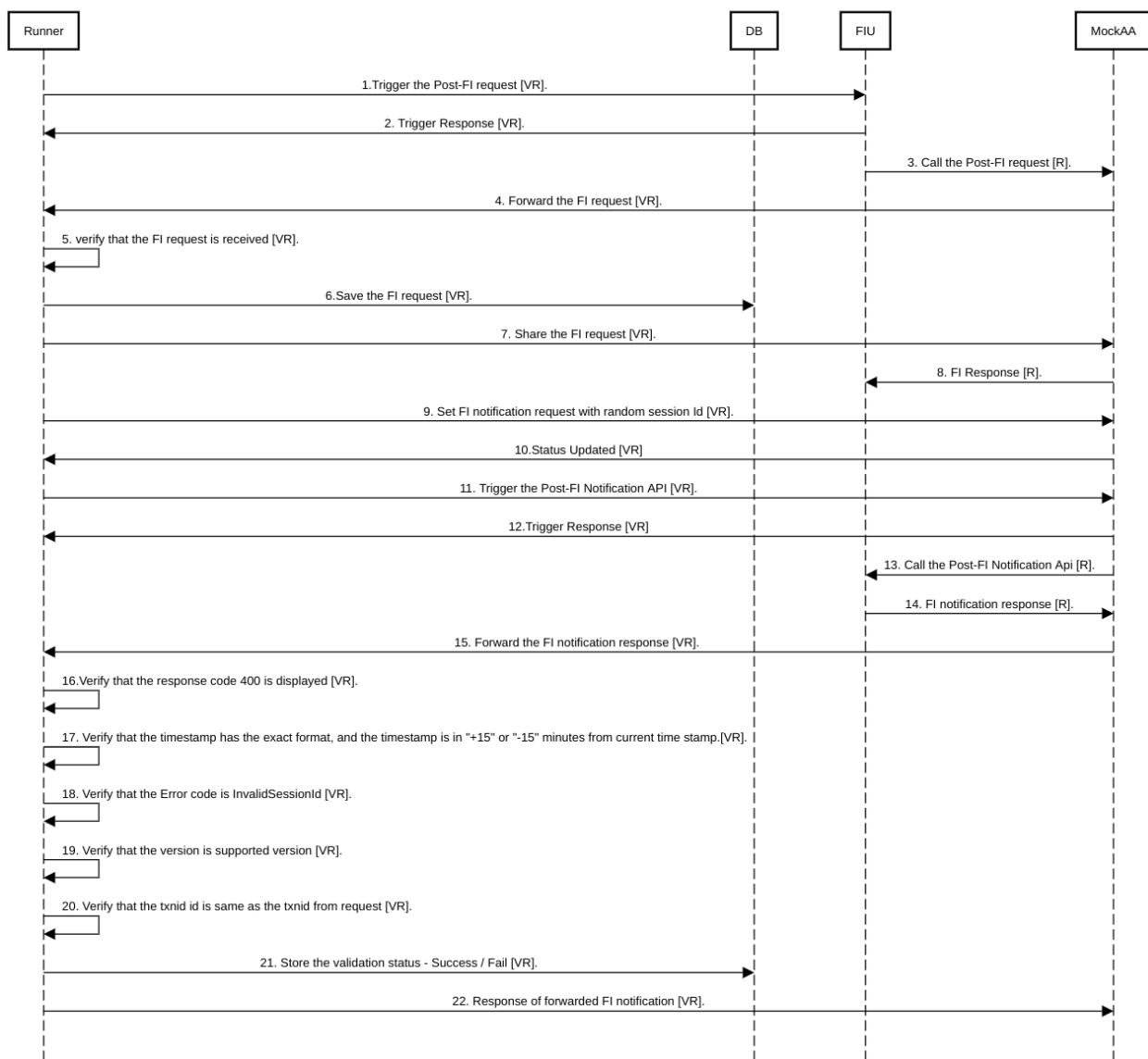
Runner->Runner:19. Verify that the version is supported version [VR].

Runner->Runner:20. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB: 21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2004_POST_FINotification.feature Scenario 2004_1



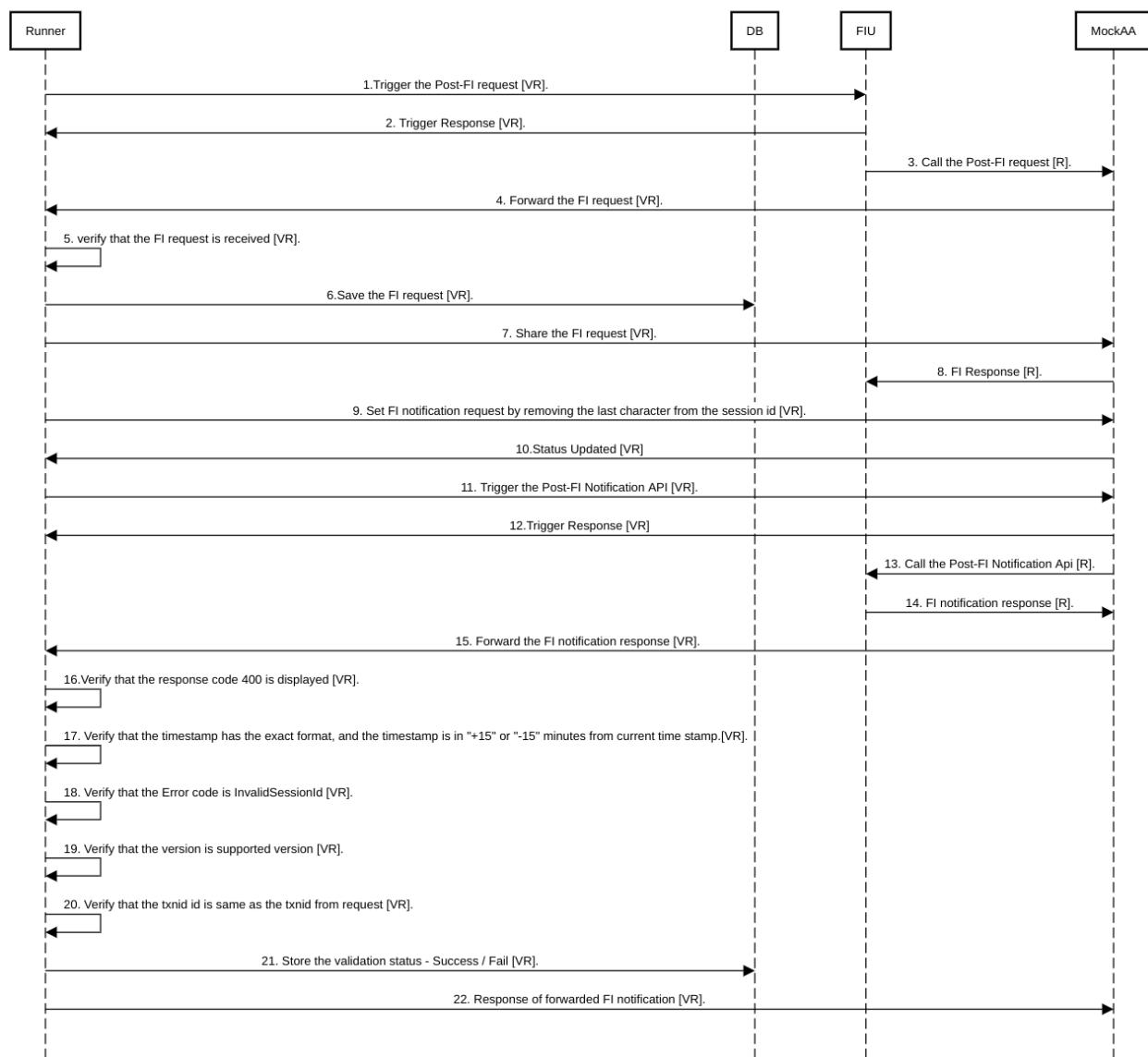
Scenario 2004_2

title FIU_2004_POST_FINotification.feature Scenario 2004_2

participant Runner
participant DB
participant FIU
participant MockAA

Runner->FIU:1. Trigger the Post-FI request [VR].
Runner<-FIU:2. Trigger Response [VR].
FIU->MockAA:3. Call the Post-FI request [R].
MockAA ->Runner:4. Forward the FI request [VR].
Runner->Runner:5. verify that the FI request is received [VR].
Runner -> DB:6. Save the FI request [VR].
Runner->MockAA:7. Share the FI request [VR].
FIU<-MockAA:8. FI Response [R].
Runner->MockAA:9. Set FI notification request by removing the last character from the session id [VR].
Runner<-MockAA:10. Status Updated [VR]
Runner->MockAA:11. Trigger the Post-FI Notification API [VR].
Runner<-MockAA:12. Trigger Response [VR]
FIU<-MockAA:13. Call the Post-FI Notification Api [R].
FIU->MockAA:14. FI notification response [R].
Runner<-MockAA:15. Forward the FI notification response [VR].
Runner->Runner:16. Verify that the response code 400 is displayed [VR].
Runner->Runner:17. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp.[VR].
Runner->Runner:18. Verify that the Error code is InvalidSessionId [VR].
Runner->Runner:19. Verify that the version is supported version [VR].
Runner->Runner:20. Verify that the txnid id is the same as the txnid from request [VR].
Runner -> DB: 21. Store the validation status - Success / Fail [VR].
Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2004_POST_FINotification.feature Scenario 2004_2



Scenario 2004_3

title FIU_2004_POST_FINotification.feature Scenario 2004_3

participant Runner

participant DB

participant FIU

participant MockAA

Runner->FIU:1.Trigger the Post-FI request [VR].

Runner<-FIU:2. Trigger Response [VR].

FIU->MockAA:3. Call the Post-FI request [R].

MockAA ->Runner:4. Forward the FI request [VR].

Runner->Runner:5. verify that the FI request is received [VR].

Runner -> DB:6.Save the FI request [VR].

Runner->MockAA:7. Share the FI request [VR].

FIU<-MockAA:8. FI Response [R].

Runner->MockAA:9. Set FI notification request by using the session id of another FI request [VR].

Runner<-MockAA:10.Status Updated [VR]

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12.Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR].

Runner->Runner:16.Verify that the response code 400 is displayed [VR].

Runner->Runner:17. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current timestamp.[VR].

Runner->Runner:18. Verify that the Error code is InvalidSessionId [VR].

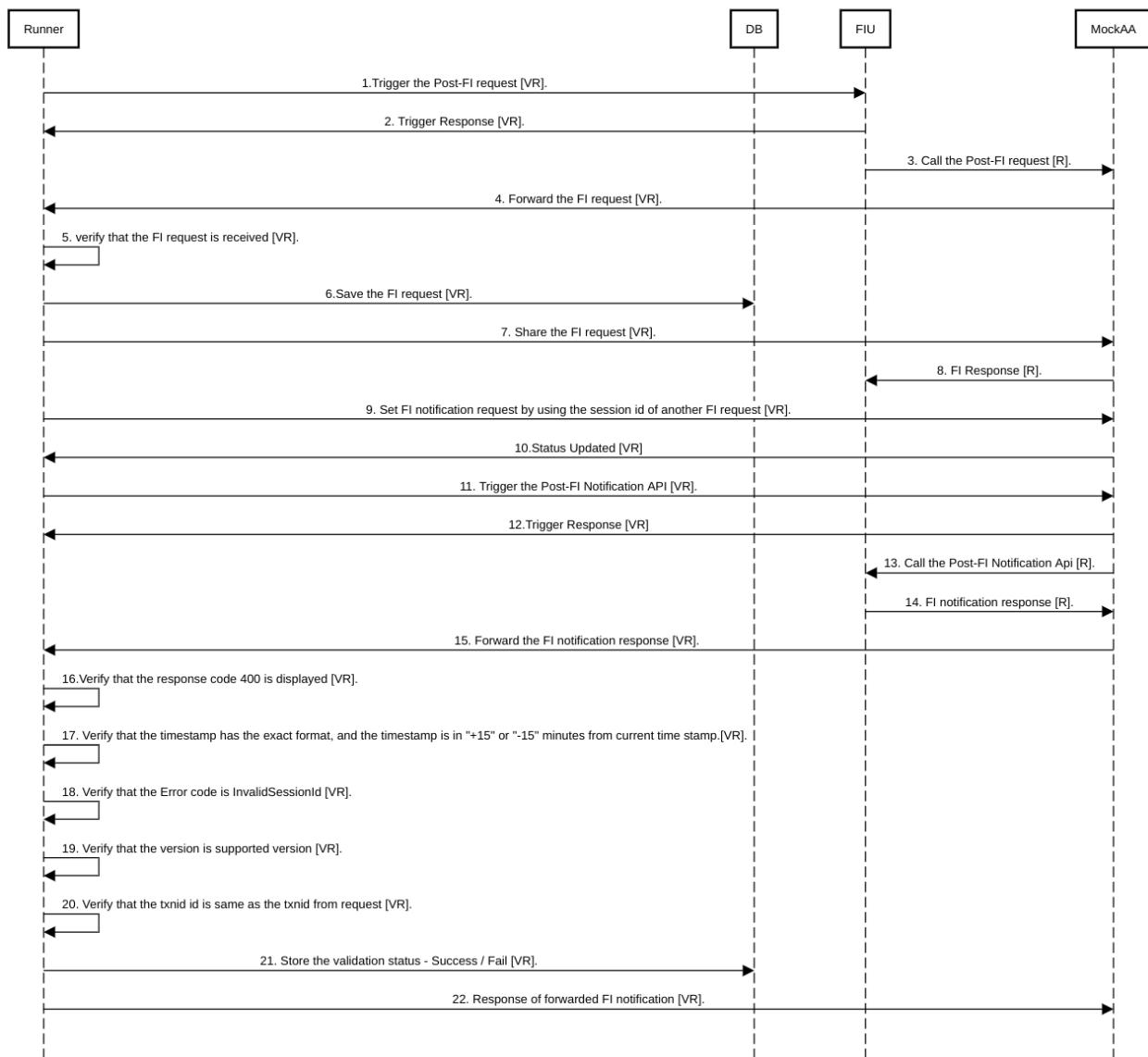
Runner->Runner:19. Verify that the version is supported version [VR].

Runner->Runner:20. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB: 21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2004_POST_FINotification.feature Scenario 2004_3



FIU_2005_POST_FINotification.feature

Scenario 2005_1

title FIU_2005_POST_FINotification.feature Scenario 2005_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->FIU:1.Trigger the Post-FI request [VR].

Runner<-FIU:2. Trigger Response [VR].

FIU->MockAA:3. Call the Post-FI request [R].

MockAA ->Runner:4. Forward the FI request [VR].

Runner->Runner:5. verify that the FI request is received [VR].

Runner -> DB:6.Save the FI request [VR].

Runner->MockAA:7. Share the FI request [VR].

FIU<-MockAA:8. FI Response [R].

Runner->MockAA:9. Set FI notification request with notifier id as alternate AA [VR].

Runner<-MockAA:10.Status Updated [VR]

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12.Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR].

Runner->Runner:16.Verify that the response code 400 is displayed [VR].

Runner->Runner:17. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp.[VR].

Runner->Runner:18. Verify that the Error code is InvalidSessionId [VR].

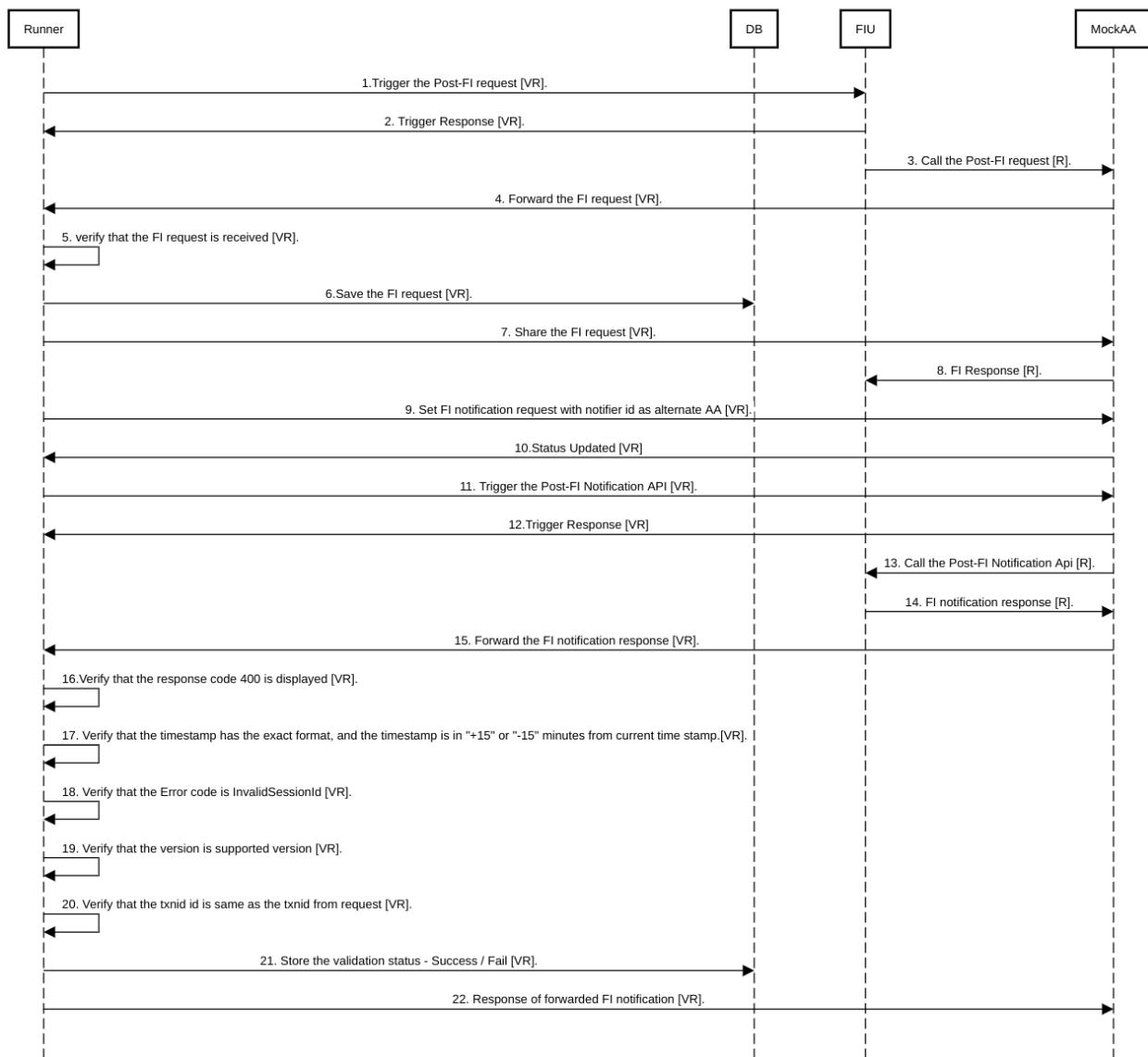
Runner->Runner:19. Verify that the version is supported version [VR].

Runner->Runner:20. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB: 21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2005_POST_FINotification.feature Scenario 2005_1



FIU_2006_POST_FINotification.feature

Scenario 2006_1

title FIU_2006_POST_FINotification.feature Scenario 2006_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->FIU:1. Trigger the Post-FI request [VR].

Runner<-FIU:2. Trigger Response [VR].

FIU->MockAA:3. Call the Post-FI request [R].

MockAA ->Runner:4. Forward the FI request [VR].

Runner->Runner:5. verify that the FI request is received [VR].

Runner -> DB:6. Save the FI request [VR].

Runner->MockAA:7. Share the FI request [VR].

FIU<-MockAA:8. FI Response [R].

Runner->MockAA:9. Set FI notification request with one required field set to null [VR].

Runner<-MockAA:10. Status Updated [VR]

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12. Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR].

Runner->Runner:16. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp.[VR].

Runner->Runner:17. Verify that the error code displayed according to the fields [VR].

Runner->Runner:18. Verify that the Error code is InvalidRequest [VR].

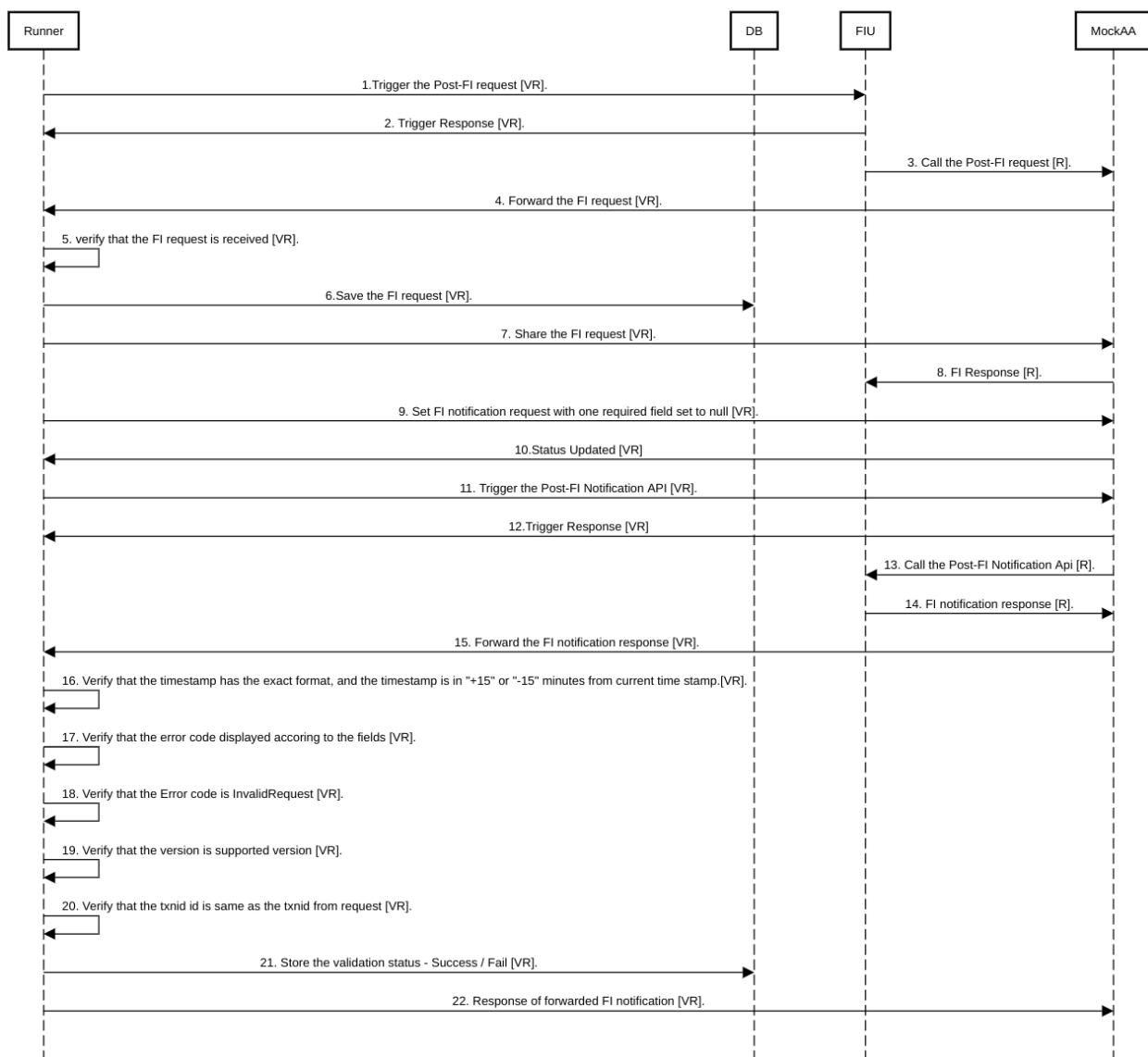
Runner->Runner:19. Verify that the version is supported version [VR].

Runner->Runner:20. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB: 21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2006_POST_FINotification.feature Scenario 2006_1



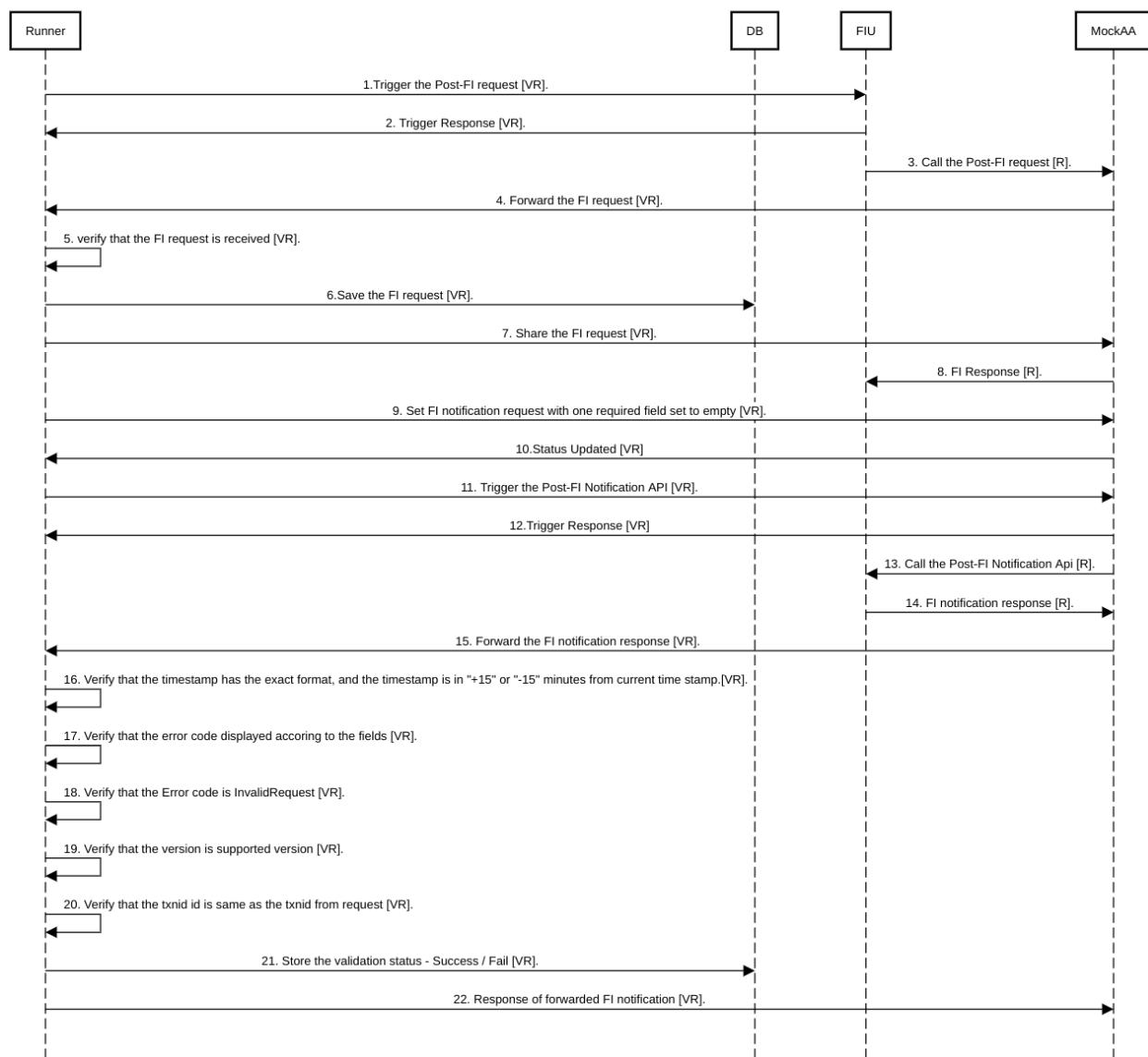
Scenario 2006_2

title FIU_2006_POST_FINotification.feature Scenario 2006_2

participant Runner
participant DB
participant FIU
participant MockAA

Runner->FIU:1. Trigger the Post-FI request [VR].
Runner<-FIU:2. Trigger Response [VR].
FIU->MockAA:3. Call the Post-FI request [R].
MockAA ->Runner:4. Forward the FI request [VR].
Runner->Runner:5. verify that the FI request is received [VR].
Runner -> DB:6. Save the FI request [VR].
Runner->MockAA:7. Share the FI request [VR].
FIU<-MockAA:8. FI Response [R].
Runner->MockAA:9. Set FI notification request with one required field set to empty [VR].
Runner<-MockAA:10. Status Updated [VR]
Runner->MockAA:11. Trigger the Post-FI Notification API [VR].
Runner<-MockAA:12. Trigger Response [VR]
FIU<-MockAA:13. Call the Post-FI Notification Api [R].
FIU->MockAA:14. FI notification response [R].
Runner<-MockAA:15. Forward the FI notification response [VR].
Runner->Runner:16. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from current time stamp.[VR].
Runner->Runner:17. Verify that the error code displayed accoring to the fields [VR].
Runner->Runner:18. Verify that the Error code is InvalidRequest [VR].
Runner->Runner:19. Verify that the version is supported version [VR].
Runner->Runner:20. Verify that the txnid id is same as the txnid from request [VR].
Runner -> DB: 21. Store the validation status - Success / Fail [VR].
Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2006_POST_FINotification.feature Scenario 2006_2



Scenario 2006_3

title FIU_2006_POST_FINotification.feature Scenario 2006_3

participant Runner

participant DB

participant FIU

participant MockAA

Runner->FIU:1.Trigger the Post-FI request [VR].

Runner<-FIU:2. Trigger Response [VR].

FIU->MockAA:3. Call the Post-FI request [R].

MockAA ->Runner:4. Forward the FI request [VR].

Runner->Runner:5. verify that the FI request is received [VR].

Runner -> DB:6.Save the FI request [VR].

Runner->MockAA:7. Share the FI request [VR].

FIU<-MockAA:8. FI Response [R].

Runner->MockAA:9. Set FI notification request with one required field set to 0(Zero) [VR].

Runner<-MockAA:10.Status Updated [VR]

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12.Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR].

Runner->Runner:16. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from current time stamp.[VR].

Runner->Runner:17. Verify that the error code displayed accoring to the fields [VR].

Runner->Runner:18. Verify that the Error code is InvalidRequest [VR].

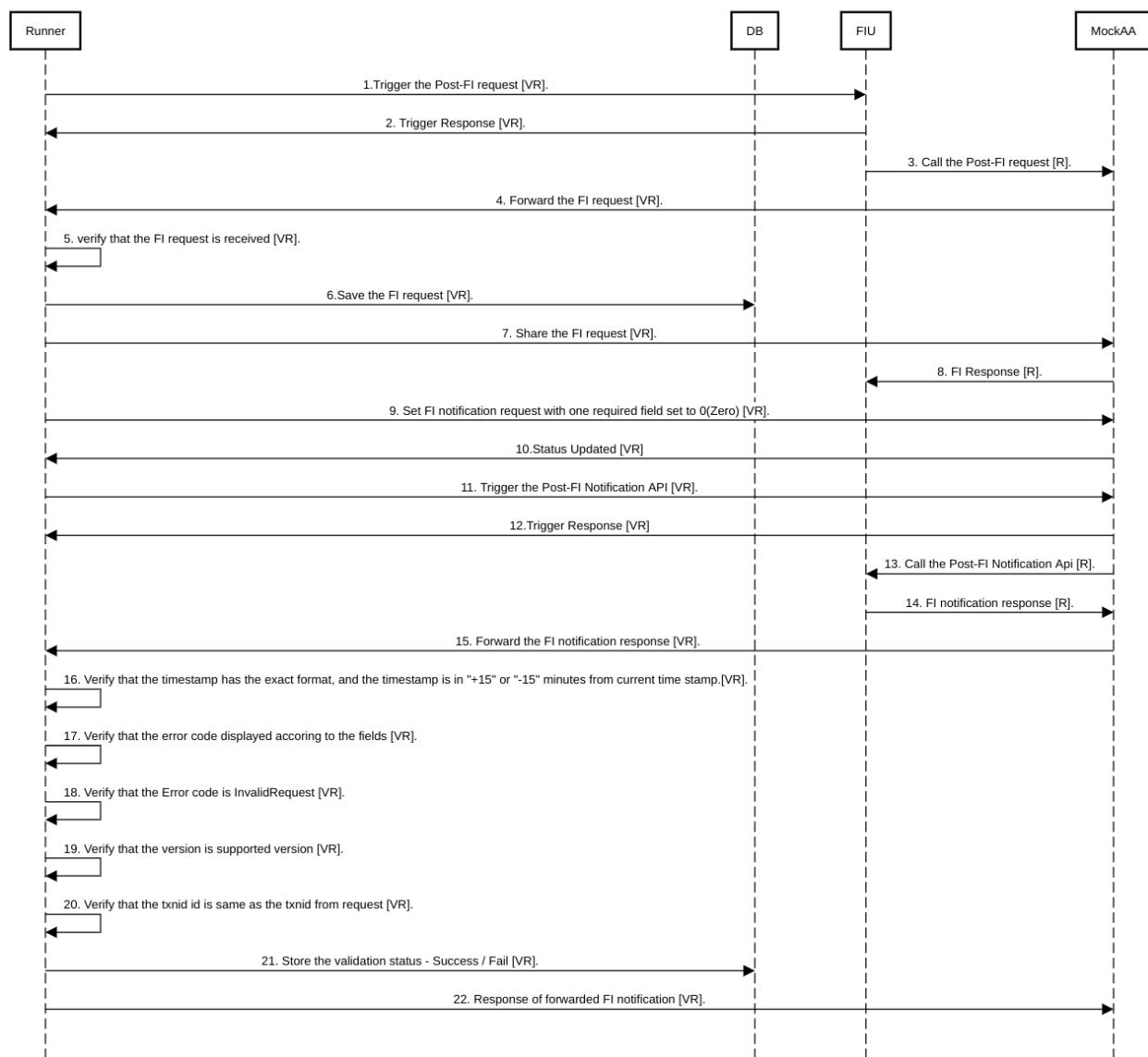
Runner->Runner:19. Verify that the version is supported version [VR].

Runner->Runner:20. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB: 21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2006_POST_FINotification.feature Scenario 2006_3



Scenario 2006_4

title FIU_2006_POST_FINotification.feature Scenario 2006_4

participant Runner

participant DB

participant FIU

participant MockAA

Runner->FIU:1. Trigger the Post-FI request [VR].

Runner<-FIU:2. Trigger Response [VR].

FIU->MockAA:3. Call the Post-FI request [R].

MockAA ->Runner:4. Forward the FI request [VR].

Runner->Runner:5. verify that the FI request is received [VR].

Runner -> DB:6. Save the FI request [VR].

Runner->MockAA:7. Share the FI request [VR].

FIU<-MockAA:8. FI Response [R].

Runner->MockAA:9. Set FI notification request with one enum field set to non enum value by adding an extra character at the end of last enum and removing the last character at the end of first enum [VR].

Runner<-MockAA:10. Status Updated [VR]

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12. Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR].

Runner->Runner:16. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from current time stamp.[VR].

Runner->Runner:17. Verify that the error code displayed accoring to the fields [VR].

Runner->Runner:18. Verify that the Error code is InvalidRequest [VR].

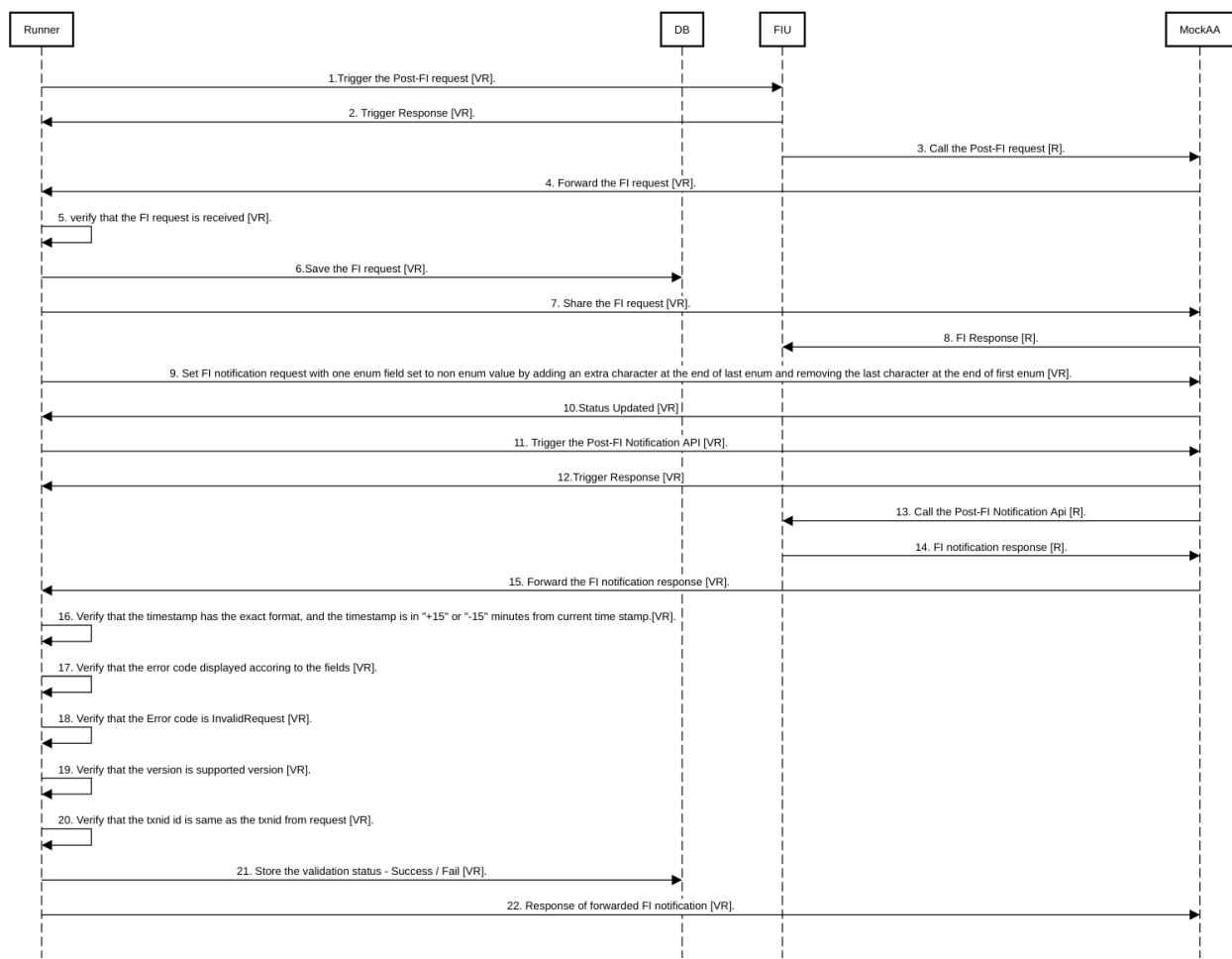
Runner->Runner:19. Verify that the version is supported version [VR].

Runner->Runner:20. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB: 21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2006_POST _FINotification.feature Scenario 2006_4



Scenario 2006_5

title FIU_2006_POST_FINotification.feature Scenario 2006_5

participant Runner

participant DB

participant FIU

participant MockAA

Runner->FIU:1. Trigger the Post-FI request [VR].

Runner<-FIU:2. Trigger Response [VR].

FIU->MockAA:3. Call the Post-FI request [R].

MockAA ->Runner:4. Forward the FI request [VR].

Runner->Runner:5. verify that the FI request is received [VR].

Runner -> DB:6. Save the FI request [VR].

Runner->MockAA:7. Share the FI request [VR].

FIU<-MockAA:8. FI Response [R].

Runner->MockAA:9. Set FI notification request with one array field set to empty array [VR].

Runner<-MockAA:10. Status Updated [VR]

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12. Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR].

Runner->Runner:16. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from current time stamp.[VR].

Runner->Runner:17. Verify that the error code displayed according to the fields [VR].

Runner->Runner:18. Verify that the Error code is InvalidRequest [VR].

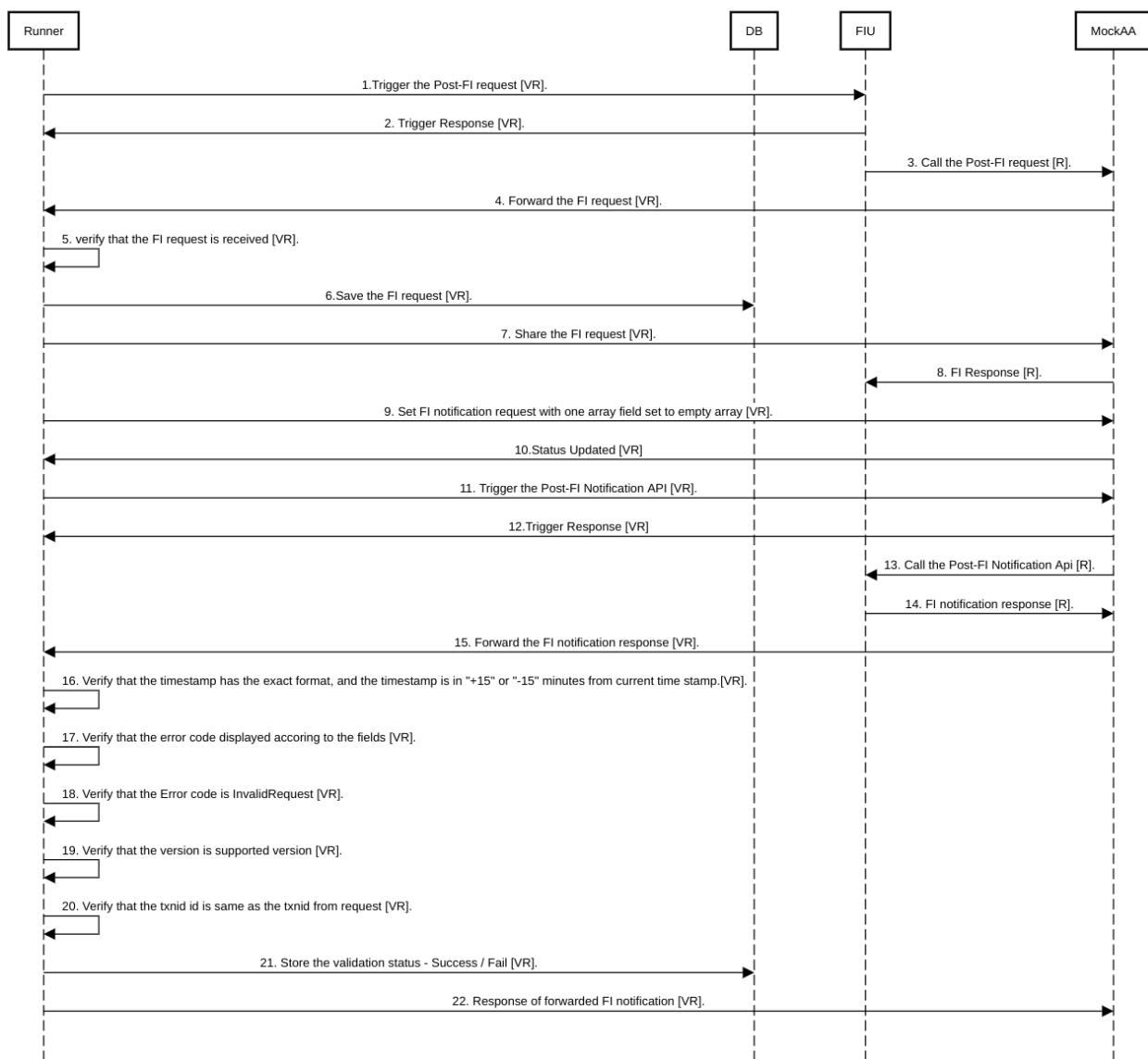
Runner->Runner:19. Verify that the version is supported version [VR].

Runner->Runner:20. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB: 21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2006_POST_FINotification.feature Scenario 2006_5



Scenario 2006_6

title FIU_2006_POST_FINotification.feature Scenario 2006_6

participant Runner

participant DB

participant FIU

participant MockAA

Runner->FIU:1. Trigger the Post-FI request [VR].

Runner<-FIU:2. Trigger Response [VR].

FIU->MockAA:3. Call the Post-FI request [R].

MockAA ->Runner:4. Forward the FI request [VR].

Runner->Runner:5. verify that the FI request is received [VR].

Runner -> DB:6. Save the FI request [VR].

Runner->MockAA:7. Share the FI request [VR].

FIU<-MockAA:8. FI Response [R].

Runner->MockAA:9. Set FI notification request with one object field set to empty object [VR].

Runner<-MockAA:10. Status Updated [VR]

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12. Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR].

Runner->Runner:16. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from current time stamp.[VR].

Runner->Runner:17. Verify that the error code displayed according to the fields [VR].

Runner->Runner:18. Verify that the Error code is InvalidRequest [VR].

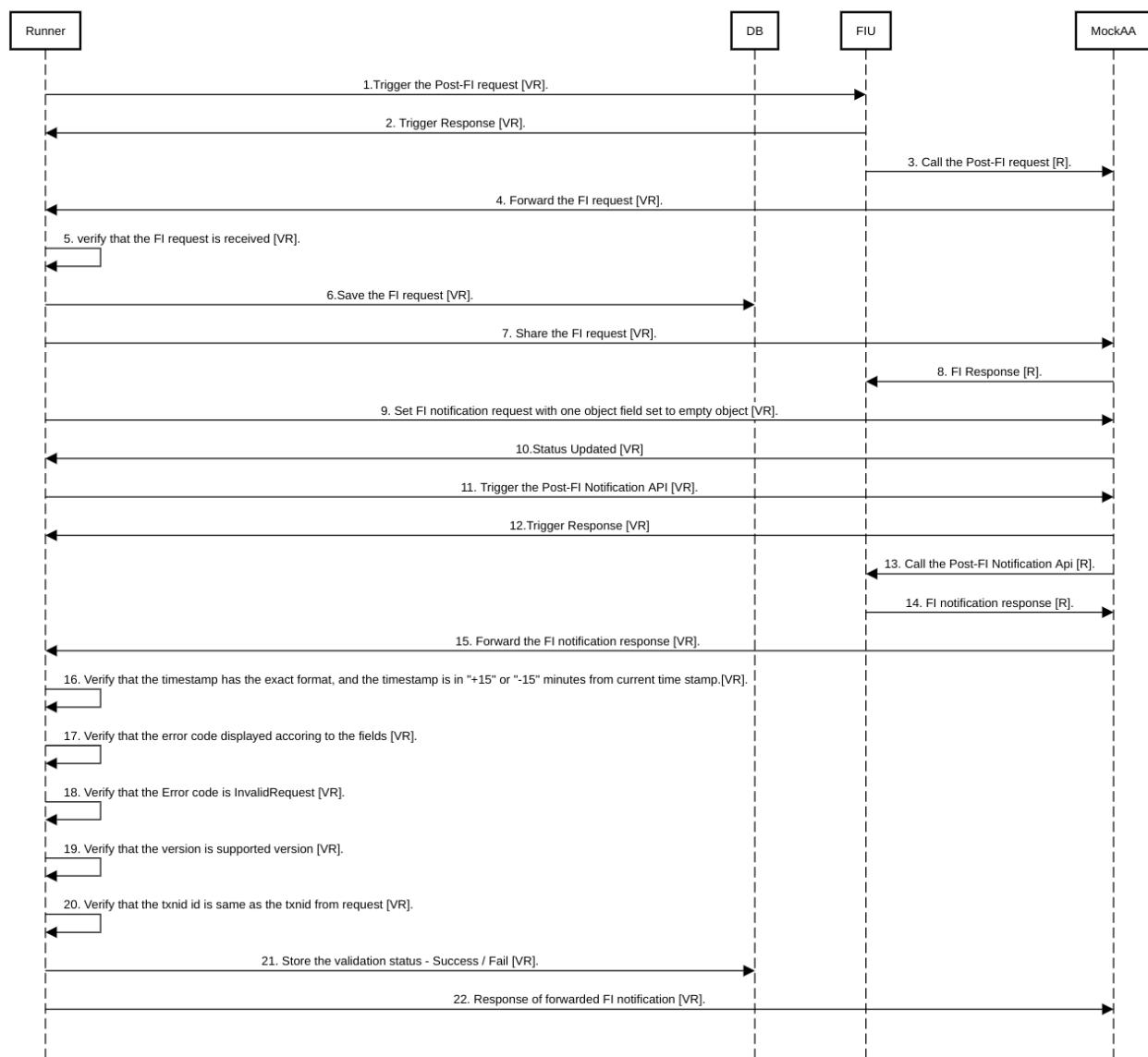
Runner->Runner:19. Verify that the version is supported version [VR].

Runner->Runner:20. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB: 21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2006_POST_FINotification.feature Scenario 2006_6



FIU_2007_POST_FINotification.feature

Scenario 2007_1

title F FIU_2007_POST_FINotification.feature Scenario 2007_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->FIU:1.Trigger the Post-FI request [VR].

Runner<-FIU:2. Trigger Response [VR].

FIU->MockAA:3. Call the Post-FI request [R].

MockAA ->Runner:4. Forward the FI request [VR].

Runner->Runner:5. verify that the FI request is received [VR].

Runner -> DB:6.Save the FI request [VR].

Runner->MockAA:7. Share the FI request [VR].

FIU<-MockAA:8. FI Response [R].

Runner->MockAA:9. Set FI notification response by generating a UUID string and adding an extra character at the end of txnid.[VR].

Runner<-MockAA:10.Status Updated [VR]

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12.Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR].

Runner->Runner:16. Verify that the response code 400 is displayed [VR].

Runner->Runner:17. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from current time stamp.[VR].

Runner->Runner:18. Verify that the Error code is InvalidRequest [VR].

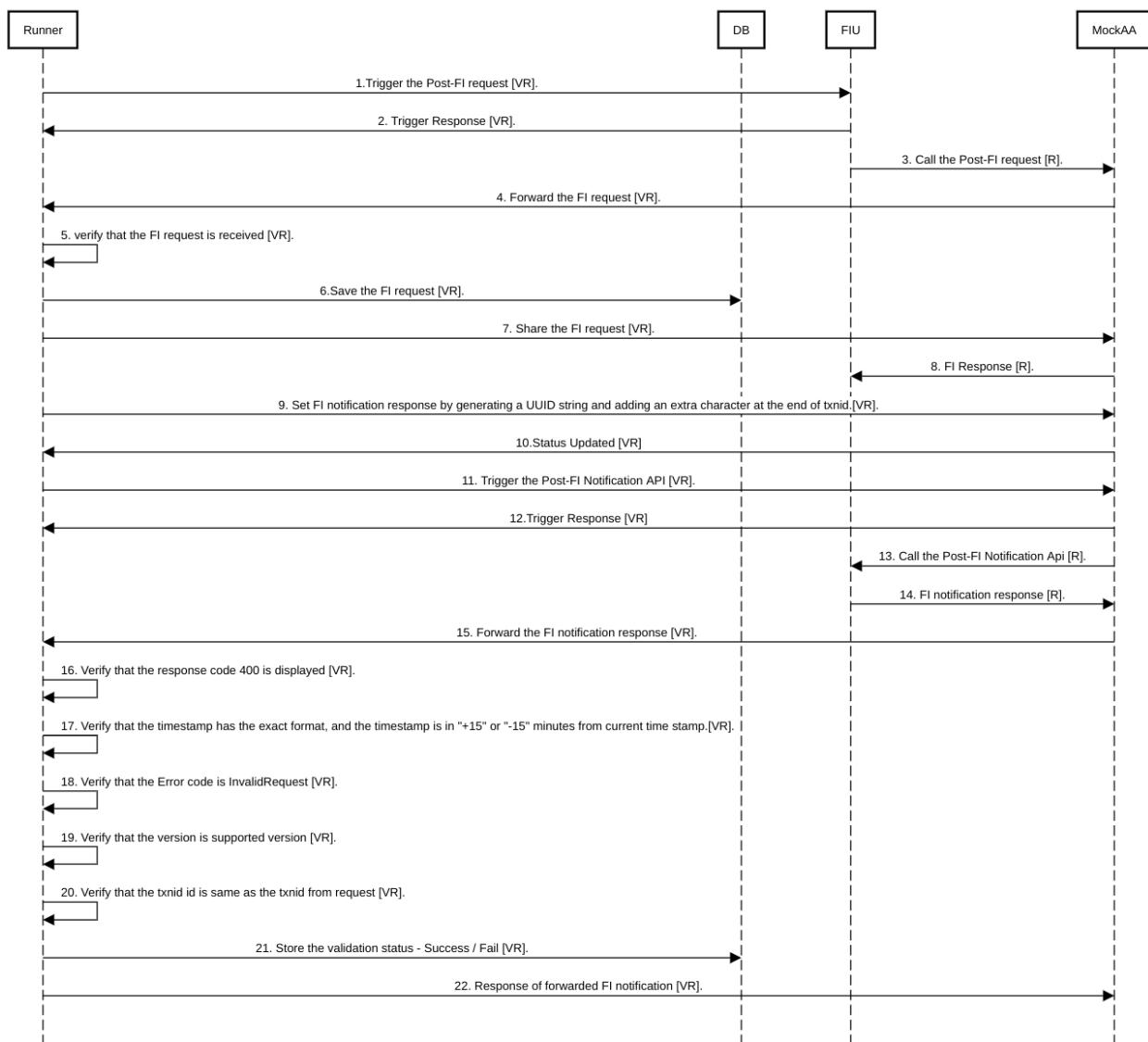
Runner->Runner:19. Verify that the version is supported version [VR].

Runner->Runner:20. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB: 21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

F FIU_2007_POST _FINotification.feature Scenario 2007_1



Scenario 2007_2

title FIU_2007_POST_FINotification.feature Scenario 2007_2

participant Runner

participant DB

participant FIU

participant MockAA

Runner->FIU:1. Trigger the Post-FI request [VR].

Runner<-FIU:2. Trigger Response [VR].

FIU->MockAA:3. Call the Post-FI request [R].

MockAA ->Runner:4. Forward the FI request [VR].

Runner->Runner:5. verify that the FI request is received [VR].

Runner -> DB:6. Save the FI request [VR].

Runner->MockAA:7. Share the FI request [VR].

FIU<-MockAA:8. FI Response [R].

Runner->MockAA:9. Set FI notification response by generating a UUID string and removing the last character at the end of txnid [VR].

Runner<-MockAA:10. Status Updated [VR]

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12. Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR].

Runner->Runner:16. Verify that the response code 400 is displayed [VR].

Runner->Runner:17. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from current time stamp.[VR].

Runner->Runner:18. Verify that the Error code is InvalidRequest [VR].

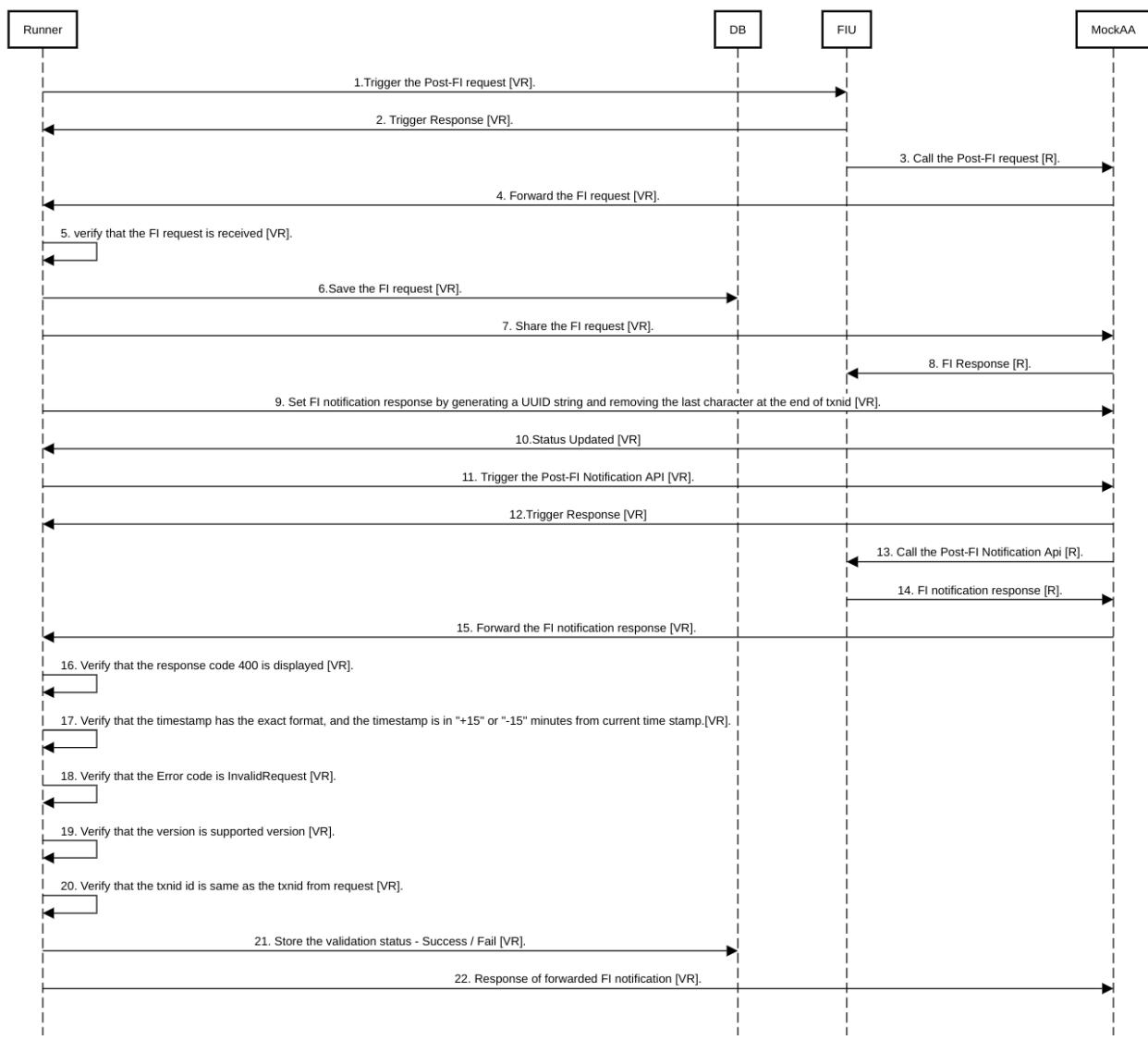
Runner->Runner:19. Verify that the version is supported version [VR].

Runner->Runner:20. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB: 21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2007_POST _FINotification.feature Scenario 2007_2



FIU_2008_POST_FINotification.feature

Scenario 2008_1

title FIU_2008_POST_FINotification.feature Scenario 2008_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->FIU:1. Trigger the Post-FI request [VR].

Runner<-FIU:2. Trigger Response [VR].

FIU->MockAA:3. Call the Post-FI request [R].

MockAA ->Runner:4. Forward the FI request [VR].

Runner->Runner:5. verify that the FI request is received [VR].

Runner -> DB:6. Save the FI request [VR].

Runner->MockAA:7. Share the FI request [VR].

FIU<-MockAA:8. FI Response [R].

Runner->MockAA:9. Set FI notification response with the future timestamp(current time + 15 minutes) [VR].

Runner<-MockAA:10. Status Updated [VR]

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12. Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR].

Runner->Runner:16. Verify that the response code 400 is displayed [VR].

Runner->Runner:17. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp.[VR].

Runner->Runner:18. Verify that the Error code is InvalidRequest [VR].

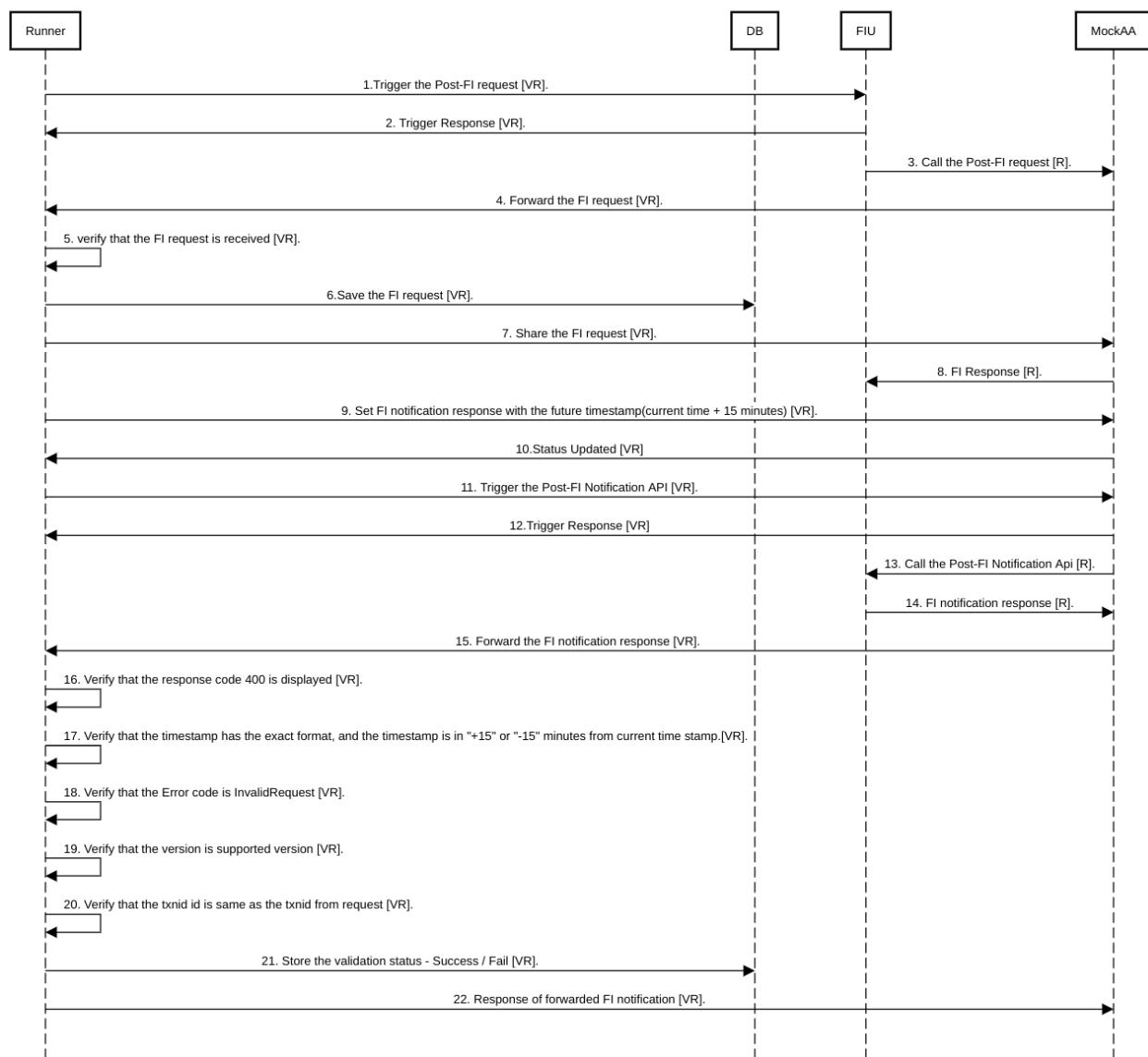
Runner->Runner:19. Verify that the version is supported version [VR].

Runner->Runner:20. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB: 21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2008_POST_FINotification.feature Scenario 2008_1



Scenario 2008_2

title FIU_2008_POST_FINotification.feature Scenario 2008_2

participant Runner

participant DB

participant FIU

participant MockAA

Runner->FIU:1. Trigger the Post-FI request [VR].

Runner<-FIU:2. Trigger Response [VR].

FIU->MockAA:3. Call the Post-FI request [R].

MockAA ->Runner:4. Forward the FI request [VR].

Runner->Runner:5. verify that the FI request is received [VR].

Runner -> DB:6. Save the FI request [VR].

Runner->MockAA:7. Share the FI request [VR].

FIU<-MockAA:8. FI Response [R].

Runner->MockAA:9. Set FI notification response with expired timestamp(current time - 15 minutes) [VR].

Runner<-MockAA:10. Status Updated [VR]

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12. Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR].

Runner->Runner:16. Verify that the response code 400 is displayed [VR].

Runner->Runner:17. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from current time stamp.[VR].

Runner->Runner:18. Verify that the Error code is InvalidRequest [VR].

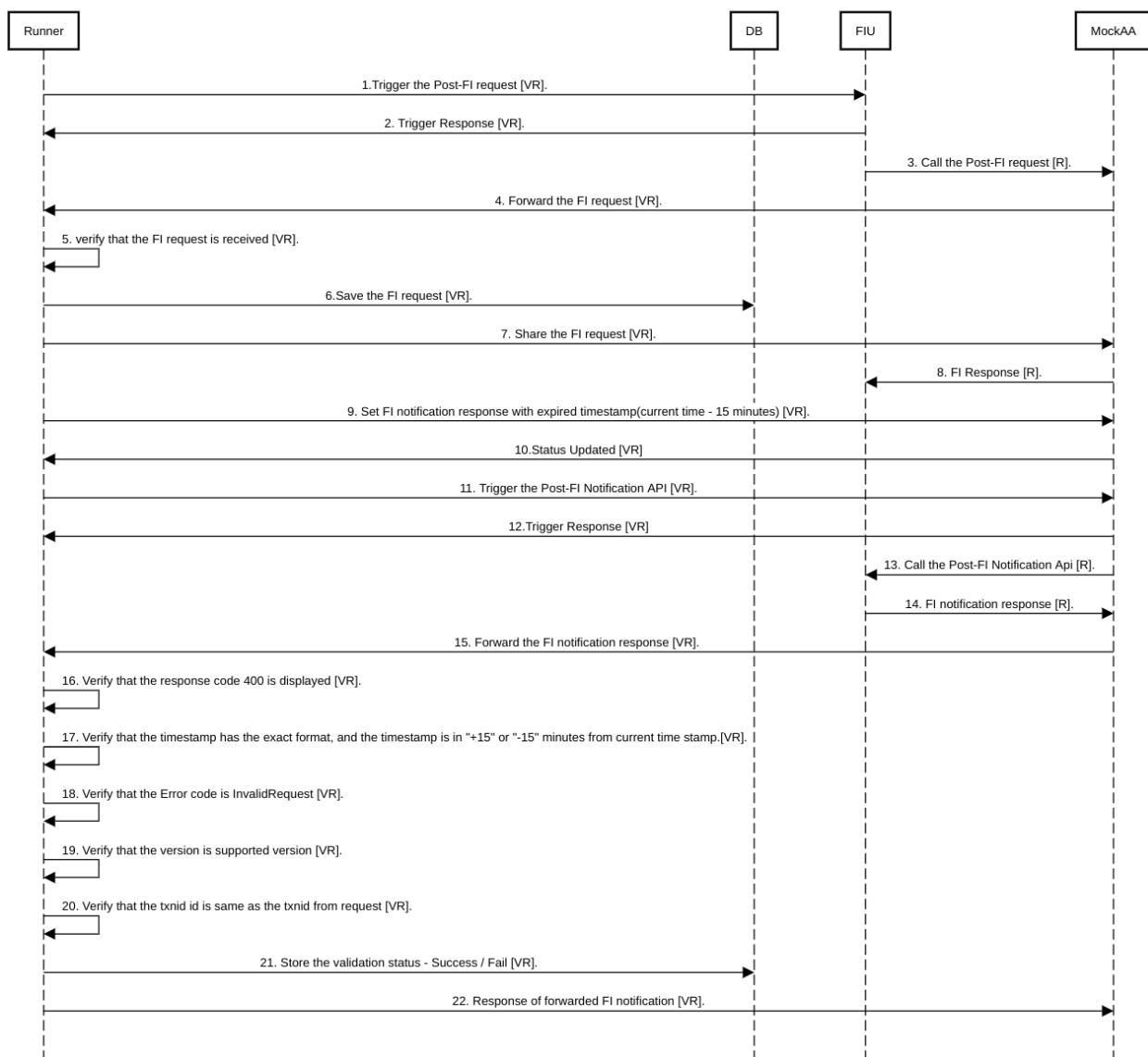
Runner->Runner:19. Verify that the version is supported version [VR].

Runner->Runner:20. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB: 21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2008_POST_FINotification.feature Scenario 2008_2



FIU_2009_POST_FIRequest.feature

Scenario 2009_1 [NC]

title FIU_2009_POST_FIRequest.feature Scenario 2009_1

participant Runner

participant DB

participant FIU

participant MockAA

participant Alternate MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR].

Runner<-MockAA:2. Status Updated [VR]

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

FIU->Alternate MockAA:11. Call the Post-Consent API [VR].

Runner<-Alternate MockAA:12. Forward the alternate mockAA consent request [VR]

Runner->Runner:13 verify that the alternate consent request is received [VR].

Runner->DB:14. Save the alternate consent request [VR].

Runner->Alternate MockAA:15. Share the alternate consent request[VR]

FIU<-Alternate MockAA:16. Consent Response [VR]

Runner->MockAA:17. Set the consent handle response with "Pending" Status [VR].

Runner<-MockAA:18. Status Updated [VR].

Runner->Alternate MockAA:19. Set the get consent response on Alternate MockAA [VR].

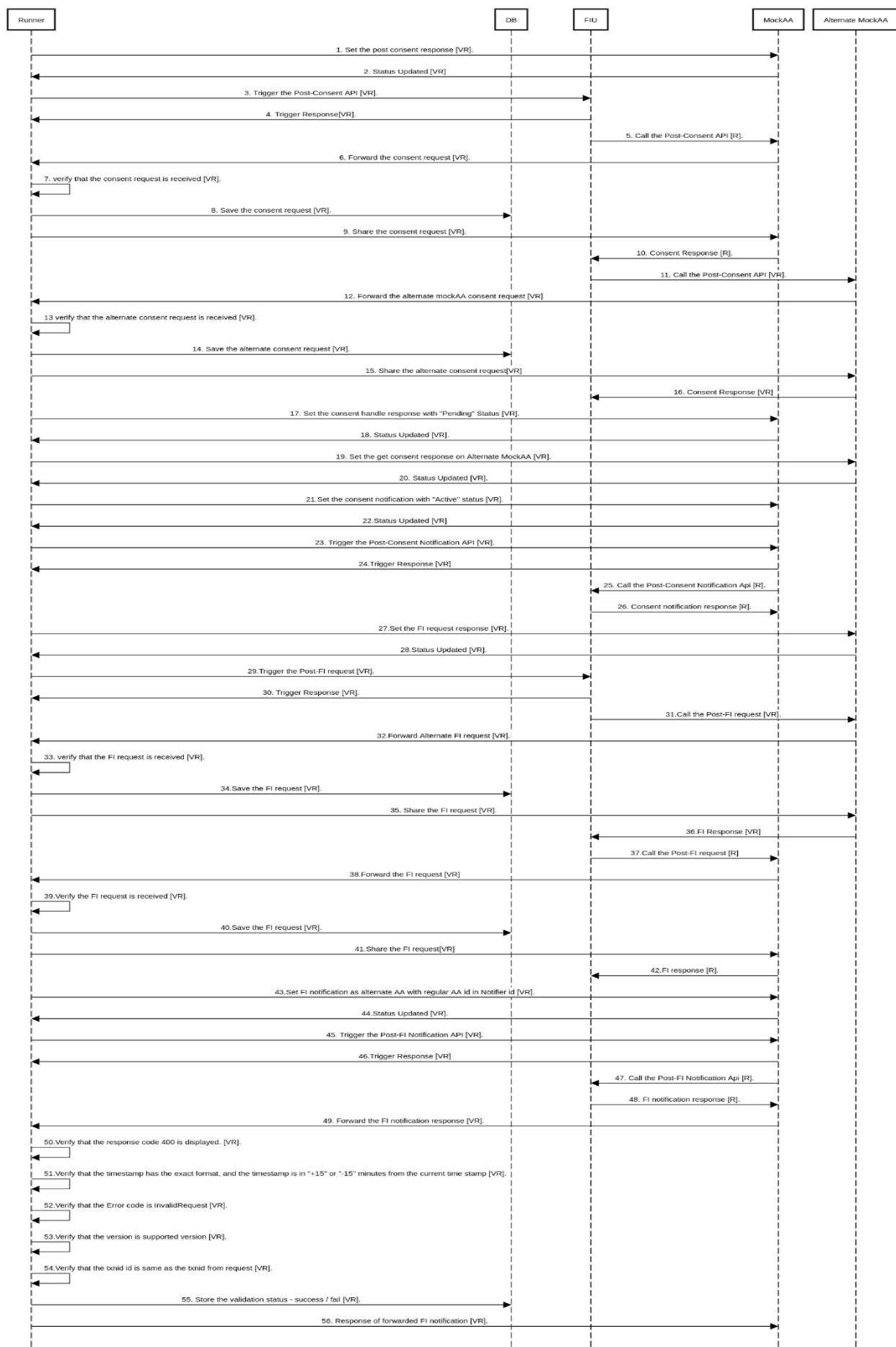
Runner<-Alternate MockAA:20. Status Updated [VR].

Runner->MockAA:21. Set the consent notification with "Active" status [VR].
Runner<-MockAA:22. Status Updated [VR]
Runner->MockAA:23. Trigger the Post-Consent Notification API [VR].
Runner<-MockAA:24. Trigger Response [VR]
FIU<-MockAA:25. Call the Post-Consent Notification Api [R].
FIU->MockAA:26. Consent notification response [R].
Runner->Alternate MockAA:27. Set the FI request response [VR].
Runner<-Alternate MockAA:28. Status Updated [VR].
Runner->FIU:29. Trigger the Post-FI request [VR].
Runner<-FIU:30. Trigger Response [VR].
FIU->Alternate MockAA:31. Call the Post-FI request [VR].
Runner<-Alternate MockAA:32. Forward Alternate FI request [VR].
Runner->Runner:33. Verify that the FI request is received [VR].
Runner -> DB:34. Save the FI request [VR].
Runner->Alternate MockAA:35. Share the FI request [VR].
FIU<-Alternate MockAA:36. FI Response [VR]
FIU->MockAA:37. Call the Post-FI request [R]
Runner<-MockAA:38. Forward the FI request [VR]
Runner->Runner:39. Verify the FI request is received [VR].
Runner->DB:40. Save the FI request [VR].
Runner->MockAA:41. Share the FI request [VR]
FIU<-MockAA:42. FI response [R].
Runner->MockAA:43. Set FI notification as alternate AA with regular AA id in Notifier id [VR].
Runner<-MockAA:44. Status Updated [VR].
Runner->MockAA:45. Trigger the Post-FI Notification API [VR].
Runner<-MockAA:46. Trigger Response [VR]
FIU<-MockAA:47. Call the Post-FI Notification Api [R].
FIU->MockAA:48. FI notification response [R].
Runner<-MockAA:49. Forward the FI notification response [VR].
Runner ->Runner:50. Verify that the response code 400 is displayed. [VR].
Runner->Runner:51. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].
Runner->Runner:52. Verify that the Error code is InvalidRequest [VR].
Runner->Runner:53. Verify that the version is supported version [VR].
Runner->Runner:54. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB:55. Store the validation status - success / fail [VR].

Runner->MockAA:56. Response of forwarded FI notification [VR].

FIU_2009_POST _FIRequest.feature Scenario 2009_1



Scenario 2009_2

title FIU_2009_POST_FIRequest.feature Scenario 2009_2

participant Runner
participant DB
participant FIU
participant MockAA
participant Alternate MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR].

Runner<-MockAA:2. Status Updated [VR]

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

FIU->Alternate MockAA:11. Call the Post-Consent API [VR].

Runner<-Alternate MockAA:12. Forward the alternate mockAA consent request [VR]

Runner->Runner:13 verify that the alternate consent request is received [VR].

Runner->DB:14. Save the alternate consent request [VR].

Runner->Alternate MockAA:15. Share the alternate consent request[VR]

FIU<-Alternate MockAA:16. Consent Response [VR]

Runner->MockAA:17. Set the consent handle response with "Pending" Status [VR].

Runner<-MockAA:18. Status Updated [VR].

Runner->Alternate MockAA:19. Set the get consent response on Alternate MockAA [VR].

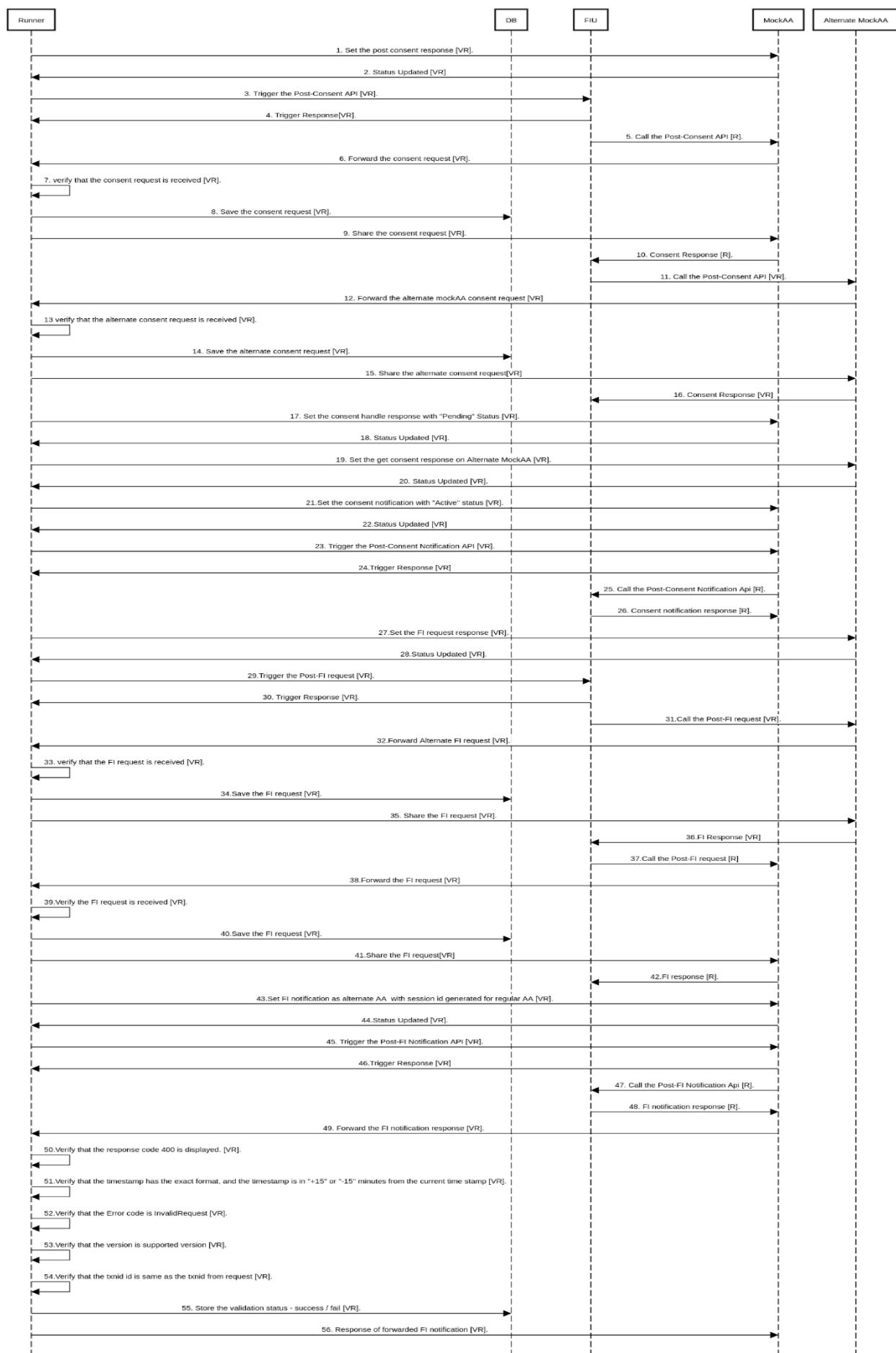
Runner<-Alternate MockAA:20. Status Updated [VR].

Runner->MockAA:21. Set the consent notification with "Active" status [VR].

Runner<-MockAA:22.Status Updated [VR]

Runner->MockAA:23. Trigger the Post-Consent Notification API [VR].
Runner<-MockAA:24. Trigger Response [VR]
FIU<-MockAA:25. Call the Post-Consent Notification Api [R].
FIU->MockAA:26. Consent notification response [R].
Runner->Alternate MockAA:27. Set the FI request response [VR].
Runner<-Alternate MockAA:28. Status Updated [VR].
Runner->FIU:29. Trigger the Post-FI request [VR].
Runner<-FIU:30. Trigger Response [VR].
FIU->Alternate MockAA:31. Call the Post-FI request [VR].
Runner<-Alternate MockAA:32. Forward Alternate FI request [VR].
Runner->Runner:33. Verify that the FI request is received [VR].
Runner -> DB:34. Save the FI request [VR].
Runner->Alternate MockAA:35. Share the FI request [VR].
FIU<-Alternate MockAA:36. FI Response [VR]
FIU->MockAA:37. Call the Post-FI request [R]
Runner<-MockAA:38. Forward the FI request [VR]
Runner->Runner:39. Verify the FI request is received [VR].
Runner->DB:40. Save the FI request [VR].
Runner->MockAA:41. Share the FI request [VR]
FIU<-MockAA:42. FI response [R].
Runner->MockAA:43. Set FI notification as alternate AA with session id generated for regular AA [VR].
Runner<-MockAA:44. Status Updated [VR].
Runner->MockAA:45. Trigger the Post-FI Notification API [VR].
Runner<-MockAA:46. Trigger Response [VR]
FIU<-MockAA:47. Call the Post-FI Notification Api [R].
FIU->MockAA:48. FI notification response [R].
Runner<-MockAA:49. Forward the FI notification response [VR].
Runner ->Runner:50. Verify that the response code 400 is displayed. [VR].
Runner->Runner:51. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].
Runner->Runner:52. Verify that the Error code is InvalidRequest [VR].
Runner->Runner:53. Verify that the version is supported version [VR].
Runner->Runner:54. Verify that the txnid id is same as the txnid from request [VR].
Runner -> DB:55. Store the validation status - success / fail [VR].
Runner->MockAA:56. Response of forwarded FI notification [VR].

FIU_2009_POST _FIRequest.feature Scenario 2009_2



Scenario 2009_3

title FIU_2009_POST_FIRequest.feature Scenario 2009_3

participant Runner
participant DB
participant FIU
participant MockAA
participant Alternate MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR].

Runner<-MockAA:2. Status Updated [VR]

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

FIU->Alternate MockAA:11. Call the Post-Consent API [VR].

Runner<-Alternate MockAA:12. Forward the alternate mockAA consent request [VR]

Runner->Runner:13 verify that the alternate consent request is received [VR].

Runner->DB:14. Save the alternate consent request [VR].

Runner->Alternate MockAA:15. Share the alternate consent request[VR]

FIU<-Alternate MockAA:16. Consent Response [VR]

Runner->MockAA:17. Set the consent handle response with "Pending" Status [VR].

Runner<-MockAA:18. Status Updated [VR].

Runner->Alternate MockAA:19. Set the get consent response on Alternate MockAA [VR].

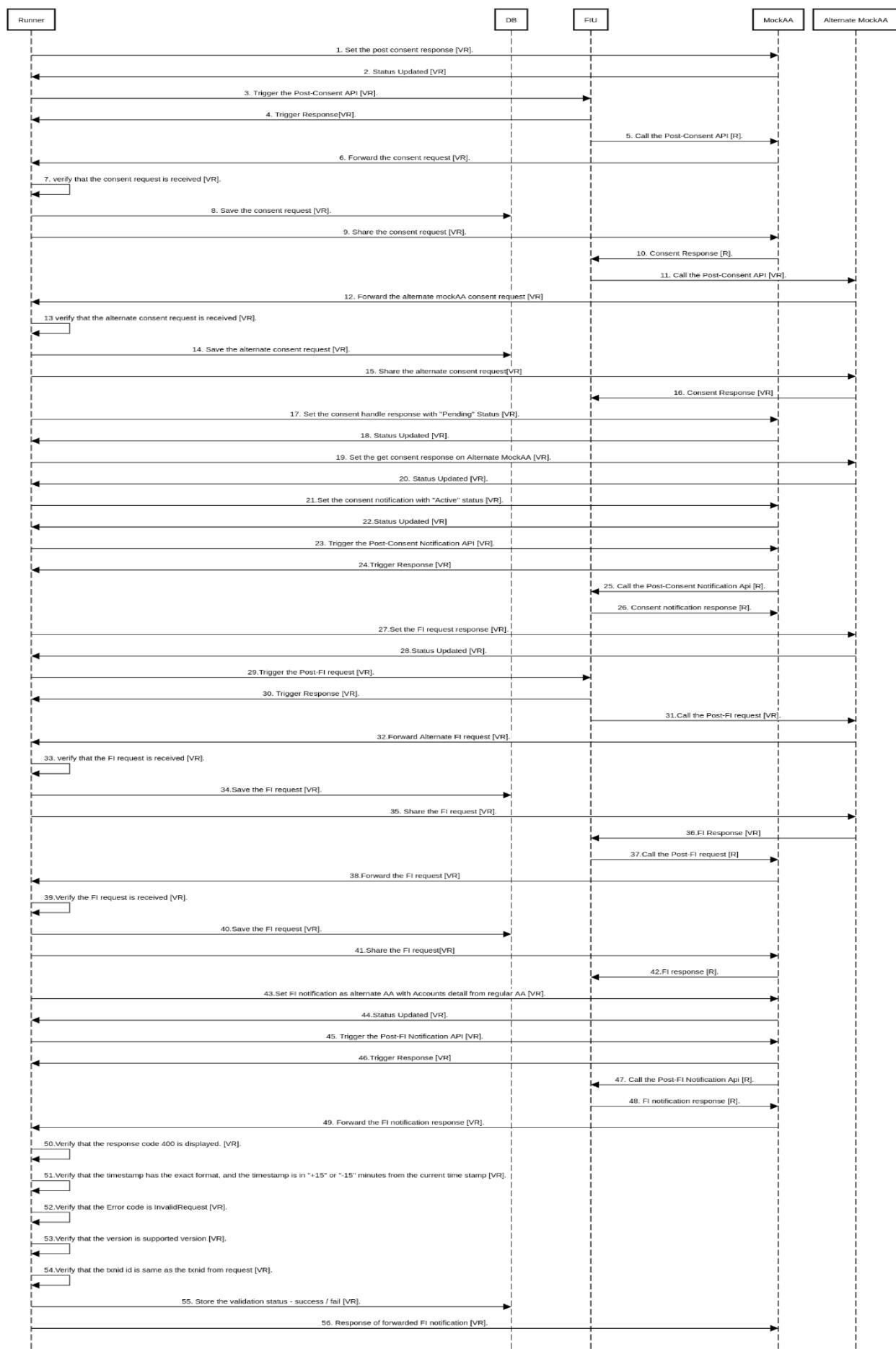
Runner<-Alternate MockAA:20. Status Updated [VR].

Runner->MockAA:21. Set the consent notification with "Active" status [VR].

Runner<-MockAA:22.Status Updated [VR]

Runner->MockAA:23. Trigger the Post-Consent Notification API [VR].
Runner<-MockAA:24. Trigger Response [VR]
FIU<-MockAA:25. Call the Post-Consent Notification Api [R].
FIU->MockAA:26. Consent notification response [R].
Runner->Alternate MockAA:27. Set the FI request response [VR].
Runner<-Alternate MockAA:28. Status Updated [VR].
Runner->FIU:29. Trigger the Post-FI request [VR].
Runner<-FIU:30. Trigger Response [VR].
FIU->Alternate MockAA:31. Call the Post-FI request [VR].
Runner<-Alternate MockAA:32. Forward Alternate FI request [VR].
Runner->Runner:33. Verify that the FI request is received [VR].
Runner -> DB:34. Save the FI request [VR].
Runner->Alternate MockAA:35. Share the FI request [VR].
FIU<-Alternate MockAA:36. FI Response [VR]
FIU->MockAA:37. Call the Post-FI request [R]
Runner<-MockAA:38. Forward the FI request [VR]
Runner->Runner:39. Verify the FI request is received [VR].
Runner->DB:40. Save the FI request [VR].
Runner->MockAA:41. Share the FI request [VR]
FIU<-MockAA:42. FI response [R].
Runner->MockAA:43. Set FI notification as alternate AA with Accounts detail from regular AA [VR].
Runner<-MockAA:44. Status Updated [VR].
Runner->MockAA:45. Trigger the Post-FI Notification API [VR].
Runner<-MockAA:46. Trigger Response [VR]
FIU<-MockAA:47. Call the Post-FI Notification Api [R].
FIU->MockAA:48. FI notification response [R].
Runner<-MockAA:49. Forward the FI notification response [VR].
Runner ->Runner:50. Verify that the response code 400 is displayed. [VR].
Runner->Runner:51. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].
Runner->Runner:52. Verify that the Error code is InvalidRequest [VR].
Runner->Runner:53. Verify that the version is supported version [VR].
Runner->Runner:54. Verify that the txnid id is same as the txnid from request [VR].
Runner -> DB:55. Store the validation status - success / fail [VR].
Runner->MockAA:56. Response of forwarded FI notification [VR].

FIU_2009_POST _FIRequest.feature Scenario 2009_3



FIU_2010_GET_FIFetchSessionId.feature

Scenario 2010_1

title FIU_2010_GET_FIFetchSessionId.feature Scenario 2010_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->FIU:1. Trigger the Post-FI request [VR].

Runner<-FIU:2. Trigger Response [VR].

FIU->MockAA:3. Call the Post-FI request [R].

MockAA ->Runner:4. Forward the FI request [VR].

Runner->Runner:5. verify that the FI request is received [VR].

Runner -> DB:6. Save the FI request [VR].

Runner->MockAA:7. Share the FI request [VR].

FIU<-MockAA:8. FI Response [R].

Runner->MockAA:9. Set FI notification status as "Ready" [VR].

Runner<-MockAA:10. Status Updated [VR]

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12. Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner->FIU:15. Trigger the Get-FI Fetch API [VR]

Runner<-FIU: 16. Trigger response [VR]

FIU->MockAA:17. Call the Get-FI Fetch Api [R]

Runner<-MockAA: 18. Forward the FI Fetch request [VR]

Runner->Runner:19. Verify that the session Id received is the valid session id [VR].

Runner->Runner:20. Verify that if the FIP id is present, it is a valid FIP id [VR].

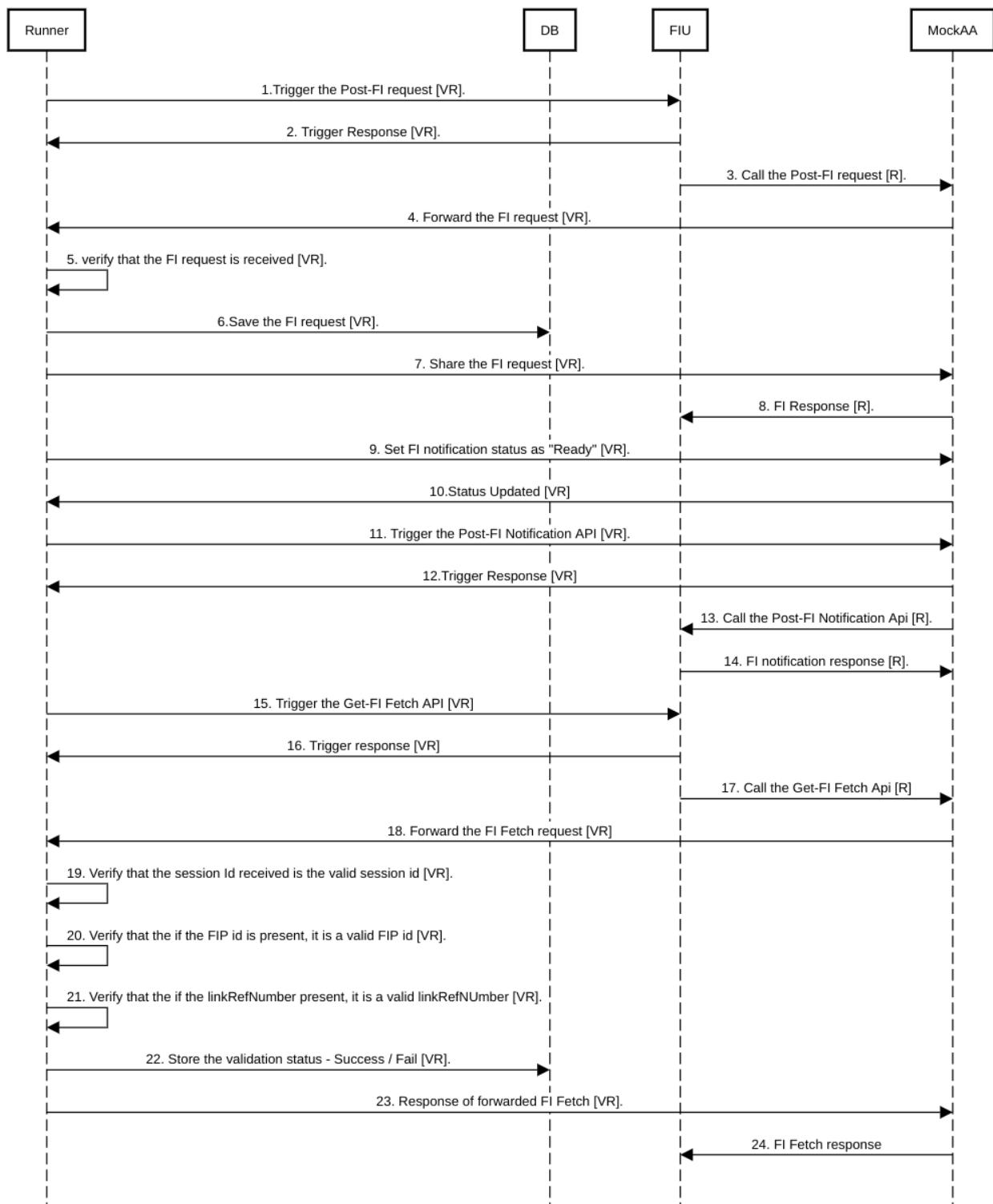
Runner->Runner:21. Verify that if the linkRefNumber present, it is a valid linkRefNUmber [VR].

Runner -> DB: 22. Store the validation status - Success / Fail [VR].

Runner->MockAA:23. Response of forwarded FI Fetch [VR].

FIU<-MockAA:24. FI Fetch response [R]

FIU_2010_GET_FIFetchSessionId.feature Scenario 2010_1



FIU_2011_POST_FINotification.feature

Scenario 2011_1

title FIU_2011_POST_FINotification.feature Scenario 2011_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->FIU:1.Trigger the Post-FI request [VR].

Runner<-FIU:2. Trigger Response [VR].

FIU->MockAA:3. Call the Post-FI request [R].

MockAA ->Runner:4. Forward the FI request [VR].

Runner->Runner:5. verify that the FI request is received [VR].

Runner -> DB:6.Save the FI request [VR].

Runner->MockAA:7. Share the FI request [VR].

FIU<-MockAA:8. FI Response [R].

Runner->MockAA:9. Set FI notification sessionStatus as "Expired" [VR].

Runner<-MockAA:10.Status Updated [VR]

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12.Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

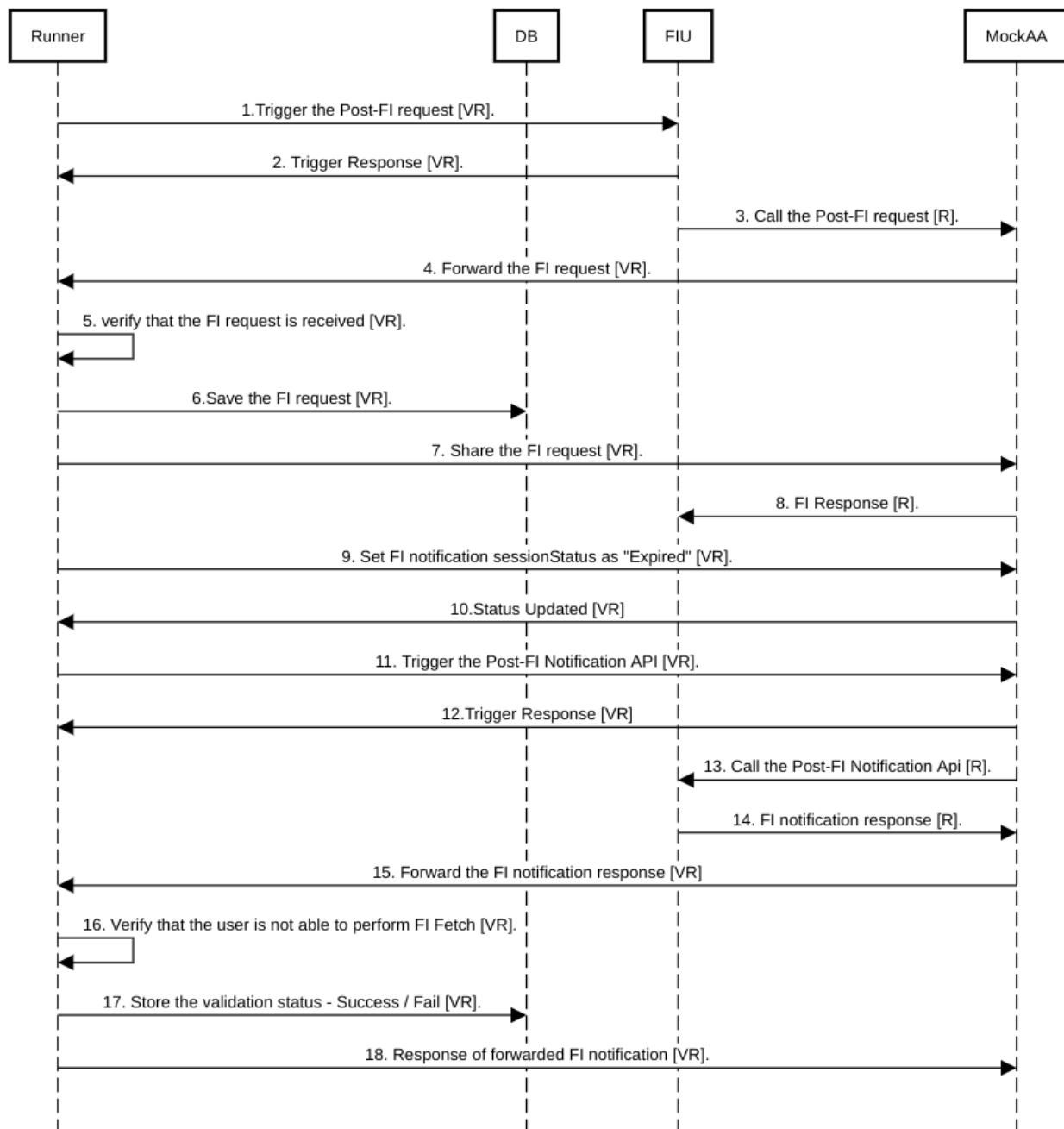
Runner<-MockAA:15. Forward the FI notification response [VR]

Runner->Runner:16. Verify that the user is not able to perform FI Fetch [VR].

Runner -> DB: 17. Store the validation status - Success / Fail [VR].

Runner->MockAA:18. Response of forwarded FI notification [VR].

FIU_2011_POST _FINotification.feature Scenario 2011_1



FIU_2012_POST_FINotification.feature

Scenario 2012_1

title FIU_2012_POST_FINotification.feature Scenario 2012_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->FIU:1.Trigger the Post-FI request [VR].

Runner<-FIU:2. Trigger Response [VR].

FIU->MockAA:3. Call the Post-FI request [R].

MockAA ->Runner:4. Forward the FI request [VR].

Runner->Runner:5. verify that the FI request is received [VR].

Runner -> DB:6.Save the FI request [VR].

Runner->MockAA:7. Share the FI request [VR].

FIU<-MockAA:8. FI Response [R].

Runner->MockAA:9. Set FI notification with Notifier type as FIP [VR].

Runner<-MockAA:10.Status Updated [VR]

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12.Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR]

Runner->Runner:16. Verify that the response code 400 is displayed [VR].

Runner->Runner:17. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:18. Verify that the Error code is InvalidRequest [VR].

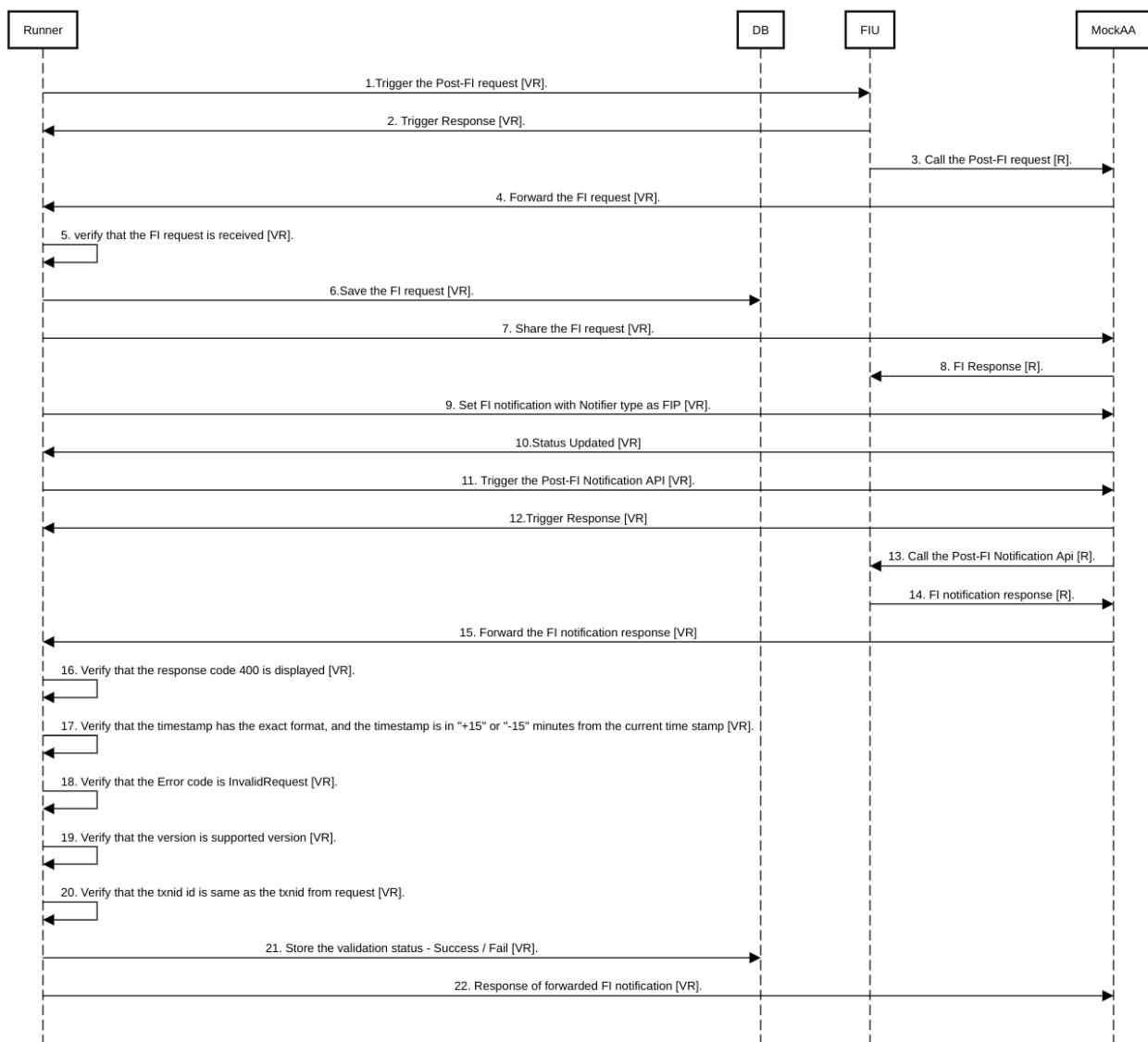
Runner->Runner:19. Verify that the version is supported version [VR].

Runner->Runner:20. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2012_POST_FINotification.feature Scenario 2012_1



Scenario 2012_2

title FIU_2012_POST_FINotification.feature Scenario 2012_2

participant Runner

participant DB

participant FIU

participant MockAA

Runner->FIU:1. Trigger the Post-FI request [VR].

Runner<-FIU:2. Trigger Response [VR].

FIU->MockAA:3. Call the Post-FI request [R].

MockAA ->Runner:4. Forward the FI request [VR].

Runner->Runner:5. verify that the FI request is received [VR].

Runner -> DB:6. Save the FI request [VR].

Runner->MockAA:7. Share the FI request [VR].

FIU<-MockAA:8. FI Response [R].

Runner->MockAA:9. Set FI notification with Notifier type as FIU [VR].

Runner<-MockAA:10. Status Updated [VR]

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12. Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR]

Runner->Runner:16. Verify that the response code 400 is displayed [VR].

Runner->Runner:17. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:18. Verify that the Error code is InvalidRequest [VR].

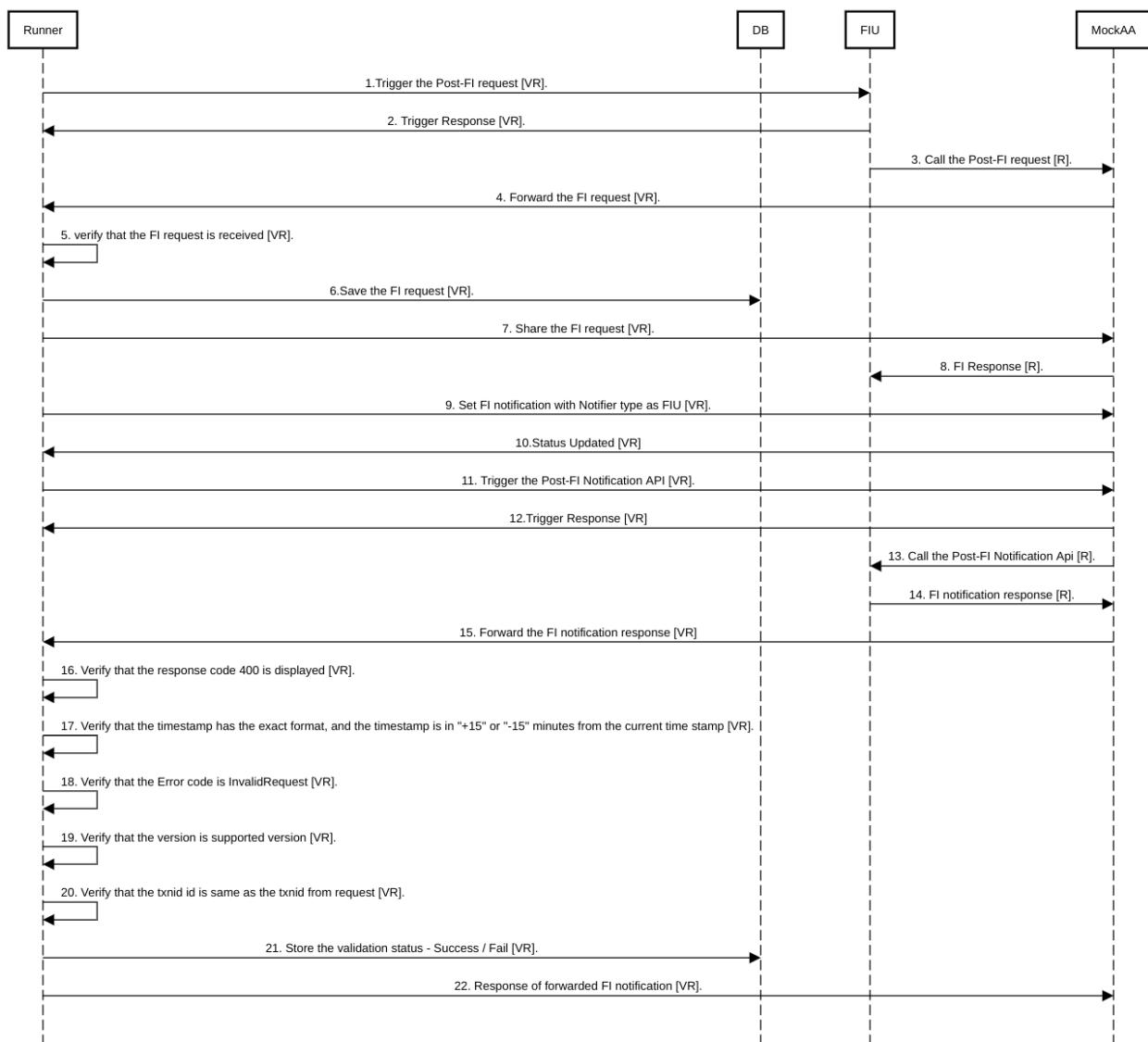
Runner->Runner:19. Verify that the version is supported version [VR].

Runner->Runner:20. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2012_POST_FINotification.feature Scenario 2012_2



FIU_2015_POST _FINotification.feature

Scenario 2015_1

title FIU_2015_POST _FINotification.feature Scenario 2015_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->FIU:1.Trigger the Post-FI request [VR].

Runner<-FIU:2. Trigger Response [VR].

FIU->MockAA:3. Call the Post-FI request [R].

MockAA ->Runner:4. Forward the FI request [VR].

Runner->Runner:5. verify that the FI request is received [VR].

Runner -> DB:6.Save the FI request [VR].

Runner->MockAA:7. Share the FI request [VR].

FIU<-MockAA:8. FI Response [R].

Runner->MockAA:9. Set FI notification sessionStatus as "FAILED" [VR].

Runner<-MockAA:10.Status Updated [VR]

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12.Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

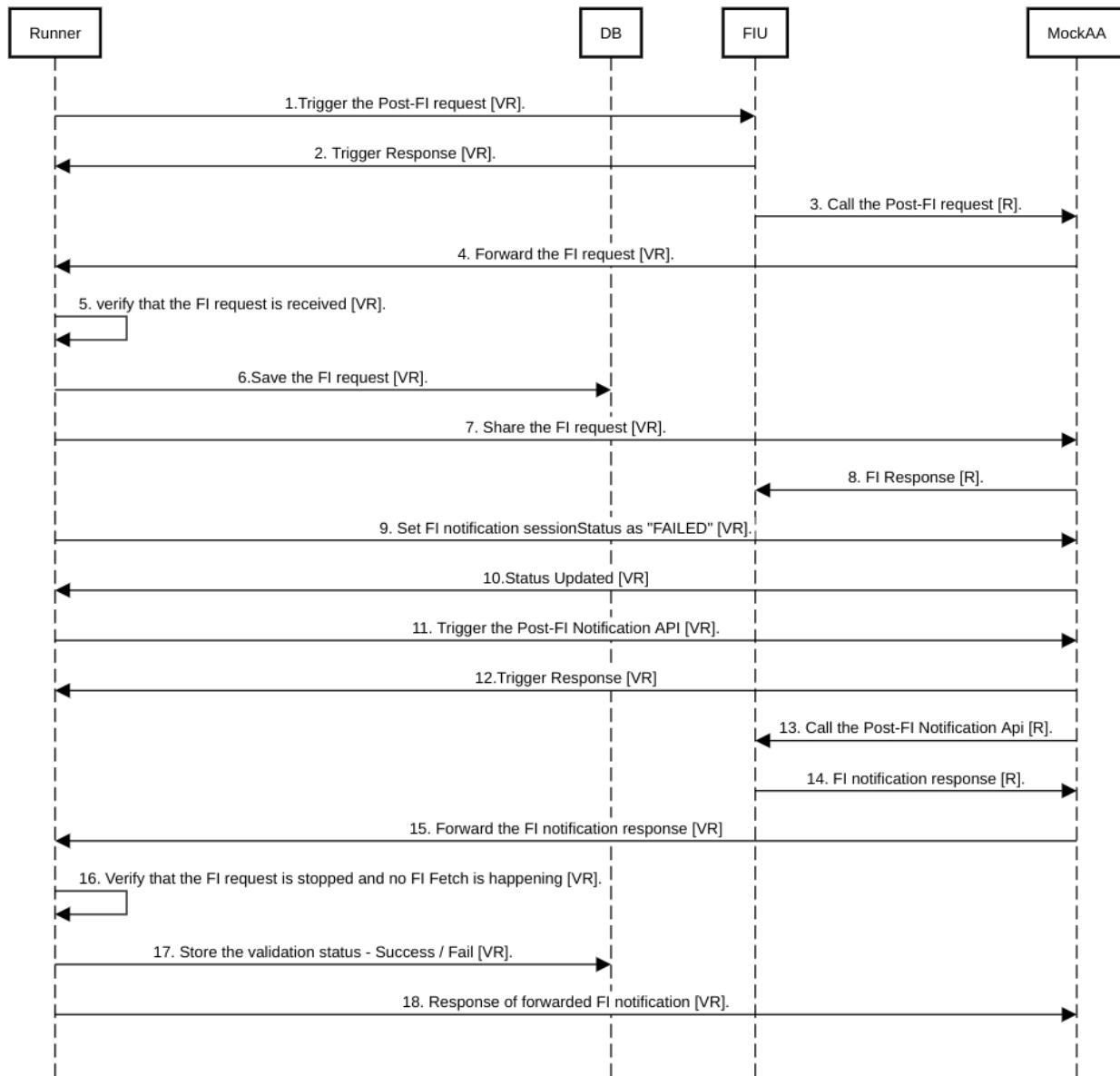
Runner<-MockAA:15. Forward the FI notification response [VR]

Runner->Runner:16. Verify that the FI request is stopped and no FI Fetch is happening [VR].

Runner -> DB:17. Store the validation status - Success / Fail [VR].

Runner->MockAA:18. Response of forwarded FI notification [VR].

FIU_2015_POST _FINotification.feature Scenario 2015_1



FIU_2016_POST_FINotification.feature

Scenario 2016_1

title FIU_2016_POST_FINotification.feature Scenario 2016_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA:1. Set FI response with null value in one required field [VR].

Runner<-MockAA:2. Status Updated [VR]

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. Verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set FI notification with "READY" status [VR].

Runner<-MockAA:12. Status Updated [VR]

Runner->MockAA:13. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:14. Trigger Response [VR]

FIU<-MockAA:15. Call the Post-FI Notification Api [R].

FIU->MockAA:16. FI notification response [R].

Runner<-MockAA:17. Forward the FI notification response [VR].

Runner->Runner:18. Verify that the response code 400 is displayed [VR].

Runner->Runner:19. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:20. Verify that the Error code is InvalidSessionId [VR].

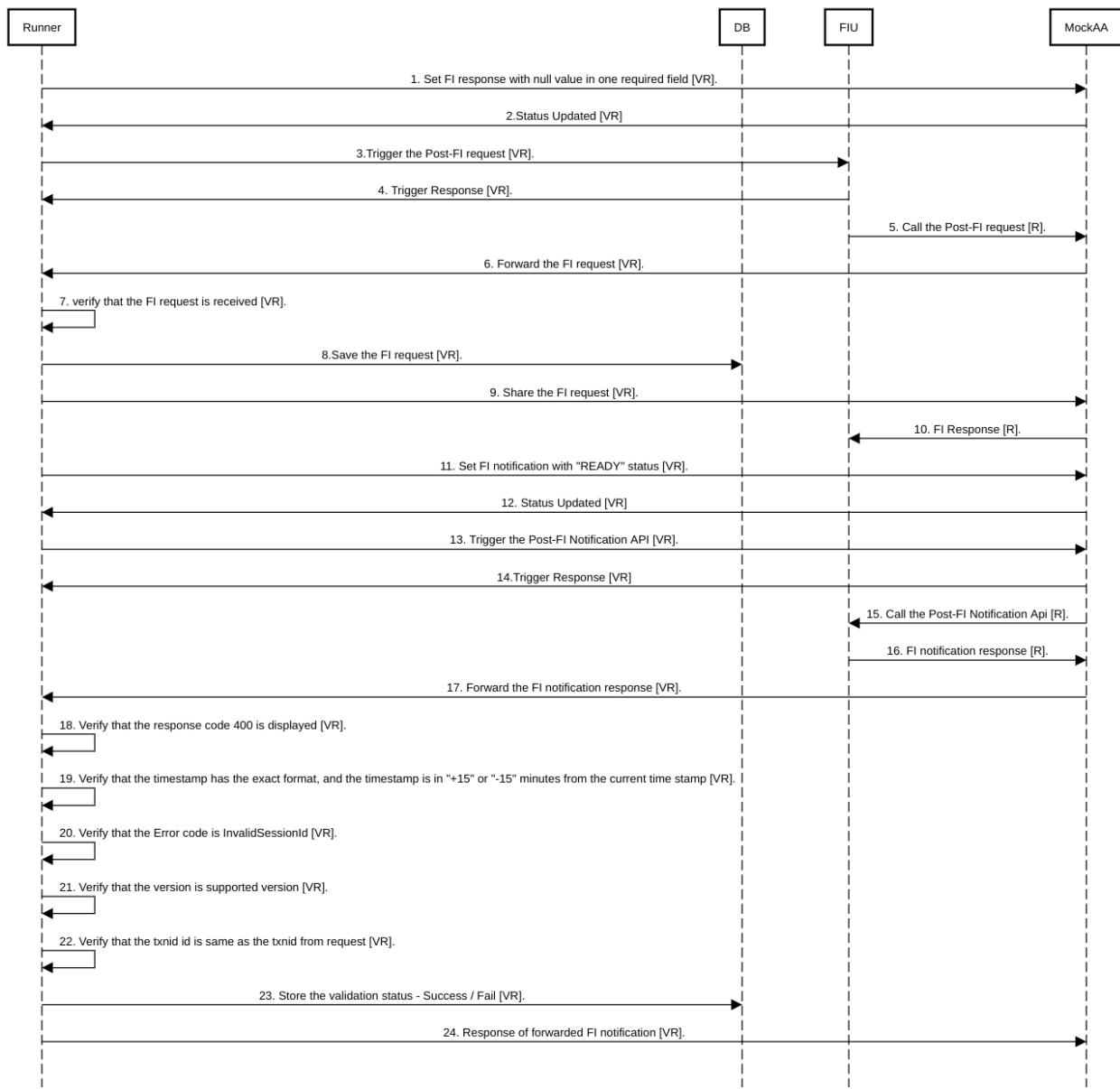
Runner->Runner:21. Verify that the version is supported version [VR].

Runner->Runner:22. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB:23. Store the validation status - Success / Fail [VR].

Runner->MockAA:24. Response of forwarded FI notification [VR].

FIU_2016_POST _FINotification.feature Scenario 2016_1



Scenario 2016_2

title FIU_2016_POST_FINotification.feature Scenario 2016_2

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA:1. Set FI response with empty string in one required field [VR].

Runner<-MockAA:2. Status Updated [VR]

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set FI notification with "READY" status [VR].

Runner<-MockAA:12. Status Updated [VR]

Runner->MockAA:13. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:14. Trigger Response [VR]

FIU<-MockAA:15. Call the Post-FI Notification Api [R].

FIU->MockAA:16. FI notification response [R].

Runner<-MockAA:17. Forward the FI notification response [VR].

Runner->Runner:18. Verify that the response code 400 is displayed [VR].

Runner->Runner:19. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:20. Verify that the Error code is InvalidSessionId [VR].

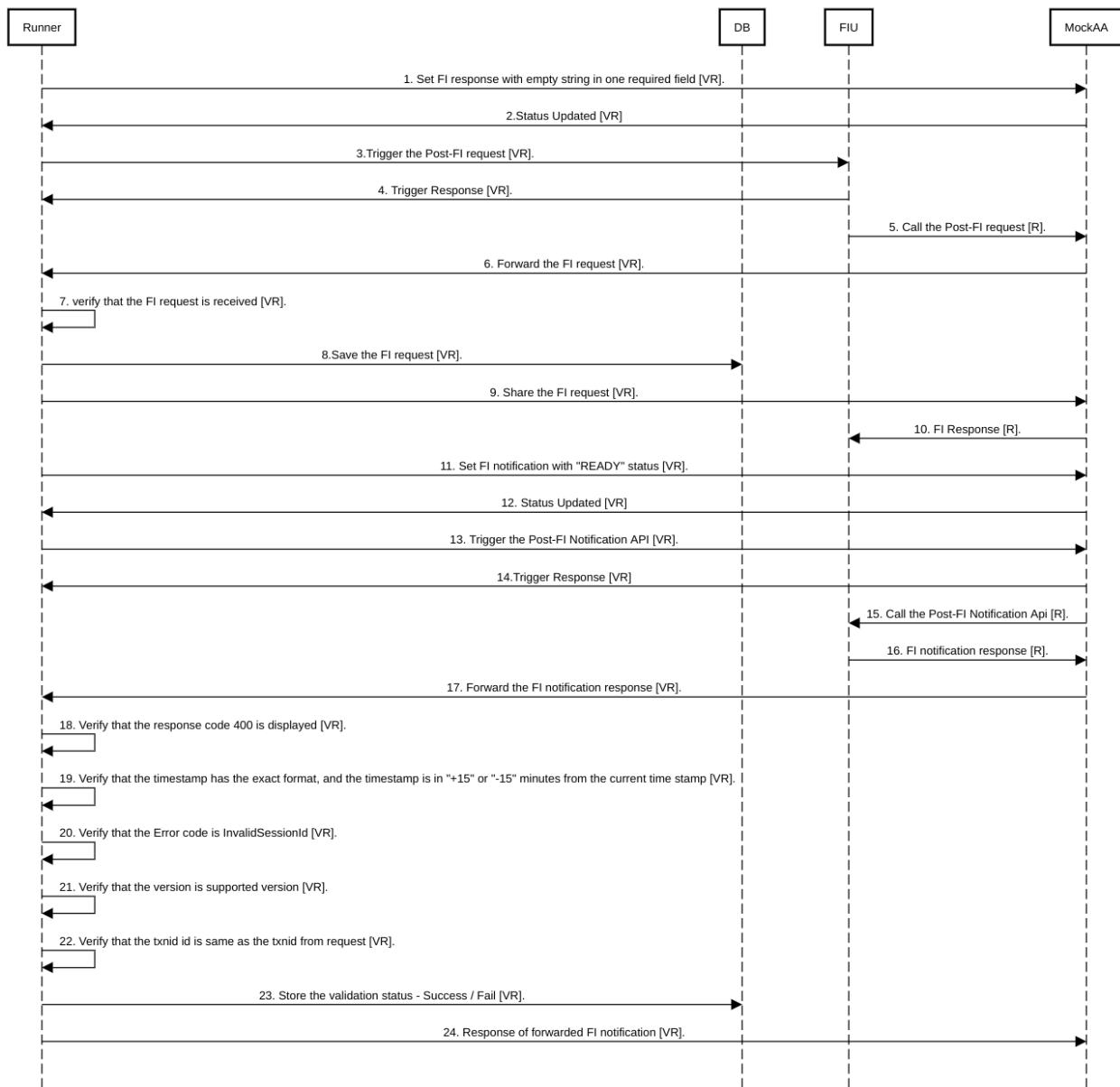
Runner->Runner:21. Verify that the version is supported version [VR].

Runner->Runner:22. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB:23. Store the validation status - Success / Fail [VR].

Runner->MockAA:24. Response of forwarded FI notification [VR].

FIU_2016_POST _FINotification.feature Scenario 2016_2



Scenario 2016_3

title FIU_2016_POST_FINotification.feature Scenario 2016_3

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA:1. Set FI response with o(Zero) value in one required field [VR].

Runner<-MockAA:2. Status Updated [VR]

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set FI notification with "READY" status [VR].

Runner<-MockAA:12. Status Updated [VR]

Runner->MockAA:13. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:14. Trigger Response [VR]

FIU<-MockAA:15. Call the Post-FI Notification Api [R].

FIU->MockAA:16. FI notification response [R].

Runner<-MockAA:17. Forward the FI notification response [VR].

Runner->Runner:18. Verify that the response code 400 is displayed [VR].

Runner->Runner:19. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:20. Verify that the Error code is InvalidSessionId [VR].

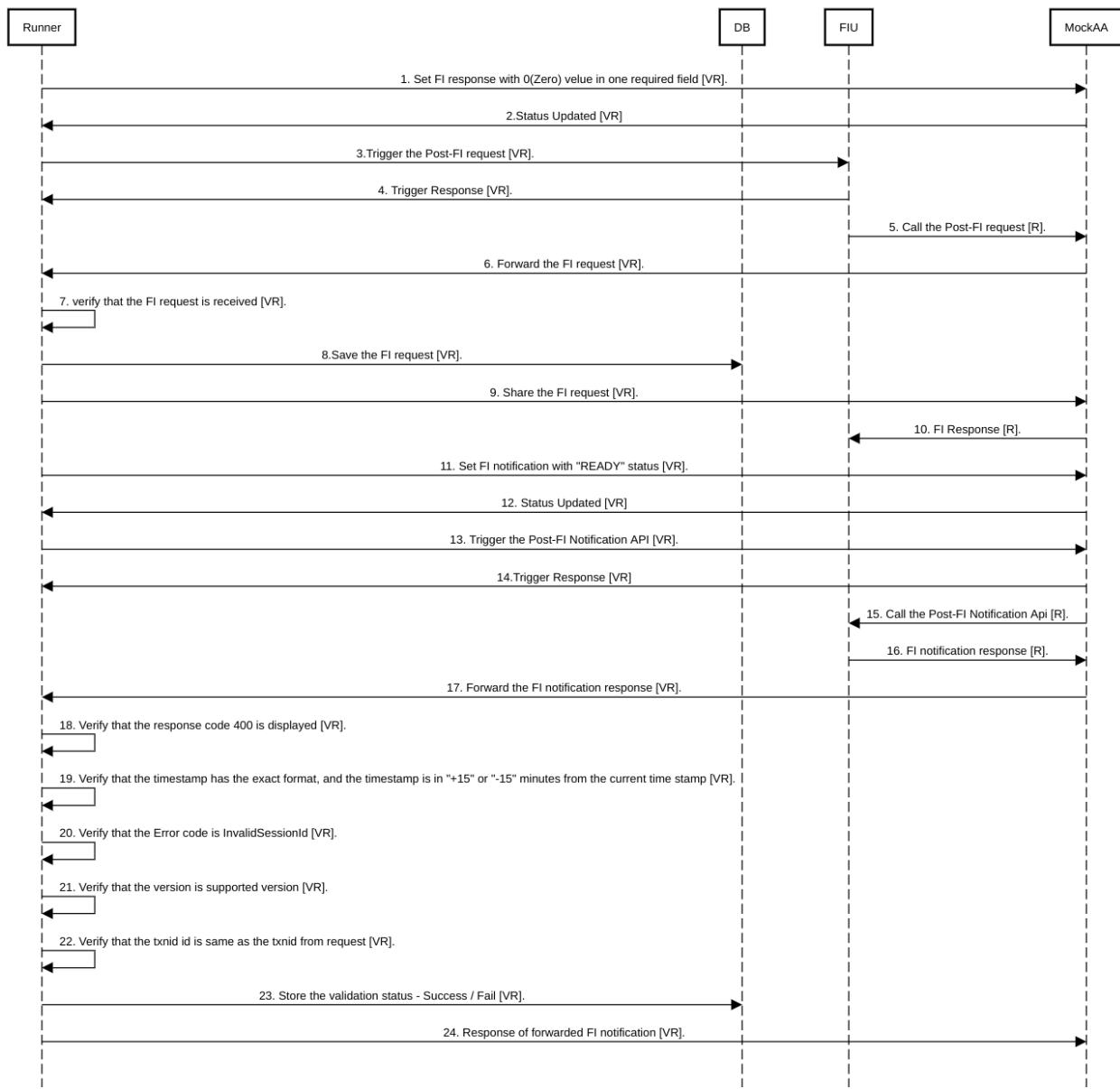
Runner->Runner:21. Verify that the version is supported version [VR].

Runner->Runner:22. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB:23. Store the validation status - Success / Fail [VR].

Runner->MockAA:24. Response of forwarded FI notification [VR].

FIU_2016_POST _FINotification.feature Scenario 2016_3



Scenario 2016_4

title FIU_2016_POST_FINotification.feature Scenario 2016_4

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA:1. Set FI response with one enum field set to non enum value by adding an extra character at the end of last enum and removing the last character at the end of first enum [VR].

Runner<-MockAA:2. Status Updated [VR]

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set FI notification with "READY" status [VR].

Runner<-MockAA:12. Status Updated [VR]

Runner->MockAA:13. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:14. Trigger Response [VR]

FIU<-MockAA:15. Call the Post-FI Notification Api [R].

FIU->MockAA:16. FI notification response [R].

Runner<-MockAA:17. Forward the FI notification response [VR].

Runner->Runner:18. Verify that the response code 400 is displayed [VR].

Runner->Runner:19. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:20. Verify that the Error code is InvalidSessionId [VR].

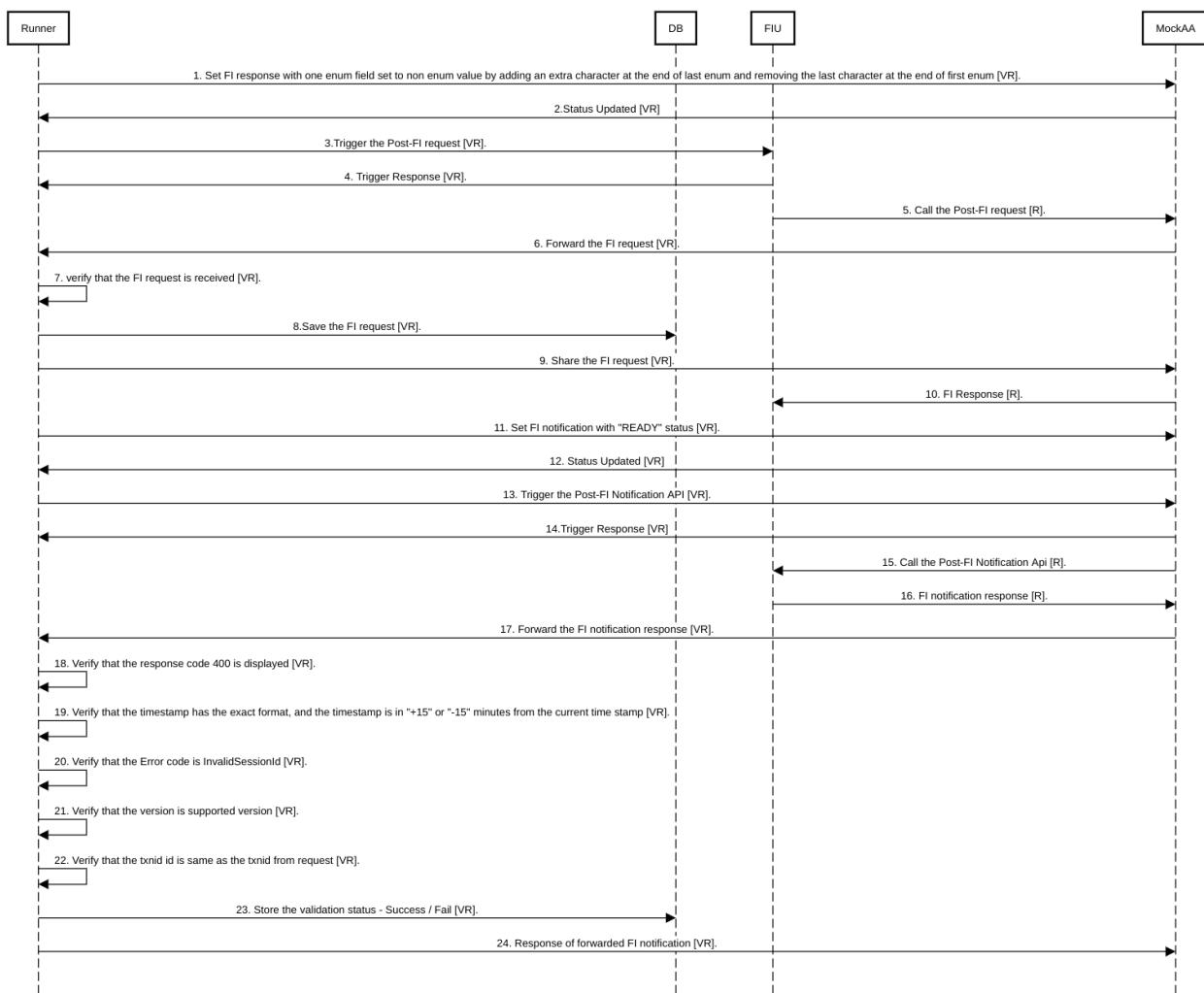
Runner->Runner:21. Verify that the version is supported version [VR].

Runner->Runner:22. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB:23. Store the validation status - Success / Fail [VR].

Runner->MockAA:24. Response of forwarded FI notification [VR].

FIU_2016_POST _FINotification.feature Scenario 2016_4



Scenario 2016_5

title FIU_2016_POST_FINotification.feature Scenario 2016_5

participant Runner
participant DB
participant FIU
participant MockAA

Runner->MockAA:1. Set FI response with empty array in one array field [VR].

Runner<-MockAA:2. Status Updated [VR]

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set FI notification with "READY" status [VR].

Runner<-MockAA:12. Status Updated [VR]

Runner->MockAA:13. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:14. Trigger Response [VR]

FIU<-MockAA:15. Call the Post-FI Notification Api [R].

FIU->MockAA:16. FI notification response [R].

Runner<-MockAA:17. Forward the FI notification response [VR].

Runner->Runner:18. Verify that the response code 400 is displayed [VR].

Runner->Runner:19. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:20. Verify that the Error code is InvalidSessionId [VR].

Runner->Runner:21. Verify that the version is supported version [VR].

Runner->Runner:22. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB:23. Store the validation status - Success / Fail [VR].

Runner->MockAA:24. Response of forwarded FI notification [VR].

FIU_2016_POST_FINotification.feature Scenario 2016_5



Scenario 2016_6

title FIU_2016_POST_FINotification.feature Scenario 2016_6

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA:1. Set FI response with empty object in one object field [VR].

Runner<-MockAA:2. Status Updated [VR]

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. Verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set FI notification with "READY" status [VR].

Runner<-MockAA:12. Status Updated [VR]

Runner->MockAA:13. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:14. Trigger Response [VR]

FIU<-MockAA:15. Call the Post-FI Notification Api [R].

FIU->MockAA:16. FI notification response [R].

Runner<-MockAA:17. Forward the FI notification response [VR].

Runner->Runner:18. Verify that the response code 400 is displayed [VR].

Runner->Runner:19. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:20. Verify that the Error code is InvalidSessionId [VR].

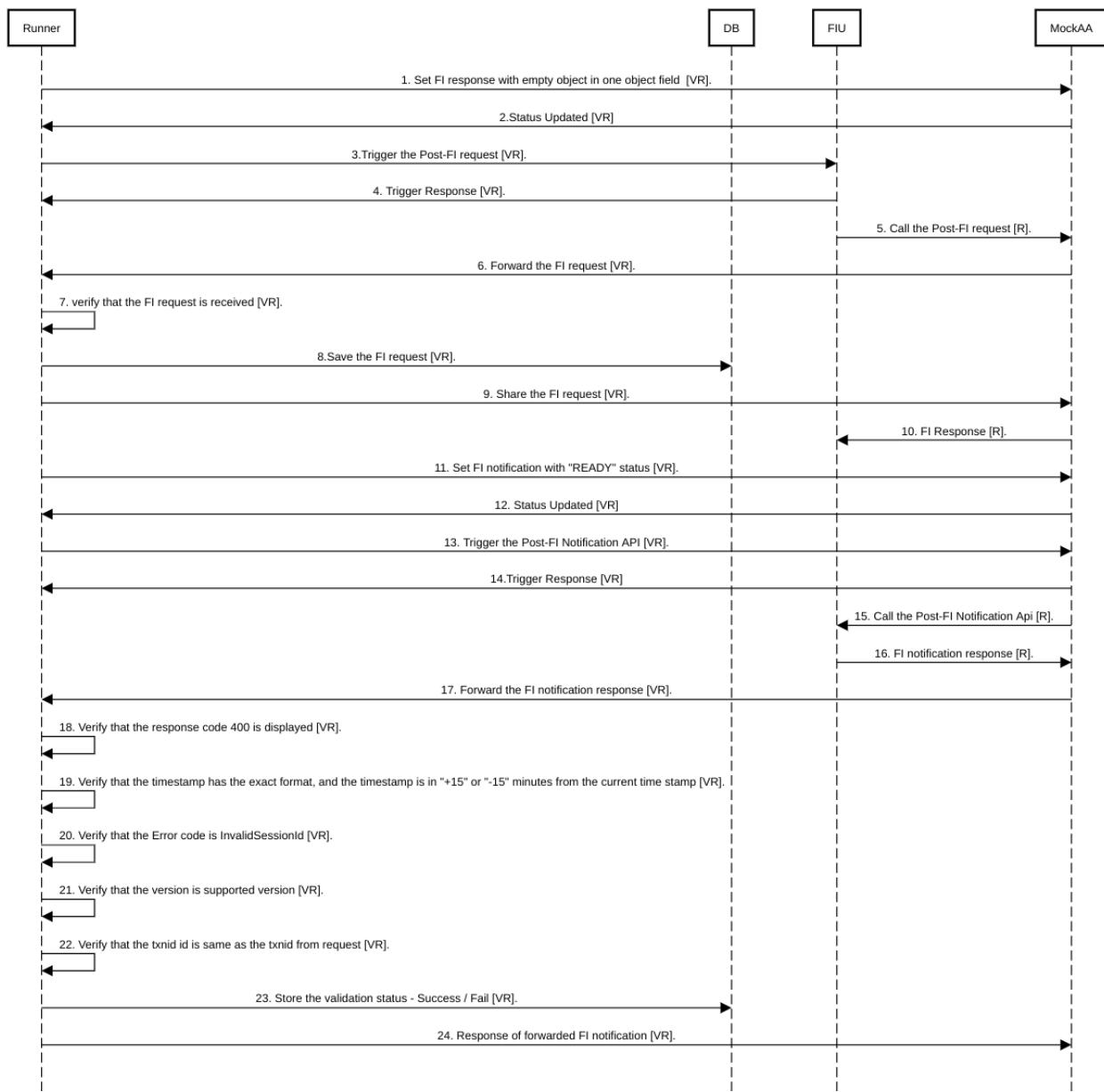
Runner->Runner:21. Verify that the version is supported version [VR].

Runner->Runner:22. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB:23. Store the validation status - Success / Fail [VR].

Runner->MockAA:24. Response of forwarded FI notification [VR].

FIU_2016_POST_FINotification.feature Scenario 2016_6



FIU_2017_POST_FINotification.feature

Scenario 2017_1

title FIU_2017_POST_FINotification.feature Scenario 2017_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA:1. Set FI response with incorrect version number [VR].

Runner<-MockAA:2. Status Updated [VR]

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. Verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12. Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification API [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR].

Runner->Runner:16. Verify that the response code 400 is displayed [VR].

Runner->Runner:17. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:18. Verify that the Error code is InvalidSessionId [VR].

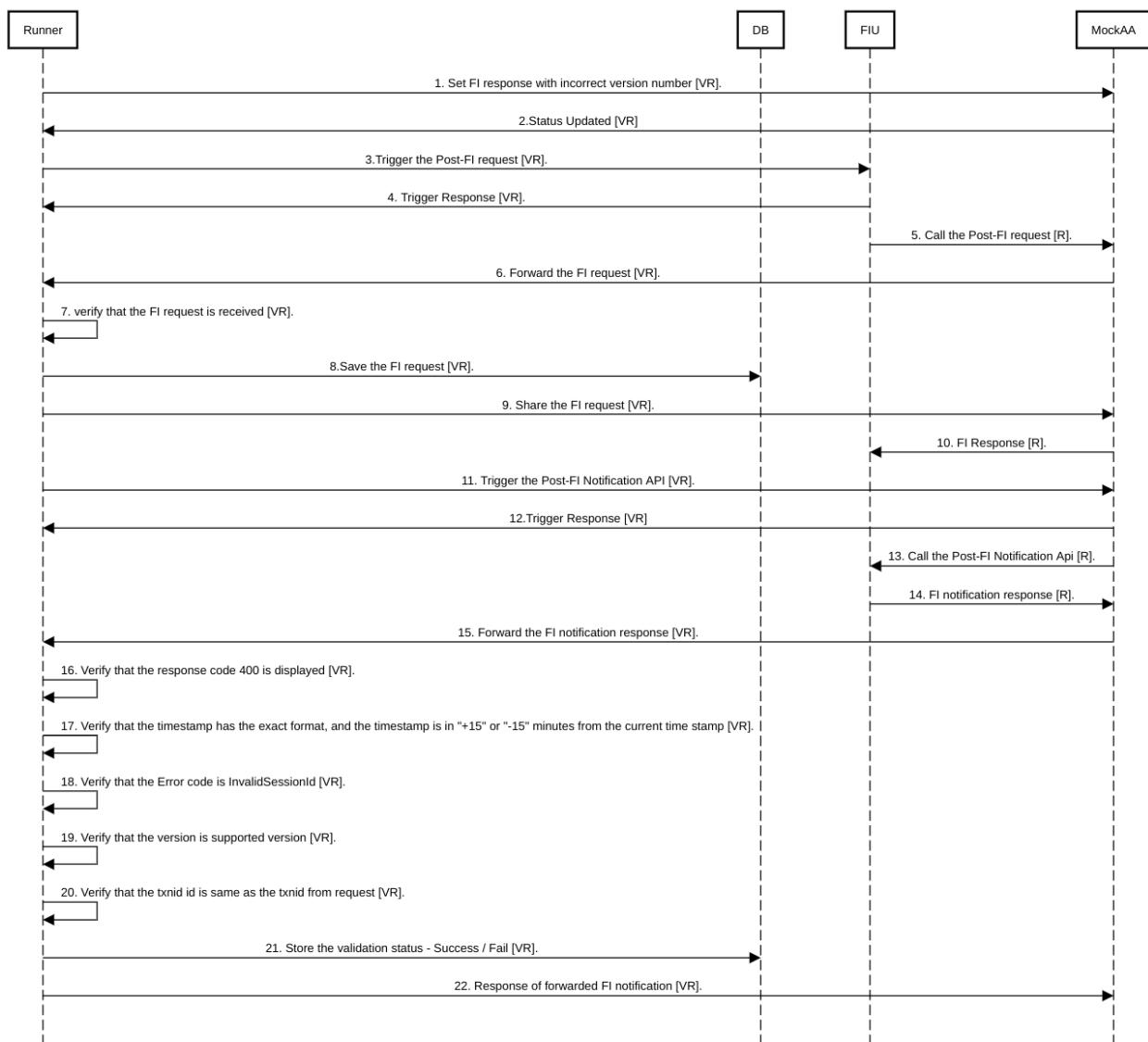
Runner->Runner:19. Verify that the version is supported version [VR].

Runner->Runner:20. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2017_POST_FINotification.feature Scenario 2017_1



Scenario 2017_2

title FIU_2017_POST_FINotification.feature Scenario 2017_2

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA:1. Set FI response by adding an extra digit at the end of the correct version number [VR].

Runner<-MockAA:2. Status Updated [VR]

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12. Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR].

Runner->Runner:16. Verify that the response code 400 is displayed [VR].

Runner->Runner:17. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current timestamp [VR].

Runner->Runner:18. Verify that the Error code is InvalidSessionId [VR].

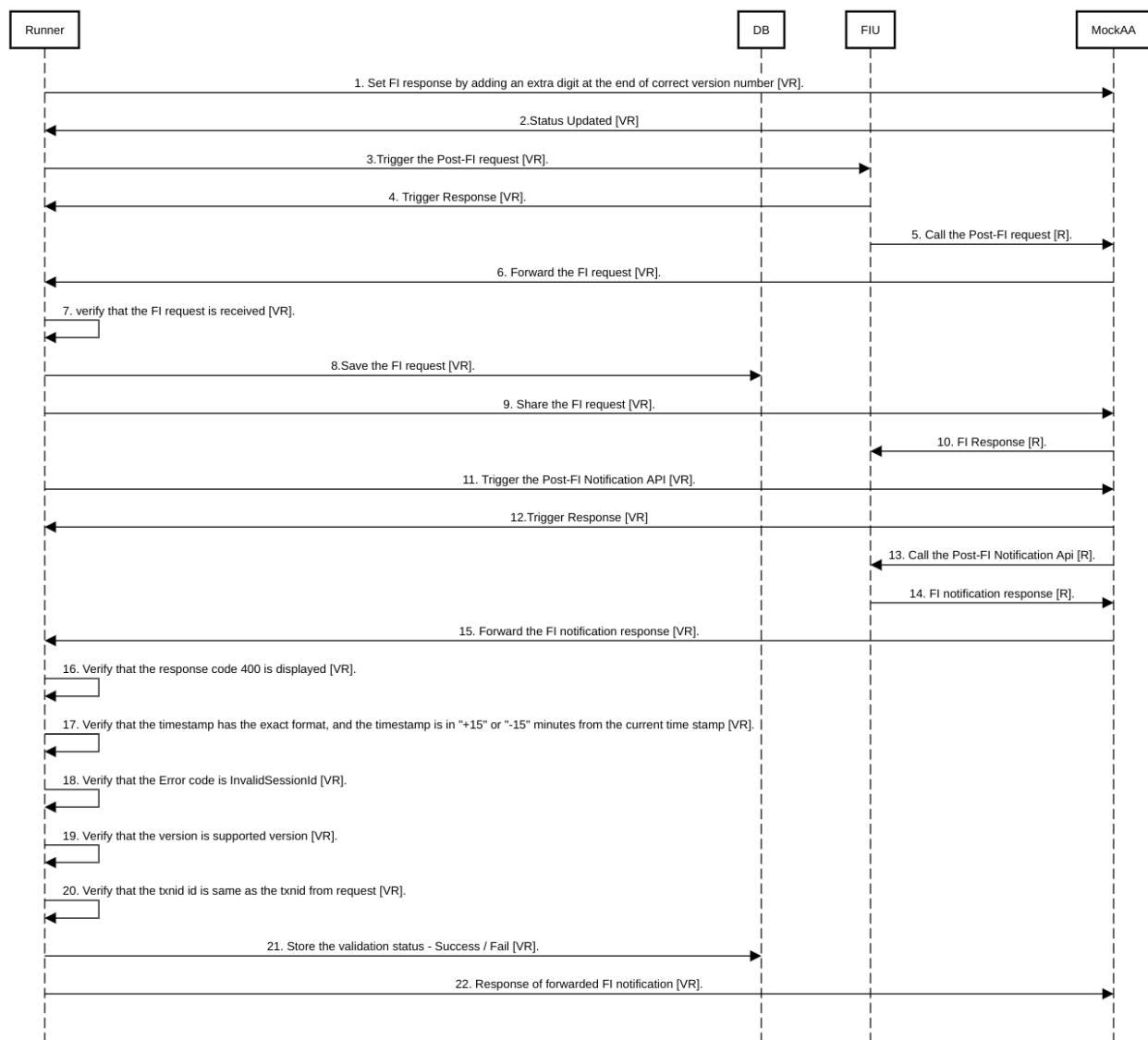
Runner->Runner:19. Verify that the version is supported version [VR].

Runner->Runner:20. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2017_POST_FINotification.feature Scenario 2017_2



Scenario 2017_3

title FIU_2017_POST_FINotification.feature Scenario 2017_3

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA:1. Set FI response by removing the last character or digit at the end of the correct version number[VR].

Runner<-MockAA:2.Status Updated [VR]

Runner->FIU:3.Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8.Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12.Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR].

Runner->Runner:16. Verify that the response code 400 is displayed [VR].

Runner->Runner:17. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:18. Verify that the Error code is InvalidSessionId [VR].

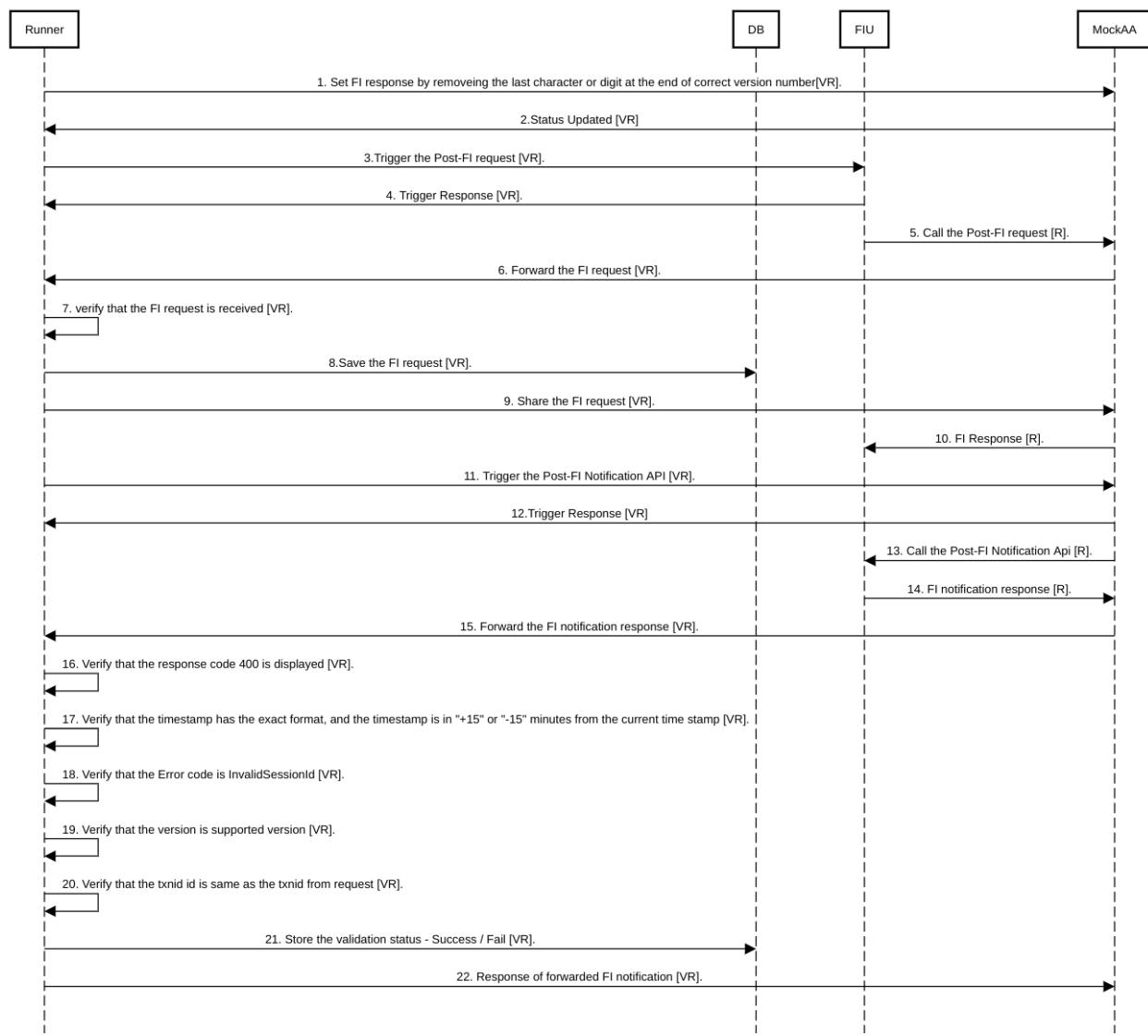
Runner->Runner:19. Verify that the version is supported version [VR].

Runner->Runner:20. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2017_POST_FINotification.feature Scenario 2017_3



FIU_2018_POST_FINotification.feature

Scenario 2018_1

title FIU_2018_POST_FINotification.feature Scenario 2018_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA:1. Set FI response with future timestamp(current time + 15 minutes) [VR].

Runner<-MockAA:2.Status Updated [VR]

Runner->FIU:3.Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8.Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12.Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR].

Runner->Runner:16. Verify that the response code 400 is displayed [VR].

Runner->Runner:17. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:18. Verify that the Error code is InvalidSessionId [VR].

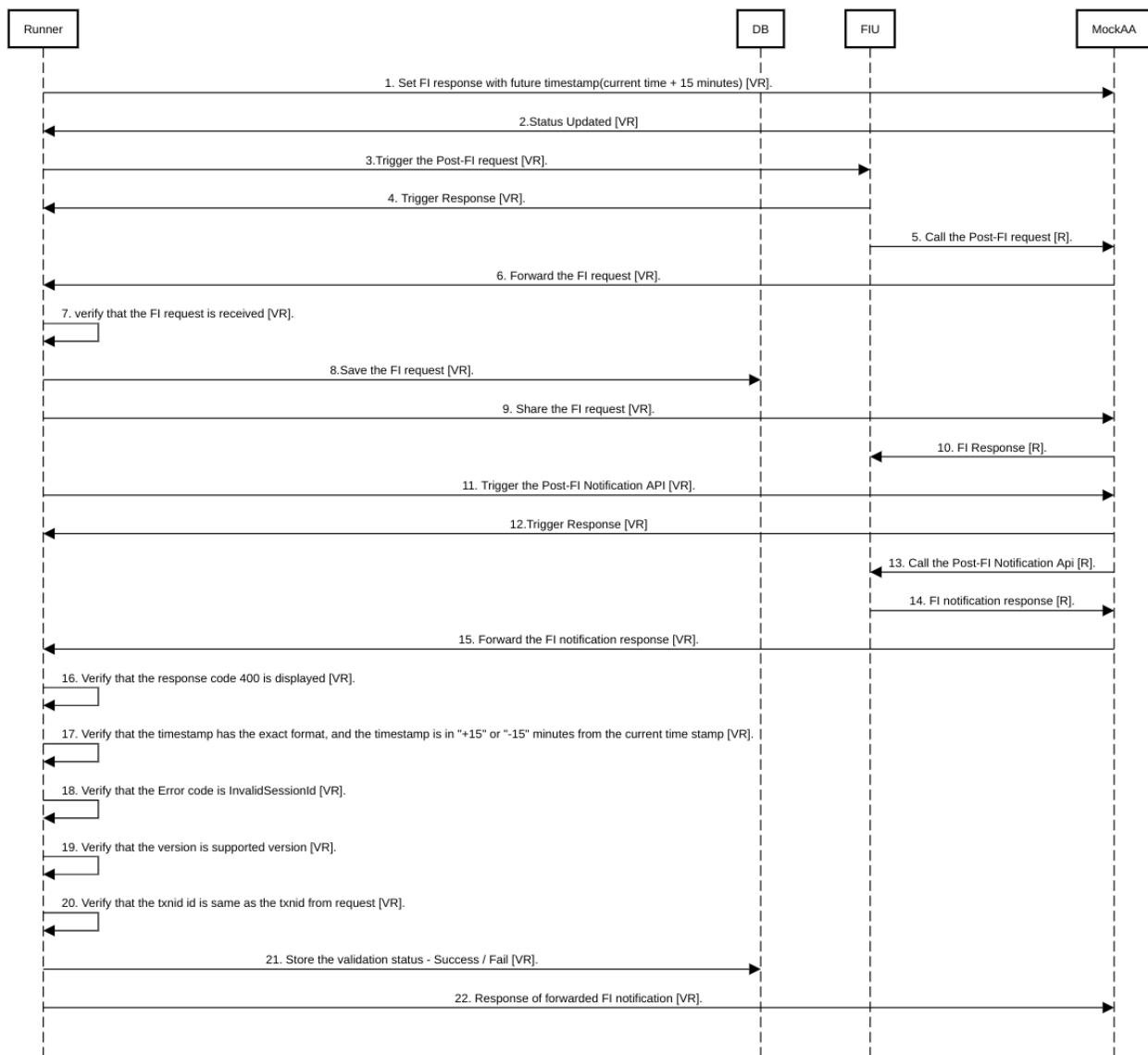
Runner->Runner:19. Verify that the version is supported version [VR].

Runner->Runner:20. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2018_POST_FINotification.feature Scenario 2018_1



Scenario 2018_2

title FIU_2018_POST_FINotification.feature Scenario 2018_2

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA:1. Set FI response with expired timestamp(current time - 15 minutes) [VR].

Runner<-MockAA:2. Status Updated [VR]

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12. Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR].

Runner->Runner:16. Verify that the response code 400 is displayed [VR].

Runner->Runner:17. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:18. Verify that the Error code is InvalidSessionId [VR].

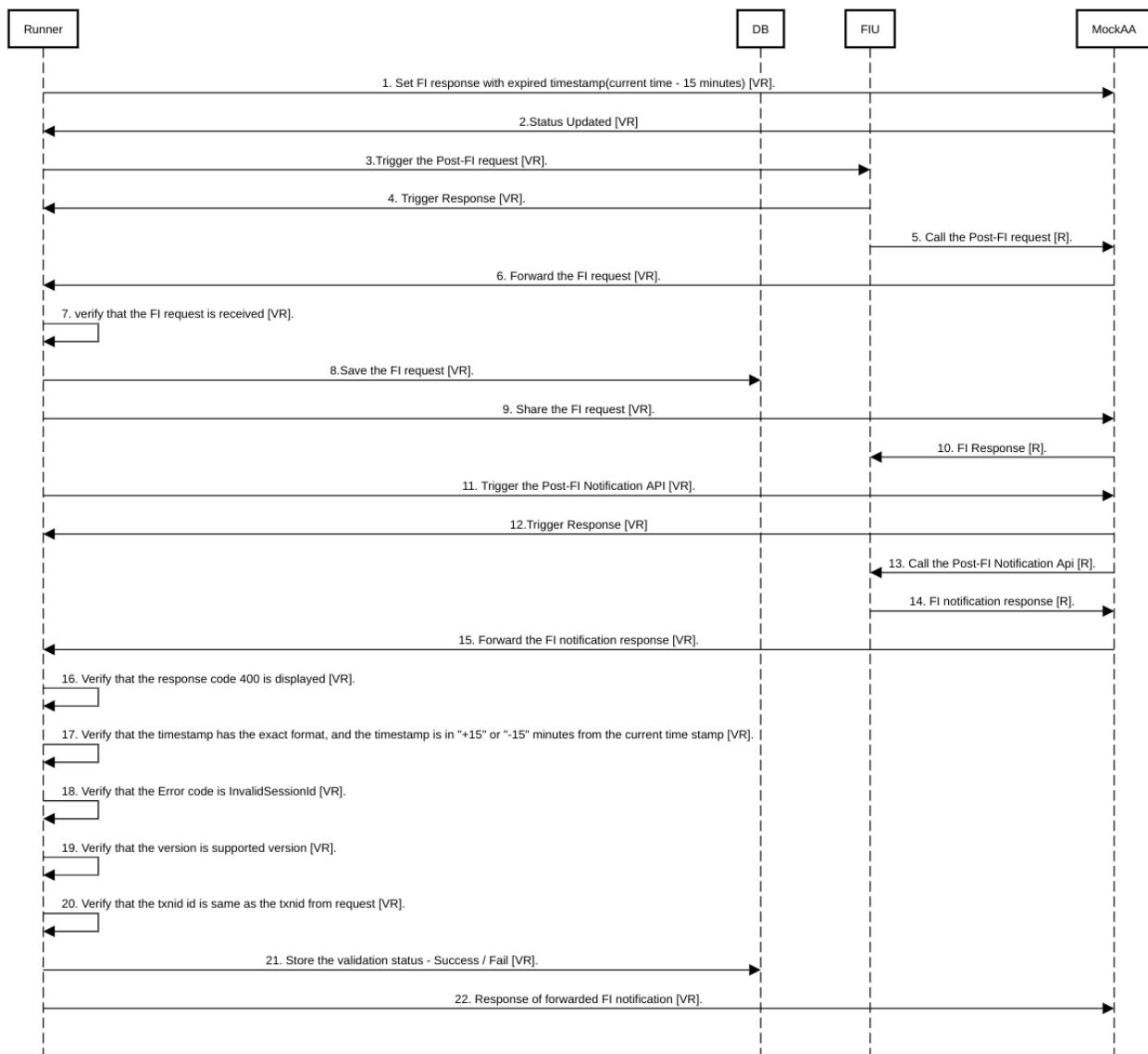
Runner->Runner:19. Verify that the version is supported version [VR].

Runner->Runner:20. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2018_POST_FINotification.feature Scenario 2018_2



FIU_2019_POST_FINotification.feature

Scenario 2019_1

title FIU_2019_POST_FINotification.feature Scenario 2019_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA:1. Set FI response with new txnid [VR].

Runner<-MockAA:2. Status Updated [VR]

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12. Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR].

Runner->Runner:16. Verify that the response code 400 is displayed [VR].

Runner->Runner:17. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:18. Verify that the Error code is InvalidSessionId [VR].

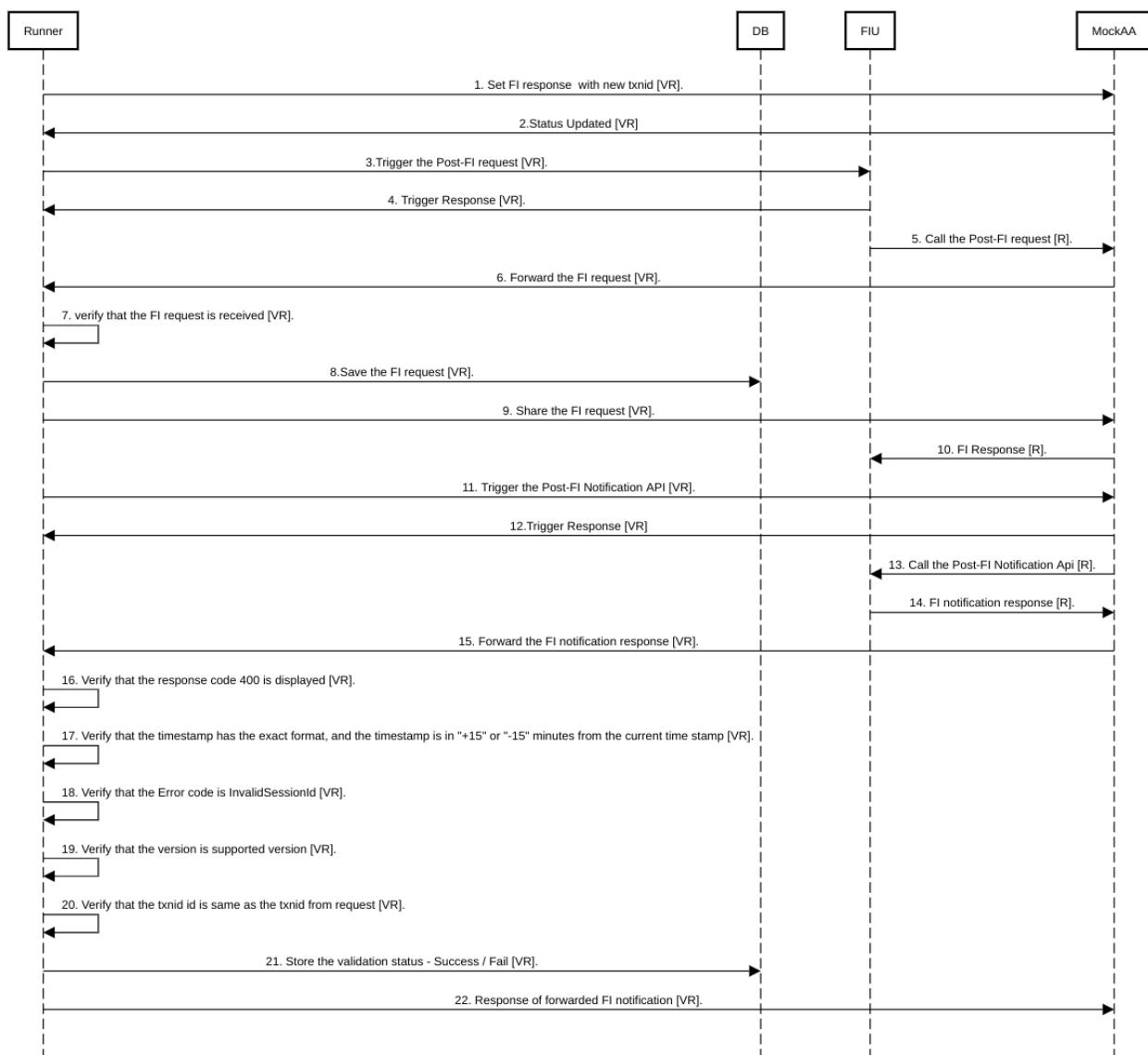
Runner->Runner:19. Verify that the version is supported version [VR].

Runner->Runner:20. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2019_POST_FINotification.feature Scenario 2019_1



FIU_2020_POST_FINotification.feature

Scenario 2020_1 [NC]

title FIU_2020_POST_FINotification.feature Scenario 2020_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the post consent response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response[VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner -> DB:8. Save the consent request [VR].

Runner->MockAA:9. Share the consent request [VR].

FIU<-MockAA:10. Consent Response [R].

Runner->MockAA:11. Set the consent handle response with "Pending" Status [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the get consent response [VR].

Runner<-MockAA:14. Status Updated [VR].

Runner->MockAA:15. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-Consent Notification Api [R].

FIU->MockAA:18. Consent notification response [R].

Runner<-MockAA:19. Forward the consent notification response [VR].

Runner->Runner:20. Verify the Get-Consent get request is received [VR].

Runner->MockAA:21. Response of forwarded consent notification [VR].

Runner->MockAA:22. Set the FI response with consent id of another consent request [VR].

Runner<-MockAA:23. Status Updated

Runner->FIU:24. Trigger the Post-FI request [VR].

Runner<-FIU:25. Trigger Response [VR].

FIU->MockAA:26. Call the Post-FI request [R].

MockAA ->Runner:27. Forward the FI request [VR].

Runner->Runner:28. verify that the FI request is received [VR].

Runner -> DB:29. Save the FI request [VR].

Runner->MockAA:30. Share the FI request [VR].

FIU<-MockAA:31. FI Response [R].

Runner->MockAA:32. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:33. Trigger Response [VR]

FIU<-MockAA:34. Call the Post-FI Notification Api [R].

FIU->MockAA:35. FI notification response [R].

Runner<-MockAA:36. Forward the FI notification response [VR].

Runner ->Runner:37. Verify that the response code 400 is displayed [VR].

Runner->Runner:38. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

Runner->Runner:39. Verify that the Error code is InvalidSessionId [VR].

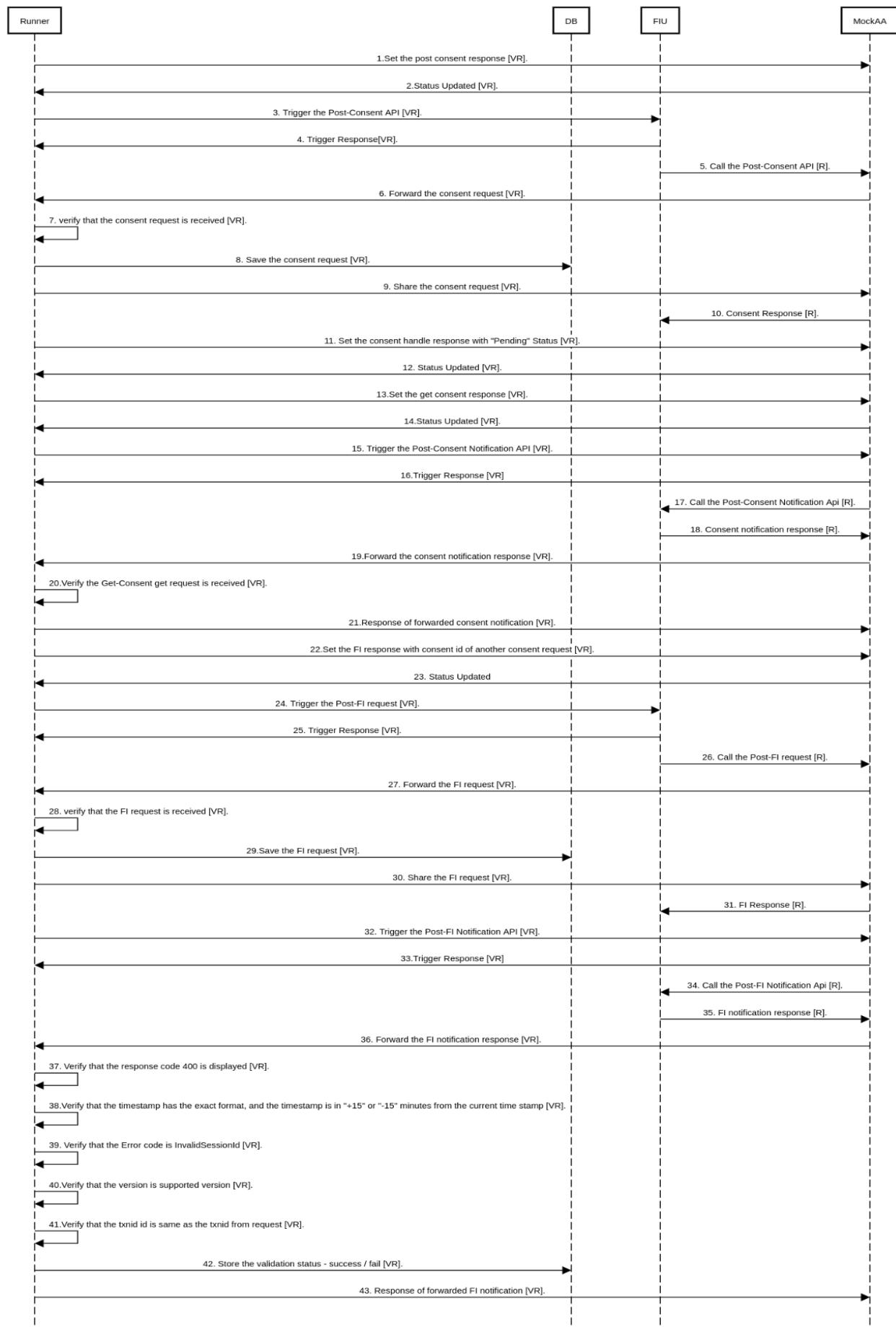
Runner->Runner:40. Verify that the version is supported version [VR].

Runner->Runner:41. Verify that the txnid id is same as the txnid from request [VR].

Runner -> DB:42. Store the validation status - success / fail [VR].

Runner->MockAA:43. Response of forwarded FI notification [VR].

FIU_2020_POST _FINotification.feature Scenario 2020_1



FIU_2021_POST_FINotification.feature

Scenario 2021_1

title FIU_2021_POST_FINotification.feature Scenario 2021_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA:1. Set FI response with invalid JWS signature. [VR].

Runner<-MockAA:2. Status Updated [VR]

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:12. Trigger Response [VR]

FIU<-MockAA:13. Call the Post-FI Notification Api [R].

FIU->MockAA:14. FI notification response [R].

Runner<-MockAA:15. Forward the FI notification response [VR].

Runner->Runner:16. Verify that the response code 400 is displayed [VR].

Runner->Runner:17. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current timestamp [VR].

Runner->Runner:18. Verify that the Error code is InvalidSessionId [VR].

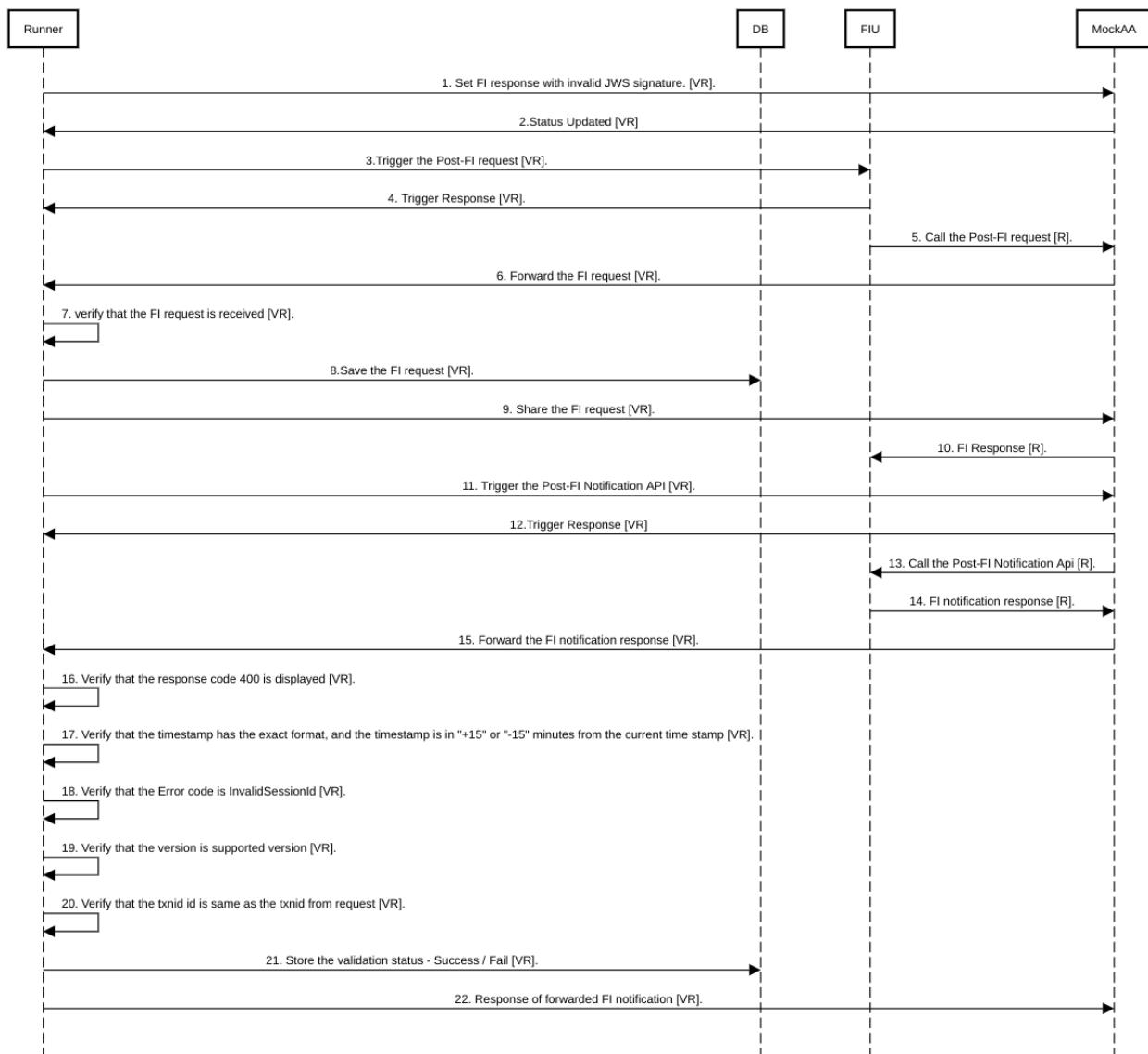
Runner->Runner:19. Verify that the version is supported version [VR].

Runner->Runner:20. Verify that the txnid id is the same as the txnid from request [VR].

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2021_POST_FINotification.feature Scenario 2021_1



FIU_2022_GET _FIFetchSessionId.feature

Scenario 2022_1

title FIU_2022_GET _FIFetchSessionId.feature Scenario 2022_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA: 1. Set the Post-FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the FI Fetch response with incorrect version number[VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the FI notification status as "Ready" [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

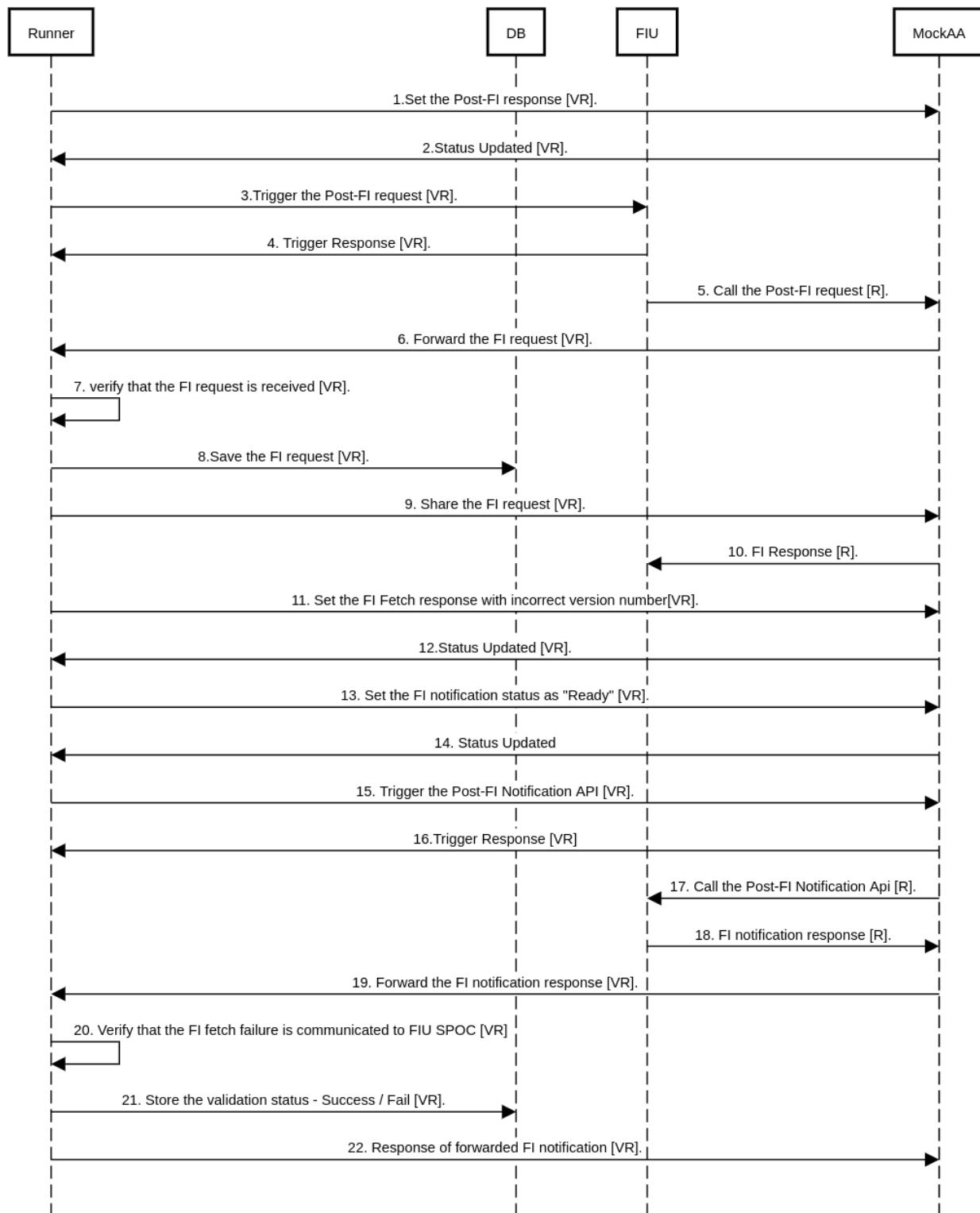
Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the FI fetch failure is communicated to FIU SPOC [VR]

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2022_GET _FIFetchSessionId.feature Scenario 2022_1



Scenario 2022_2

title FIU_2022_GET_FIFetchSessionId.feature Scenario 2022_2

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA: 1. Set the Post-FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the FI Fetch response by adding an extra digit at the end of the correct version number [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the FI notification status as "Ready" [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

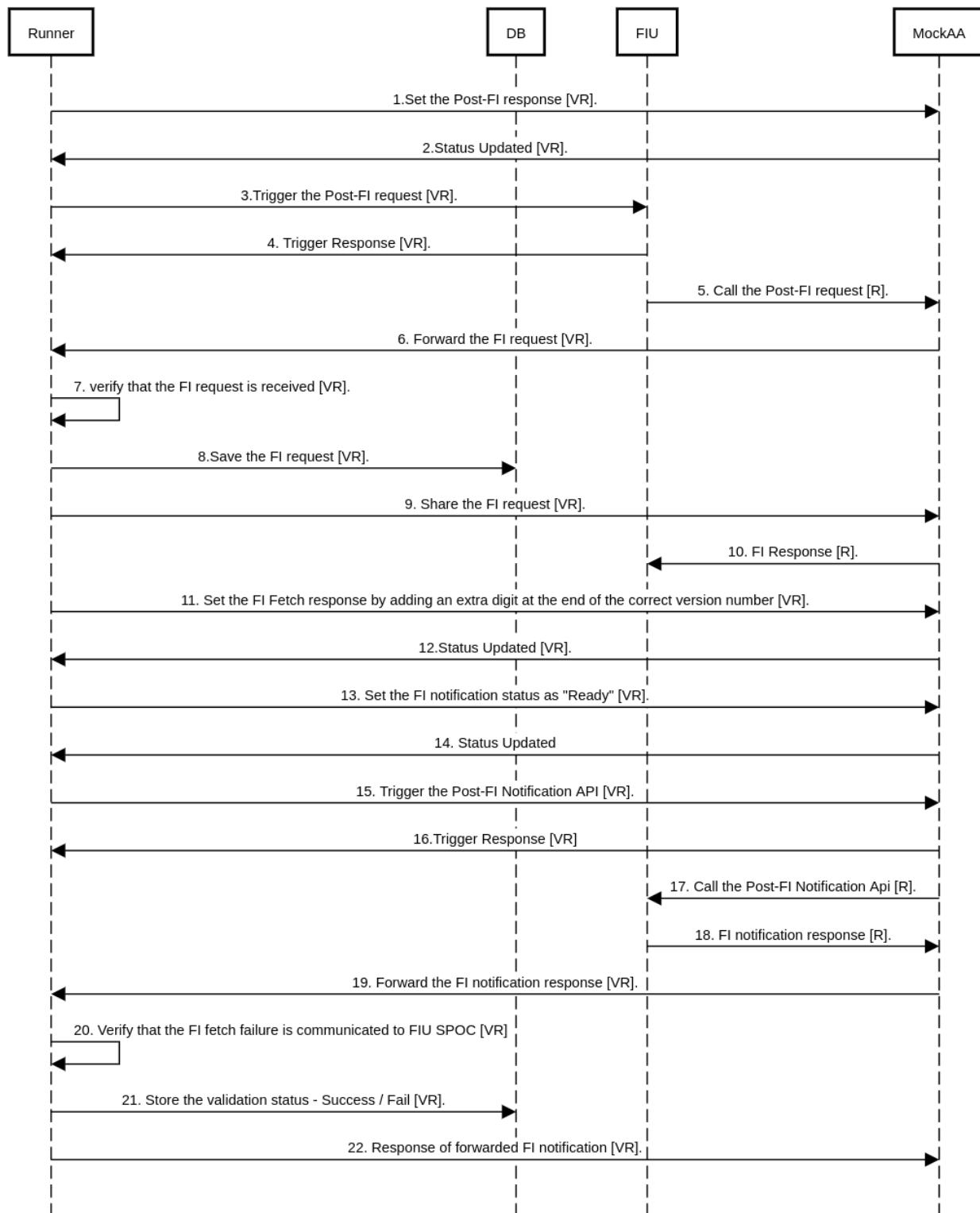
Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the FI fetch failure is communicated to FIU SPOC [VR]

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2022_GET _FIFetchSessionId.feature Scenario 2022_2



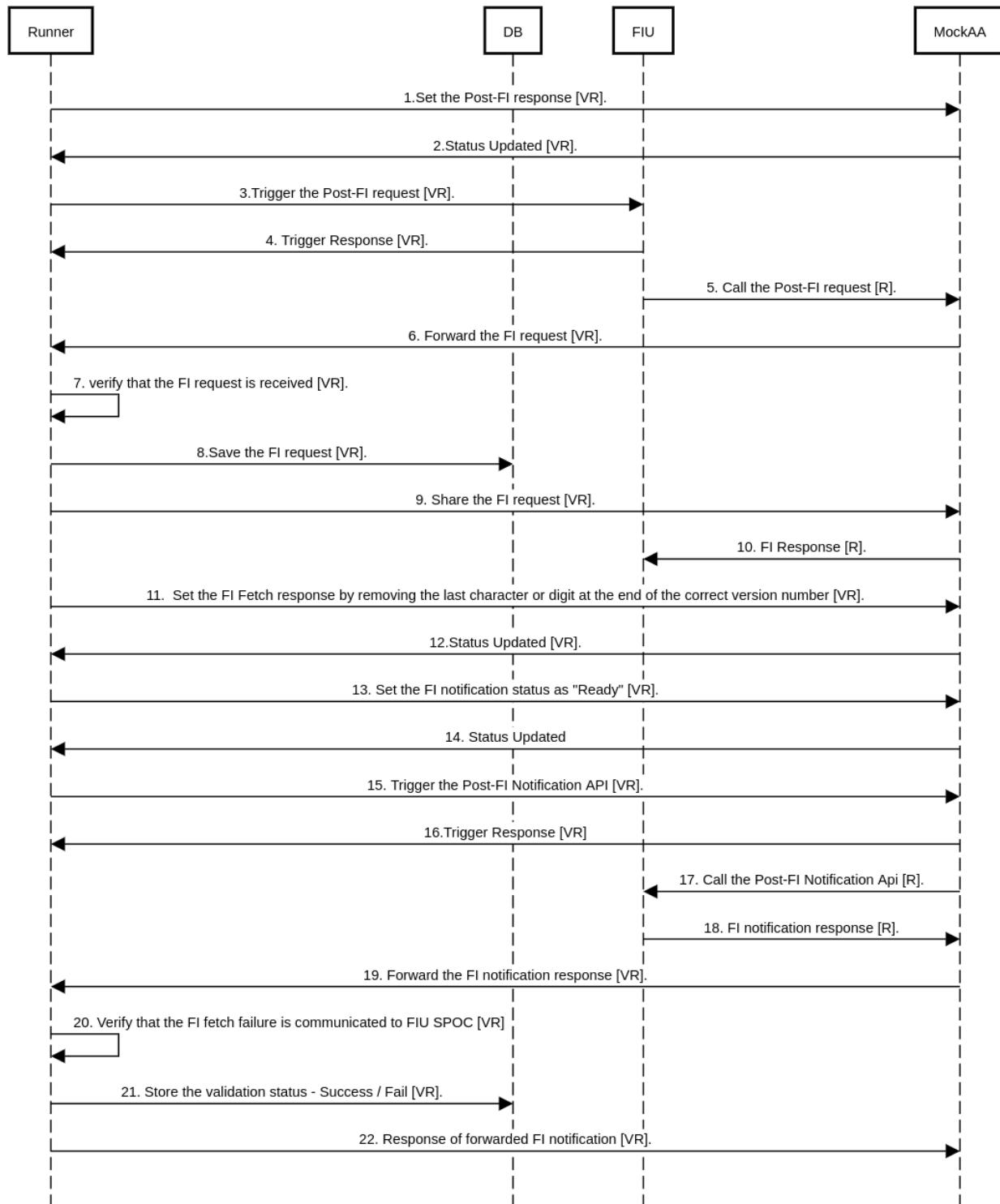
Scenario 2022_3

title FIU_2022_GET_FIFetchSessionId.feature Scenario 2022_3

participant Runner
participant DB
participant FIU
participant MockAA

Runner->MockAA: 1. Set the Post-FI response [VR].
Runner<-MockAA:2. Status Updated [VR].
Runner->FIU:3. Trigger the Post-FI request [VR].
Runner<-FIU:4. Trigger Response [VR].
FIU->MockAA:5. Call the Post-FI request [R].
MockAA ->Runner:6. Forward the FI request [VR].
Runner->Runner:7. verify that the FI request is received [VR].
Runner -> DB:8. Save the FI request [VR].
Runner->MockAA:9. Share the FI request [VR].
FIU<-MockAA:10. FI Response [R].
Runner->MockAA:11. Set the FI Fetch response by removing the last character or digit at the end of the correct version number [VR].
Runner<-MockAA:12. Status Updated [VR].
Runner->MockAA:13. Set the FI notification status as "Ready" [VR].
Runner<-MockAA:14. Status Updated
Runner->MockAA:15. Trigger the Post-FI Notification API [VR].
Runner<-MockAA:16. Trigger Response [VR]
FIU<-MockAA:17. Call the Post-FI Notification Api [R].
FIU->MockAA:18. FI notification response [R].
Runner<-MockAA:19. Forward the FI notification response [VR].
Runner->Runner:20. Verify that the FI fetch failure is communicated to FIU SPOC [VR]
Runner -> DB:21. Store the validation status - Success / Fail [VR].
Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2022_GET _FIFetchSessionId.feature Scenario 2022_3



FIU_2023_GET _FIFetchSessionId.feature

Scenario 2023_1

title FIU_2023_GET _FIFetchSessionId.feature Scenario 2023_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA: 1. Set the Post-FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the FI Fetch response with the future timestamp(current time + 15 minutes) [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the FI notification status as "Ready" [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

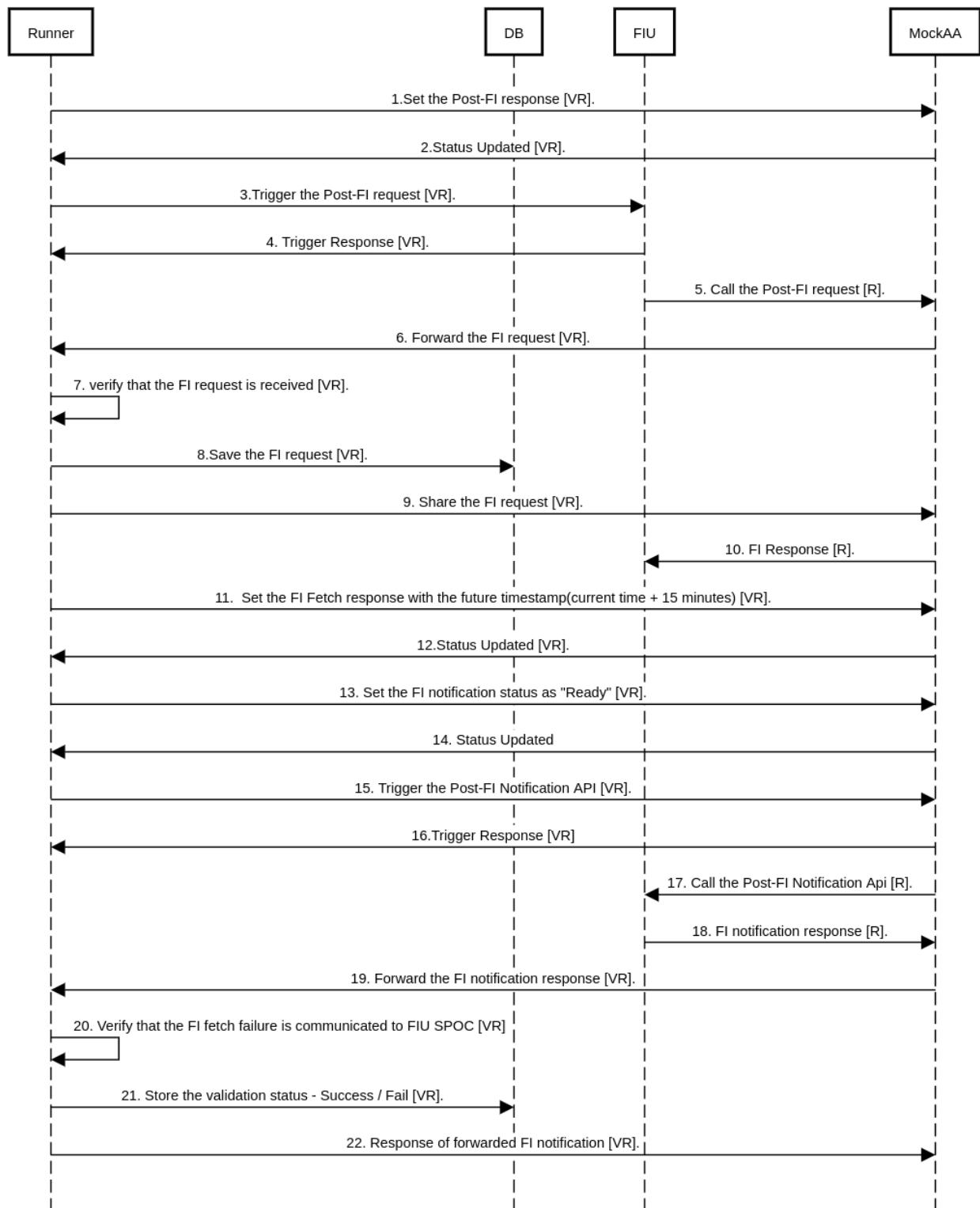
Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the FI fetch failure is communicated to FIU SPOC [VR]

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2023_GET _FIFetchSessionId.feature Scenario 2023_1



Scenario 2023_2

title FIU_2023_GET_FIFetchSessionId.feature Scenario 2023_2

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA: 1. Set the Post-FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the FI Fetch response with expired timestamp(current time - 15 minutes) [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the FI notification status as "Ready" [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

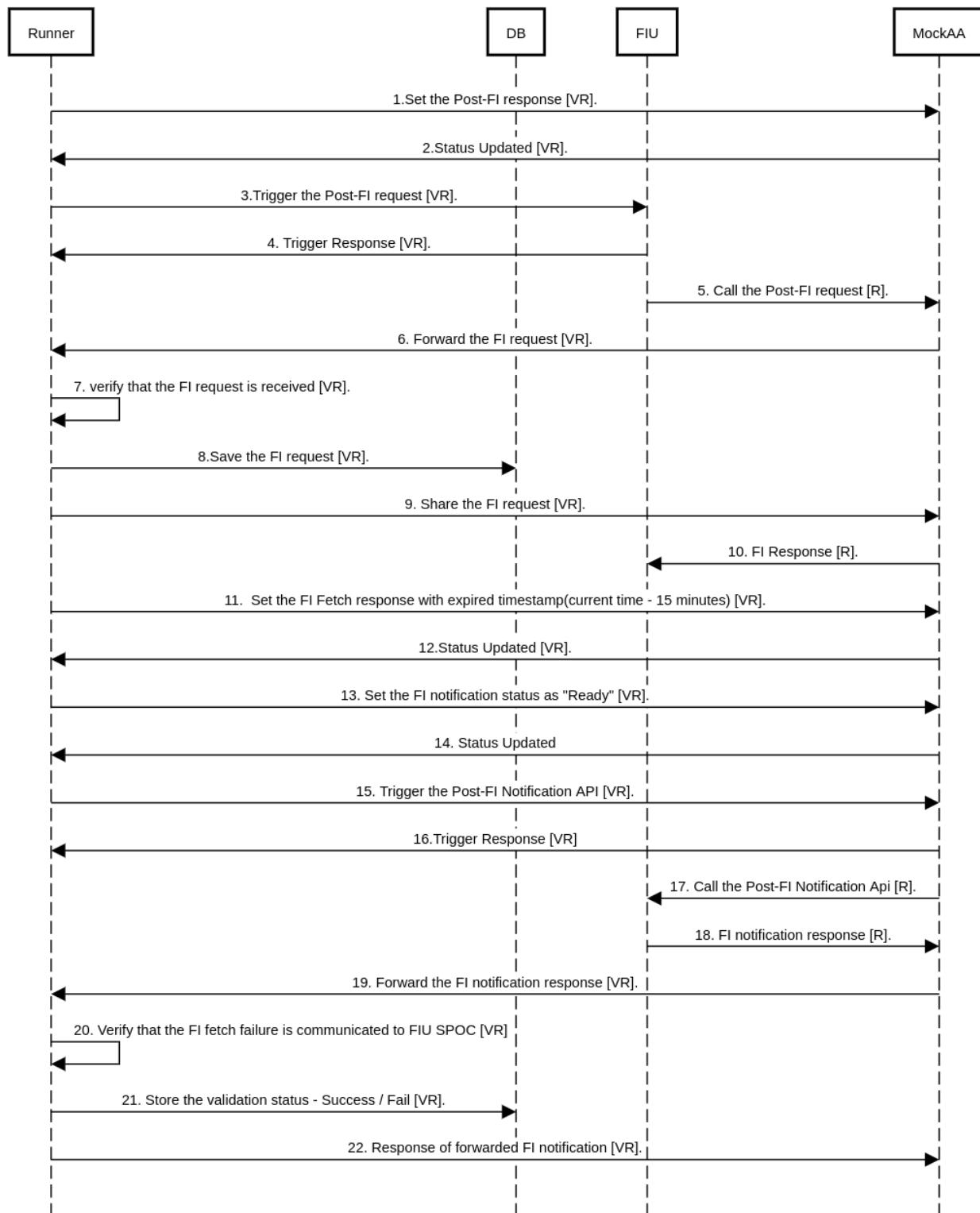
Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the FI fetch failure is communicated to FIU SPOC [VR]

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2023_GET _FIFetchSessionId.feature Scenario 2023_2



FIU_2024_GET _FIFetchSessionId.feature

Scenario 2024_1

title FIU_2024_GET _FIFetchSessionId.feature Scenario 2024_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA: 1. Set the Post-FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the FI Fetch response with the random linkRefNumber for FI Data object [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the FI notification status as "Ready" [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

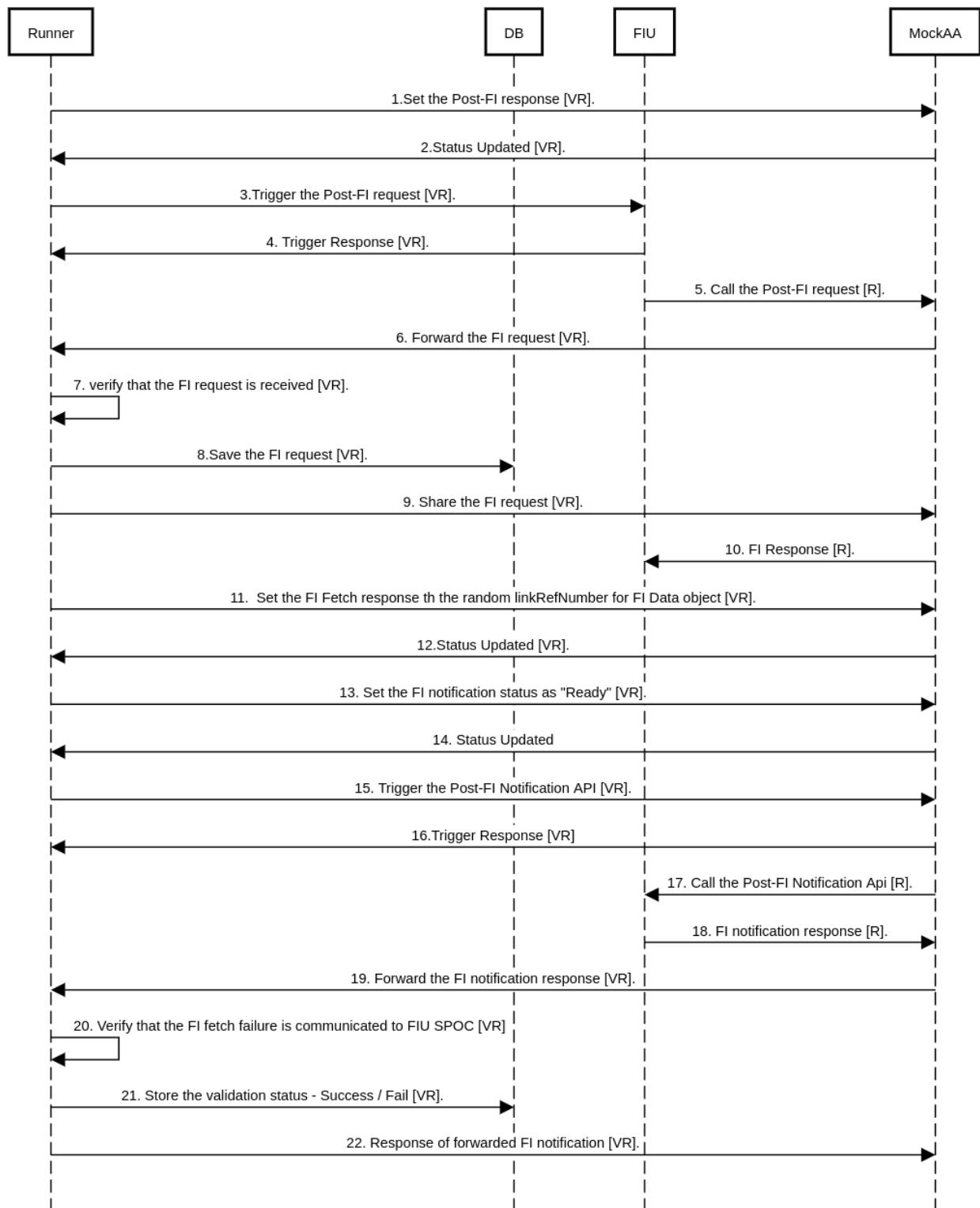
Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the FI fetch failure is communicated to FIU SPOC [VR]

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2024_GET _FIFetchSessionId.feature Scenario 2024_1



Scenario 2024_2

title FIU_2024_GET_FIFetchSessionId.feature Scenario 2024_2

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA: 1. Set the Post-FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the FI Fetch response with the random
MaskedAccountNumber for FI Data object [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the FI notification status as "Ready" [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

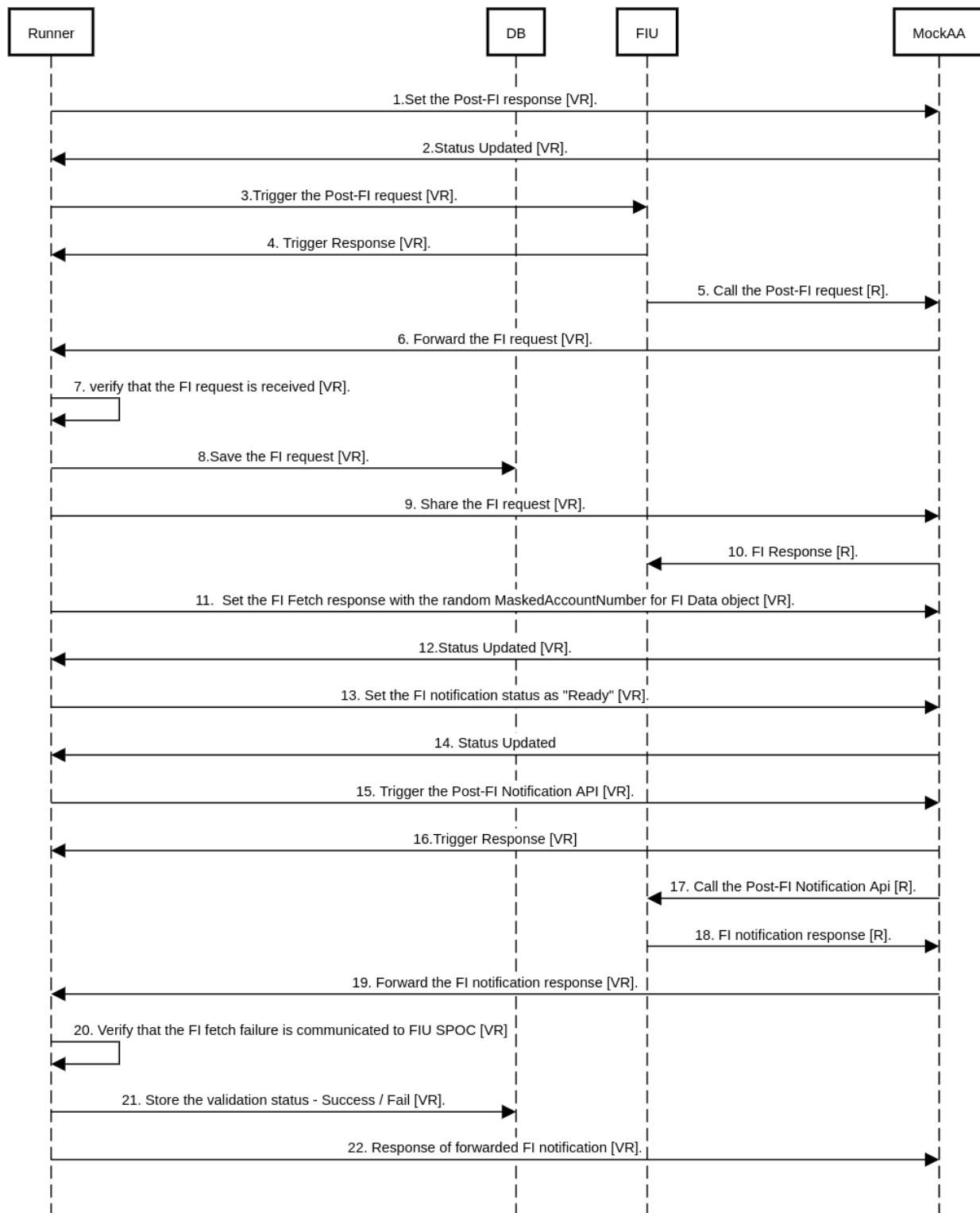
Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the FI fetch failure is communicated to FIU
SPOC [VR]

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2024_GET _FIFetchSessionId.feature Scenario 2024_2



Scenario 2024_3

title FIU_2024_GET_FIFetchSessionId.feature Scenario 2024_3

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA: 1. Set the Post-FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the FI Fetch response by adding an extra character to Encrypted FI for FI Data object [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the FI notification status as "Ready" [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

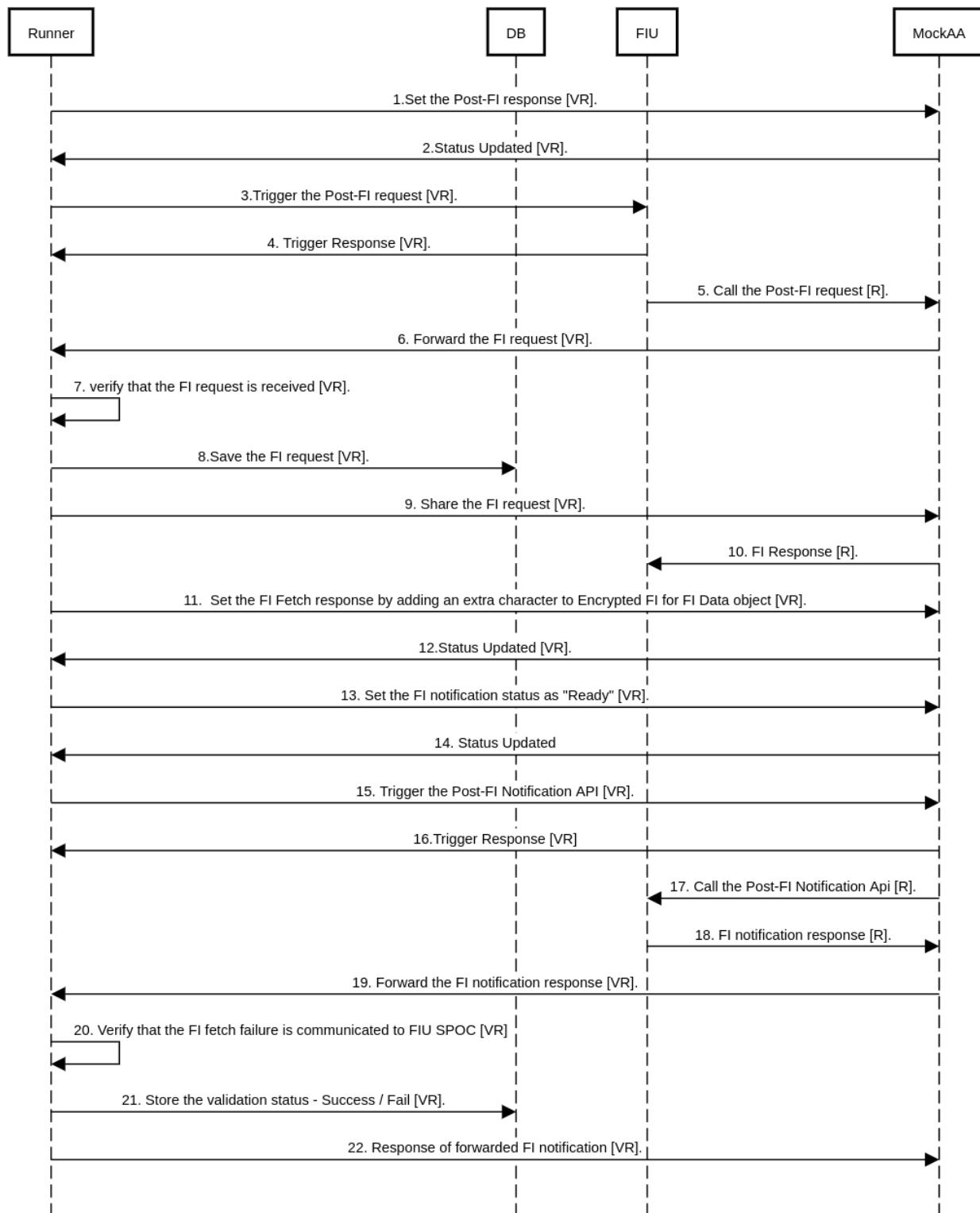
Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the FI fetch failure is communicated to FIU SPOC [VR]

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2024_GET _FIFetchSessionId.feature Scenario 2024_3



Scenario 2024_4

title FIU_2024_GET_FIFetchSessionId.feature Scenario 2024_4

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA: 1. Set the Post-FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the FI Fetch response by removing a single character at the beginning of encrypted FI.[VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the FI notification status as "Ready" [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

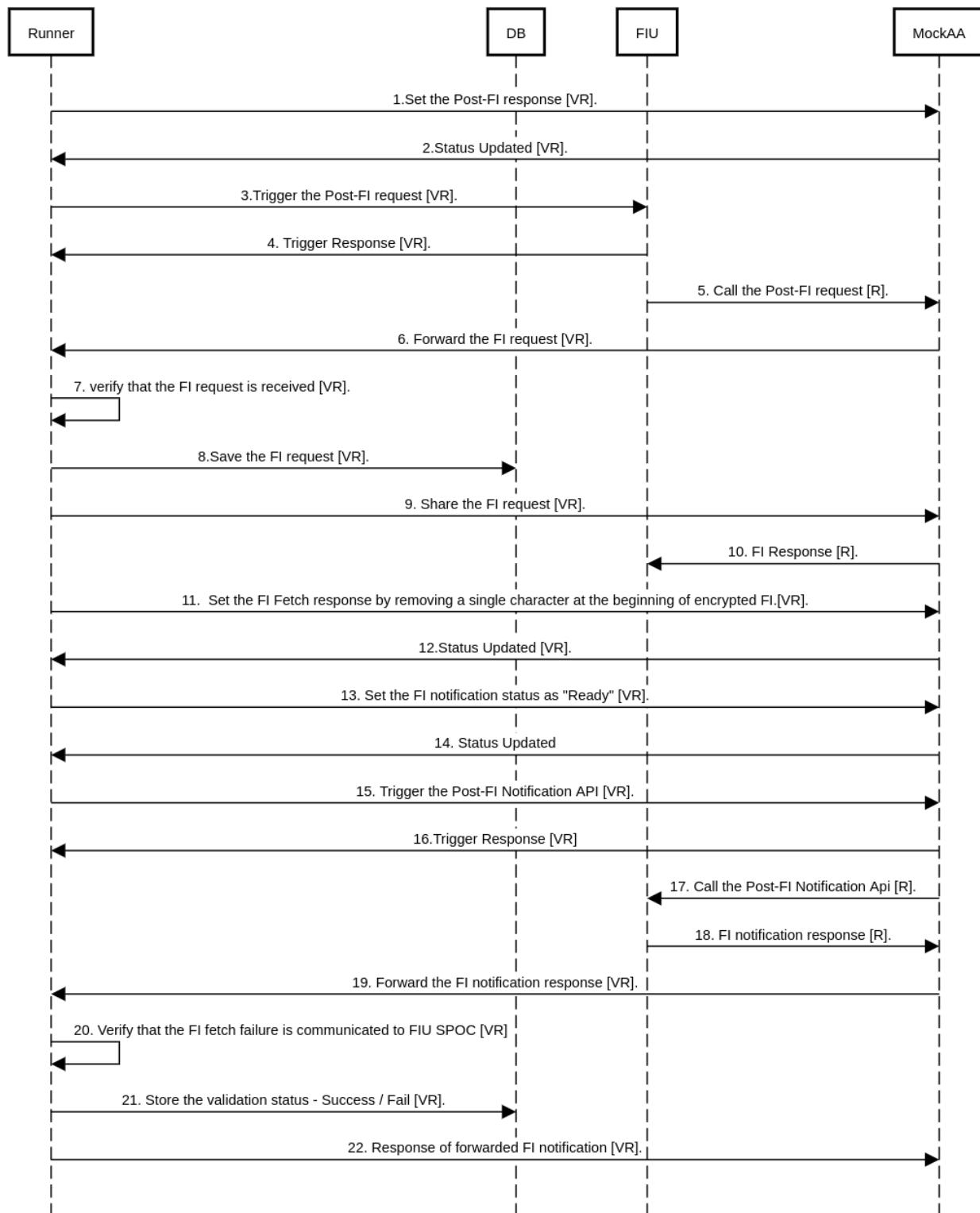
Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the FI fetch failure is communicated to FIU SPOC [VR]

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2024_GET _FIFetchSessionId.feature Scenario 2024_4



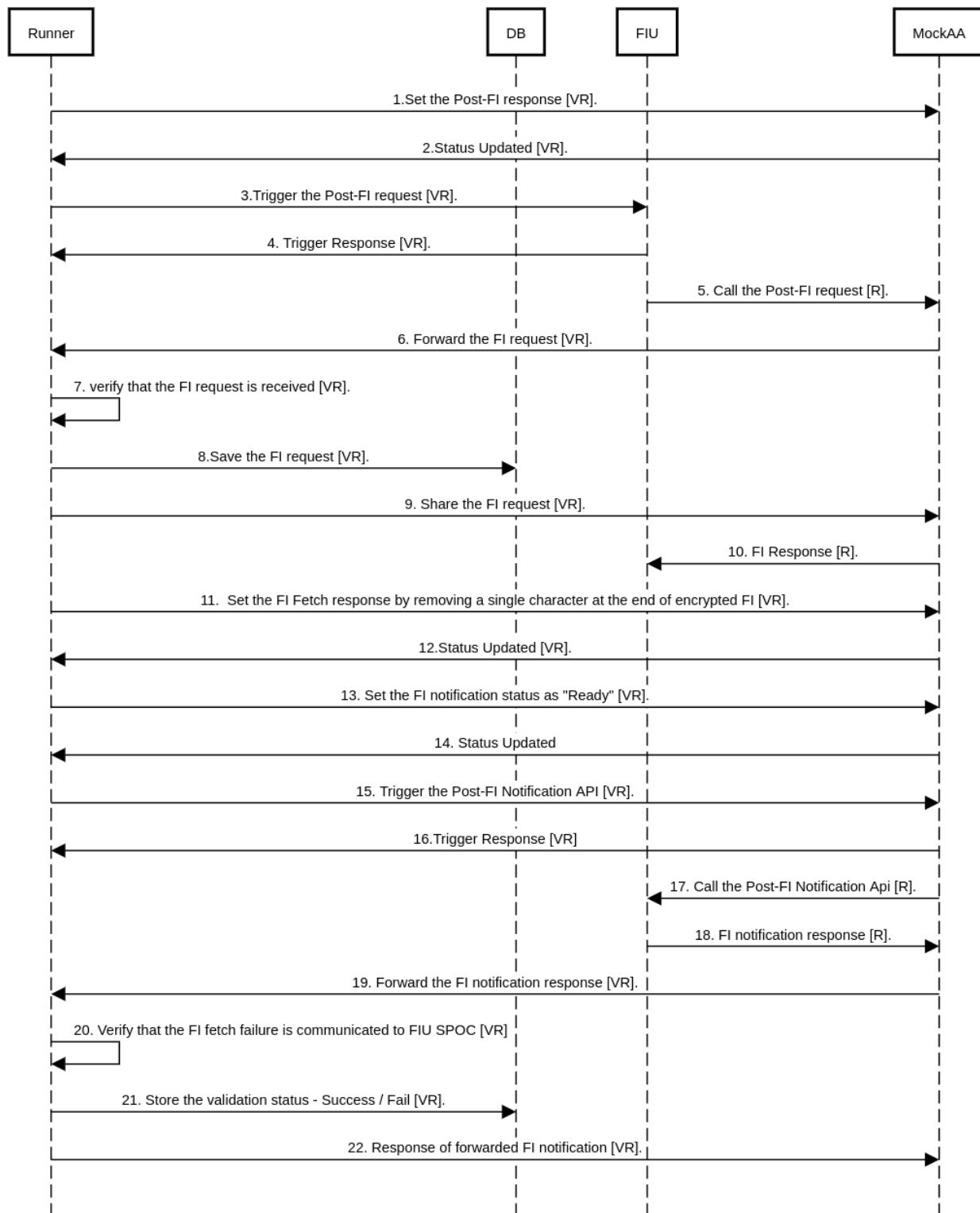
Scenario 2024_5

title FIU_2024_GET_FIFetchSessionId.feature Scenario 2024_5

participant Runner
participant DB
participant FIU
participant MockAA

Runner->MockAA: 1. Set the Post-FI response [VR].
Runner<-MockAA:2. Status Updated [VR].
Runner->FIU:3. Trigger the Post-FI request [VR].
Runner<-FIU:4. Trigger Response [VR].
FIU->MockAA:5. Call the Post-FI request [R].
MockAA ->Runner:6. Forward the FI request [VR].
Runner->Runner:7. verify that the FI request is received [VR].
Runner -> DB:8. Save the FI request [VR].
Runner->MockAA:9. Share the FI request [VR].
FIU<-MockAA:10. FI Response [R].
Runner->MockAA:11. Set the FI Fetch response by removing a single character at the end of encrypted FI [VR].
Runner<-MockAA:12. Status Updated [VR].
Runner->MockAA:13. Set the FI notification status as "Ready" [VR].
Runner<-MockAA:14. Status Updated
Runner->MockAA:15. Trigger the Post-FI Notification API [VR].
Runner<-MockAA:16. Trigger Response [VR]
FIU<-MockAA:17. Call the Post-FI Notification Api [R].
FIU->MockAA:18. FI notification response [R].
Runner<-MockAA:19. Forward the FI notification response [VR].
Runner->Runner:20. Verify that the FI fetch failure is communicated to FIU SPOC [VR]
Runner -> DB:21. Store the validation status - Success / Fail [VR].
Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2024_GET _FIFetchSessionId.feature Scenario 2024_5



FIU_2025_GET _FIFetchSessionId.feature

Scenario 2025_1

title FIU_2025_GET _FIFetchSessionId.feature Scenario 2025_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA: 1. Set the Post-FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the FI Fetch response with invalid Crypto Algo for KeyMaterial object [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the FI notification status as "Ready" [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

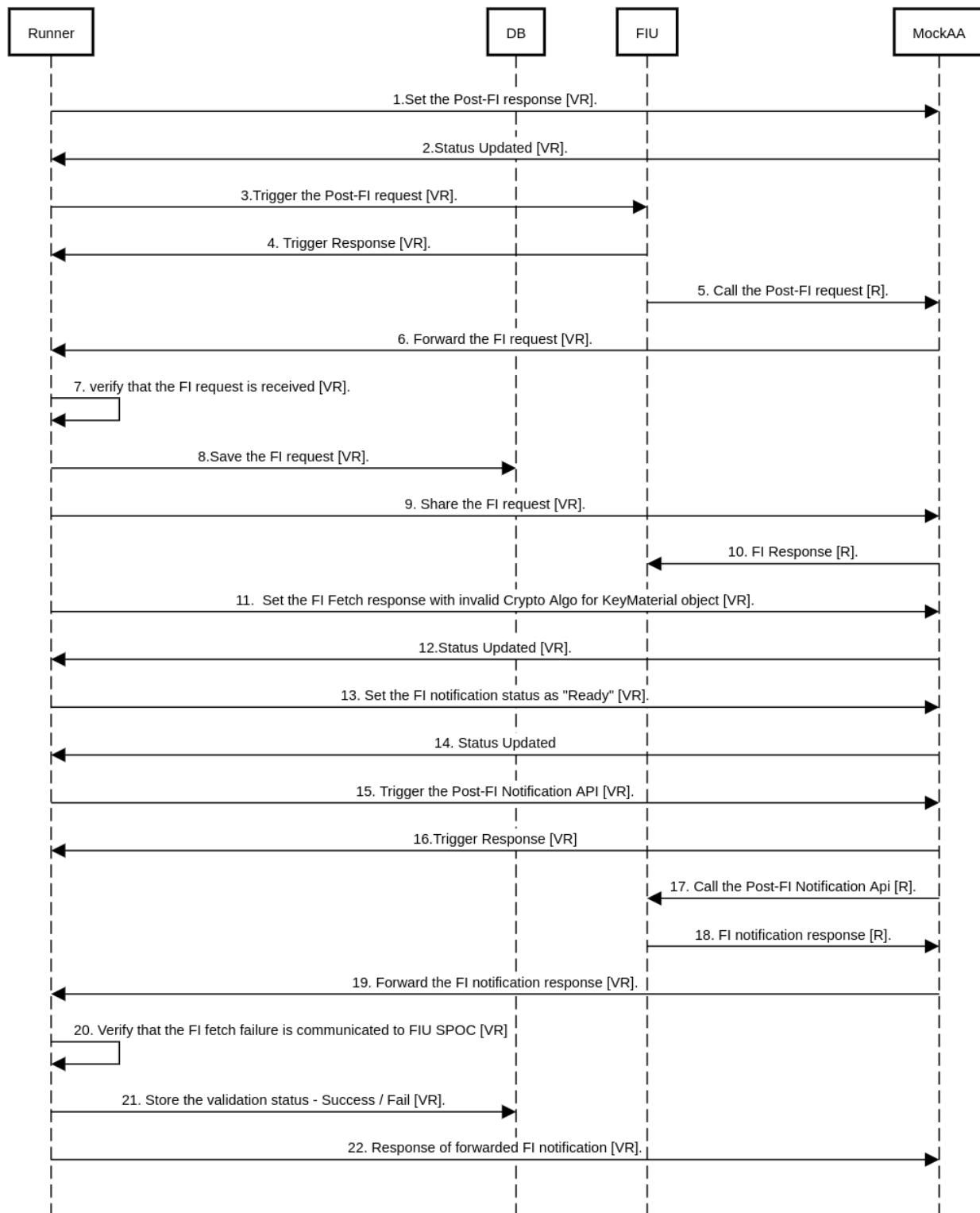
Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the FI fetch failure is communicated to FIU SPOC [VR]

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2025_GET _FIFetchSessionId.feature Scenario 2025_1



Scenario 2025_2

title FIU_2025_GET_FIFetchSessionId.feature Scenario 2025_2

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA: 1. Set the Post-FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the FI Fetch response with invalid Curve for KeyMaterial object. [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the FI notification status as "Ready" [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

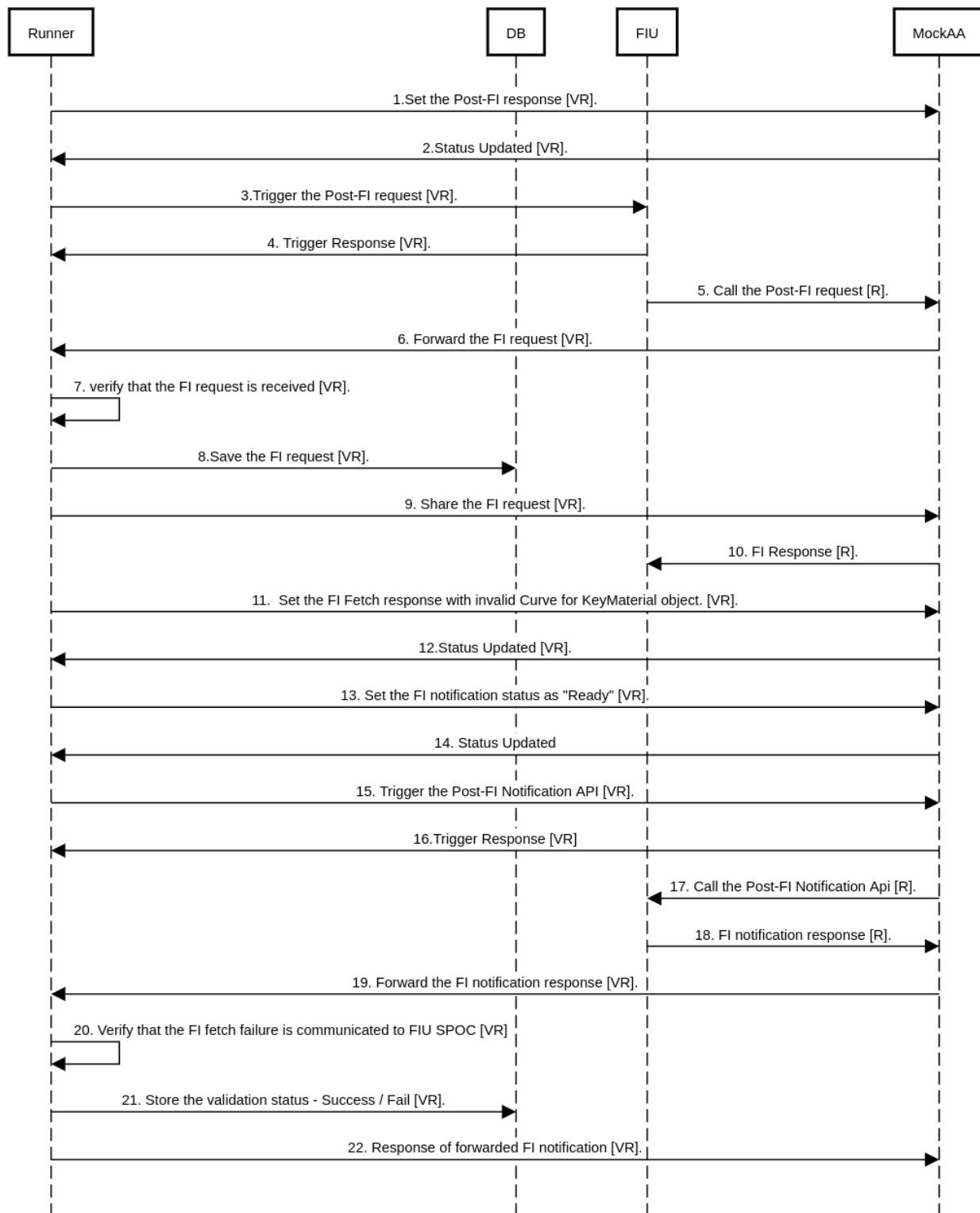
Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the FI fetch failure is communicated to FIU SPOC [VR]

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2025_GET _FIFetchSessionId.feature Scenario 2025_2



Scenario 2025_3

title FIU_2025_GET_FIFetchSessionId.feature Scenario 2025_3

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA: 1. Set the Post-FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the FI Fetch response with Params as interger for KeyMaterial object [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the FI notification status as "Ready" [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

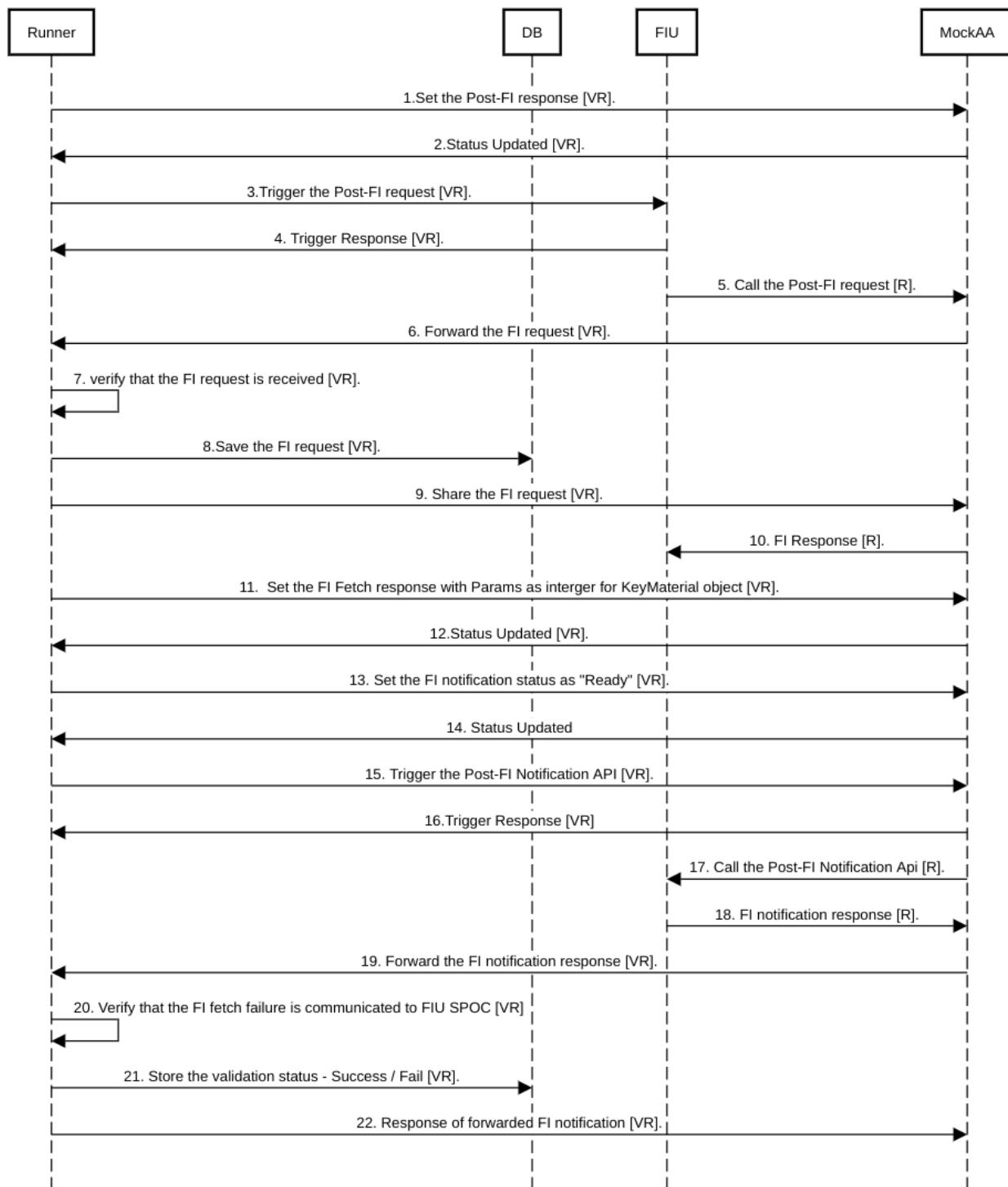
Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the FI fetch failure is communicated to FIU SPOC [VR]

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2025_GET _FIFetchSessionId.feature Scenario 2025_3



Scenario 2025_4

title FIU_2025_GET_FIFetchSessionId.feature Scenario 2025_4

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA: 1. Set the Post-FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the FI Fetch response with Invalid expiry date format for KeyMaterial object [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the FI notification status as "Ready" [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

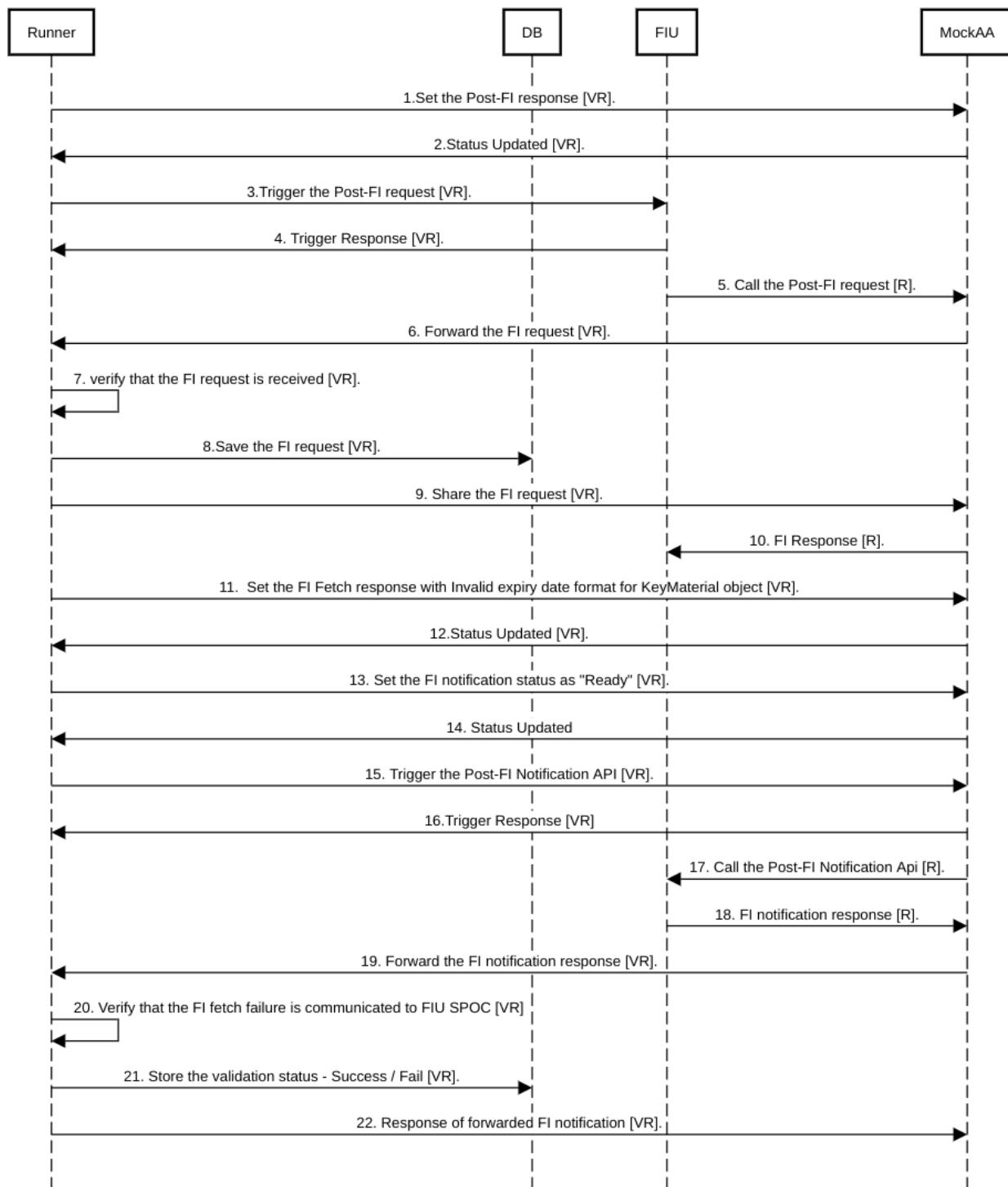
Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the FI fetch failure is communicated to FIU SPOC [VR]

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2025_GET _FIFetchSessionId.feature Scenario 2025_4



Scenario 2025_5

title FIU_2025_GET_FIFetchSessionId.feature Scenario 2025_5

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA: 1. Set the Post-FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the FI Fetch response with expired date in expiry date format for KeyMaterial object [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the FI notification status as "Ready" [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

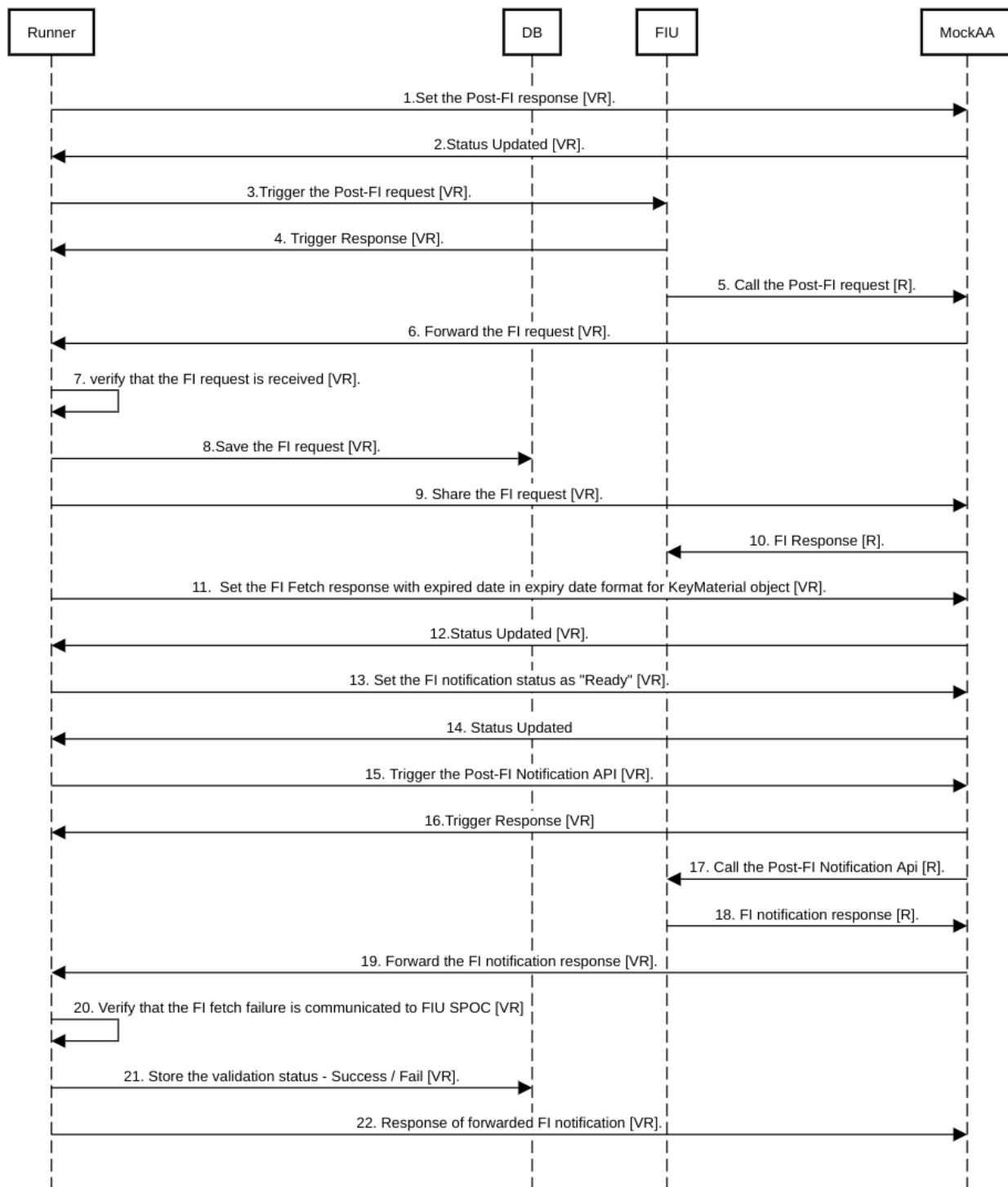
Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the FI fetch failure is communicated to FIU SPOC [VR]

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2025_GET _FIFetchSessionId.feature Scenario 2025_5



Scenario 2025_6

title FIU_2025_GET_FIFetchSessionId.feature Scenario 2025_6

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA: 1. Set the Post-FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the FI Fetch response with integer value for parameter for KeyMaterial object [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the FI notification status as "Ready" [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

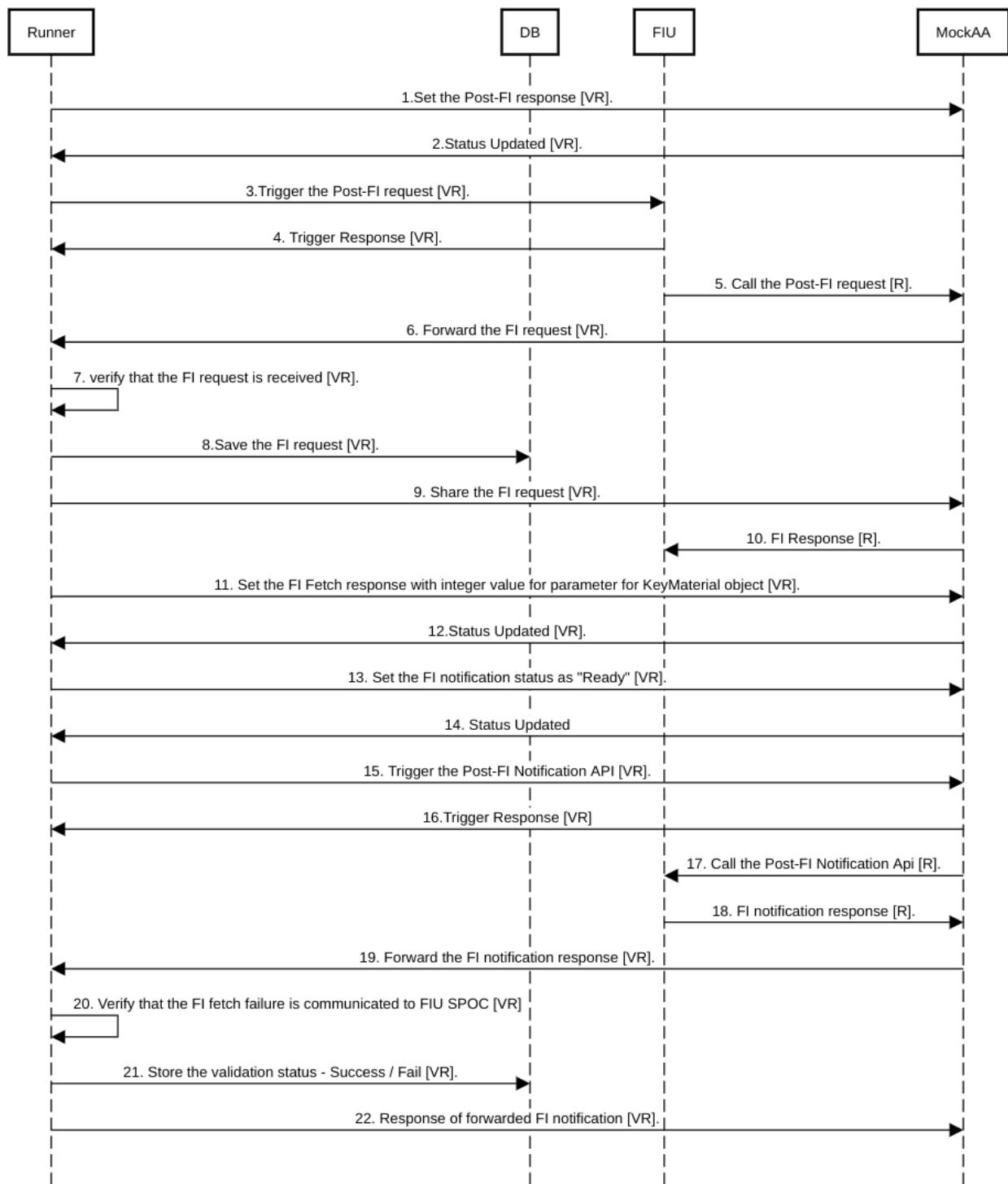
Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the FI fetch failure is communicated to FIU SPOC [VR]

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2025_GET _FIFetchSessionId.feature Scenario 2025_6



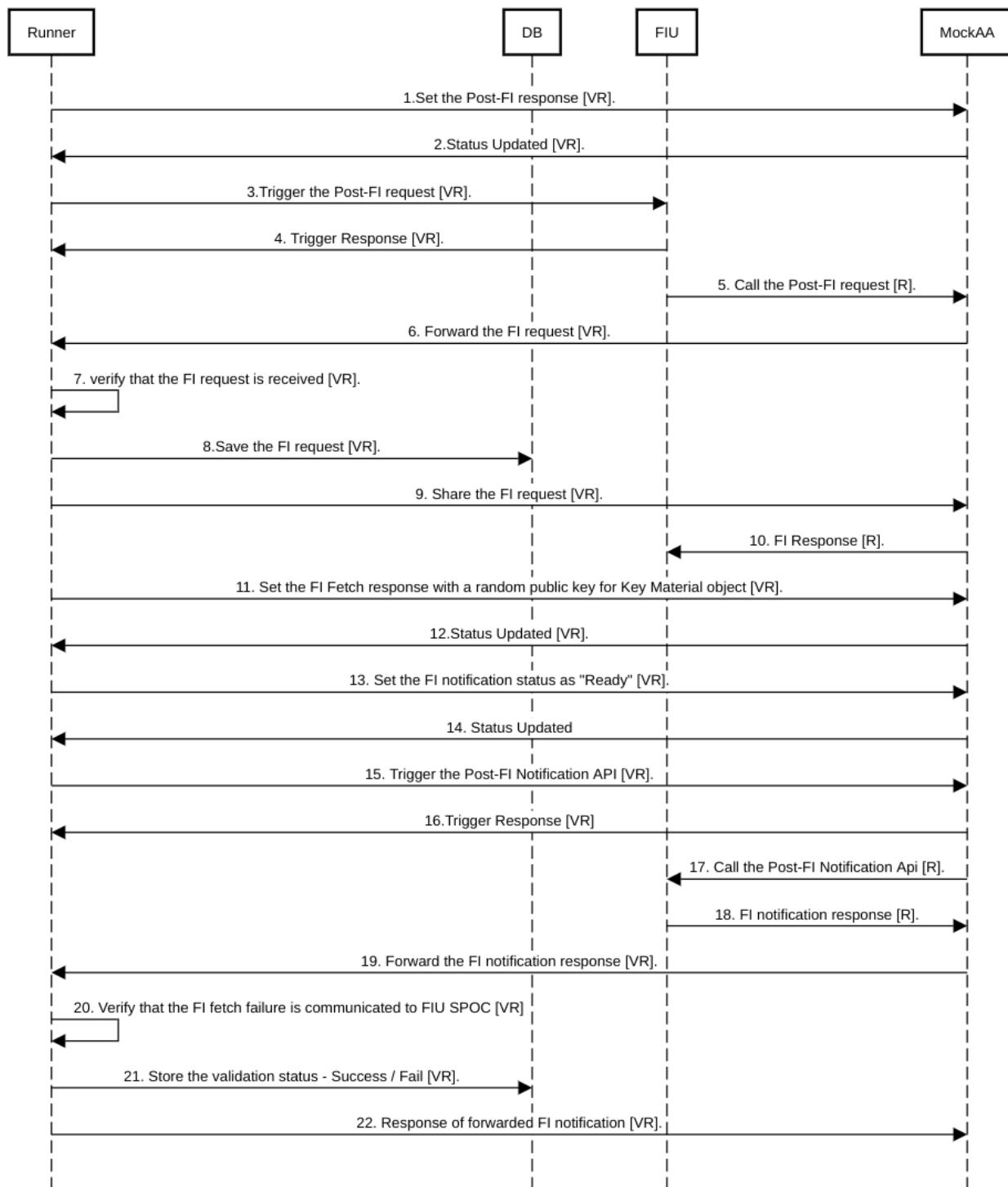
Scenario 2025_7

title FIU_2025_GET_FIFetchSessionId.feature Scenario 2025_7

participant Runner
participant DB
participant FIU
participant MockAA

Runner->MockAA: 1. Set the Post-FI response [VR].
Runner<-MockAA:2. Status Updated [VR].
Runner->FIU:3. Trigger the Post-FI request [VR].
Runner<-FIU:4. Trigger Response [VR].
FIU->MockAA:5. Call the Post-FI request [R].
MockAA ->Runner:6. Forward the FI request [VR].
Runner->Runner:7. verify that the FI request is received [VR].
Runner -> DB:8. Save the FI request [VR].
Runner->MockAA:9. Share the FI request [VR].
FIU<-MockAA:10. FI Response [R].
Runner->MockAA:11. Set the FI Fetch response with a random public key for Key Material object [VR].
Runner<-MockAA:12. Status Updated [VR].
Runner->MockAA:13. Set the FI notification status as "Ready" [VR].
Runner<-MockAA:14. Status Updated
Runner->MockAA:15. Trigger the Post-FI Notification API [VR].
Runner<-MockAA:16. Trigger Response [VR]
FIU<-MockAA:17. Call the Post-FI Notification Api [R].
FIU->MockAA:18. FI notification response [R].
Runner<-MockAA:19. Forward the FI notification response [VR].
Runner->Runner:20. Verify that the FI fetch failure is communicated to FIU SPOC [VR]
Runner -> DB:21. Store the validation status - Success / Fail [VR].
Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2025_GET _FIFetchSessionId.feature Scenario 2025_7



Scenario 2025_8

title FIU_2025_GET_FIFetchSessionId.feature Scenario 2025_8

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA: 1. Set the Post-FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the FI Fetch response with integer value for public key for KeyMaterial object [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the FI notification status as "Ready" [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

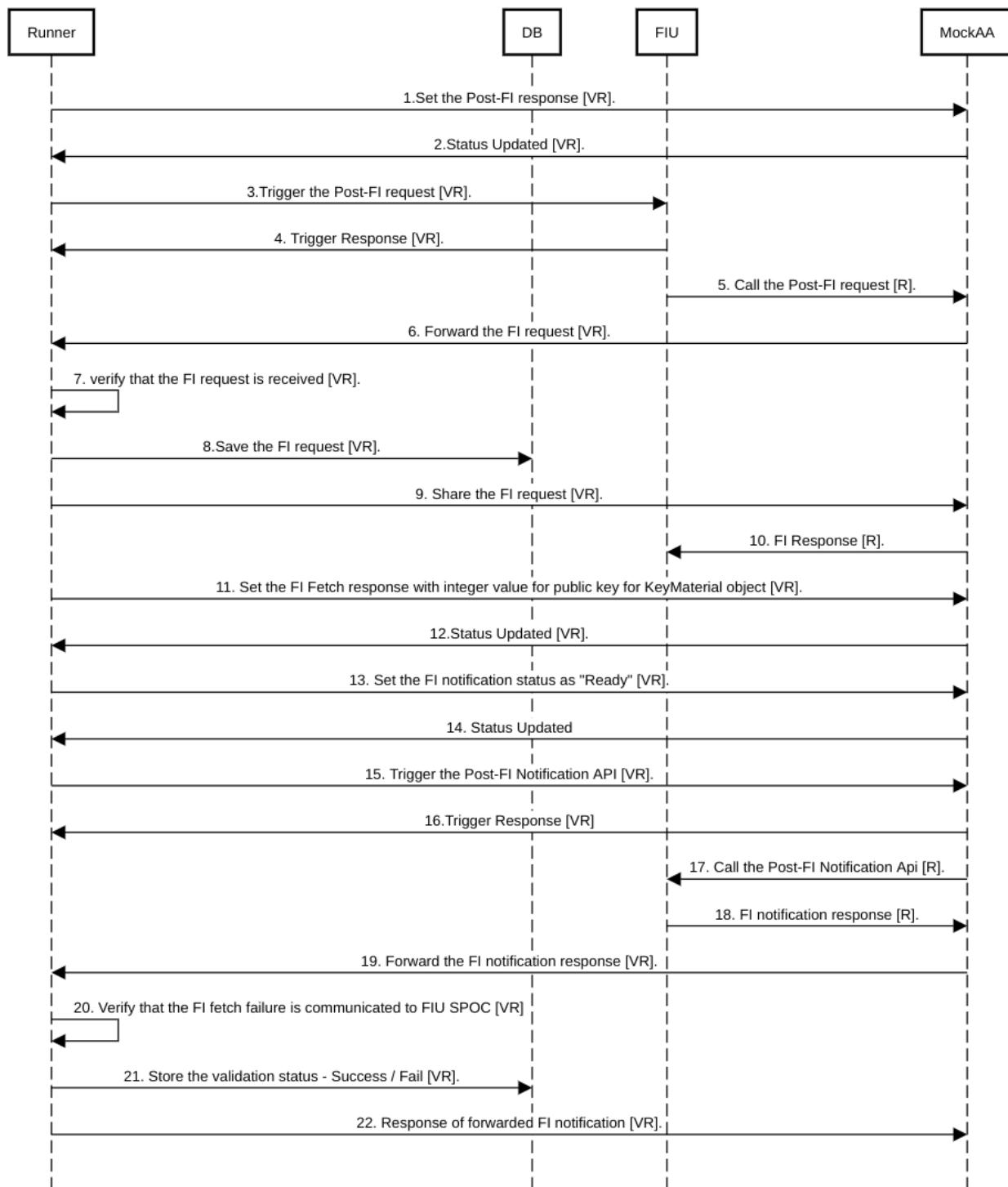
Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the FI fetch failure is communicated to FIU SPOC [VR]

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2025_GET _FIFetchSessionId.feature Scenario 2025_8



Scenario 2025_9

title FIU_2025_GET_FIFetchSessionId.feature Scenario 2025_9

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA: 1. Set the Post-FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the FI Fetch response with integer value for nonce for KeyMaterial object [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the FI notification status as "Ready" [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

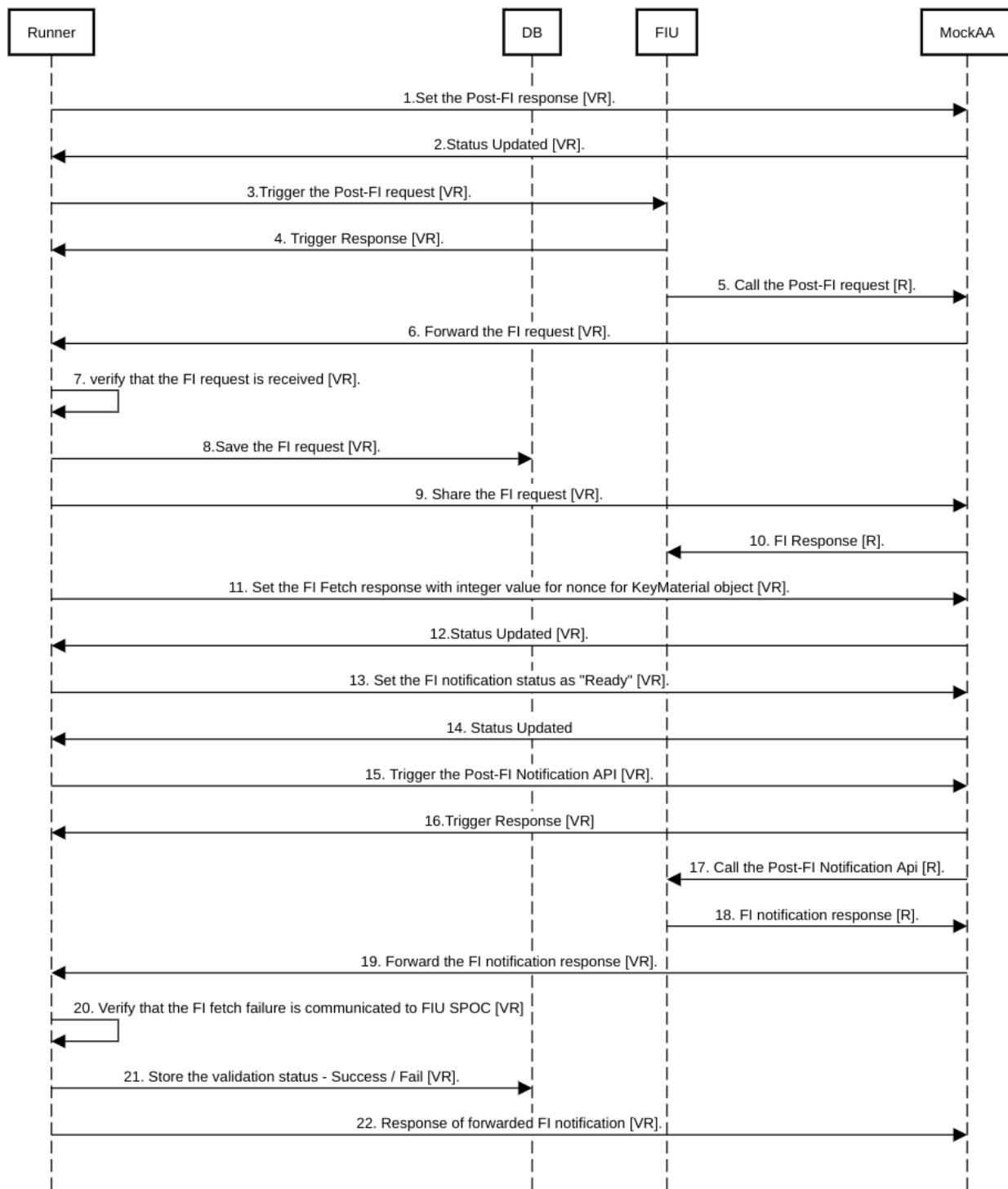
Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the FI fetch failure is communicated to FIU SPOC [VR]

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2025_GET _FIFetchSessionId.feature Scenario 2025_9



Scenario 2025_10

title FIU_2025_GET_FIFetchSessionId.feature Scenario 2025_10

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA: 1. Set the Post-FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the FI Fetch response with iRandom nounce value which is not used to generate EncryptedFI for KeyMayerial object [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the FI notification status as "Ready" [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

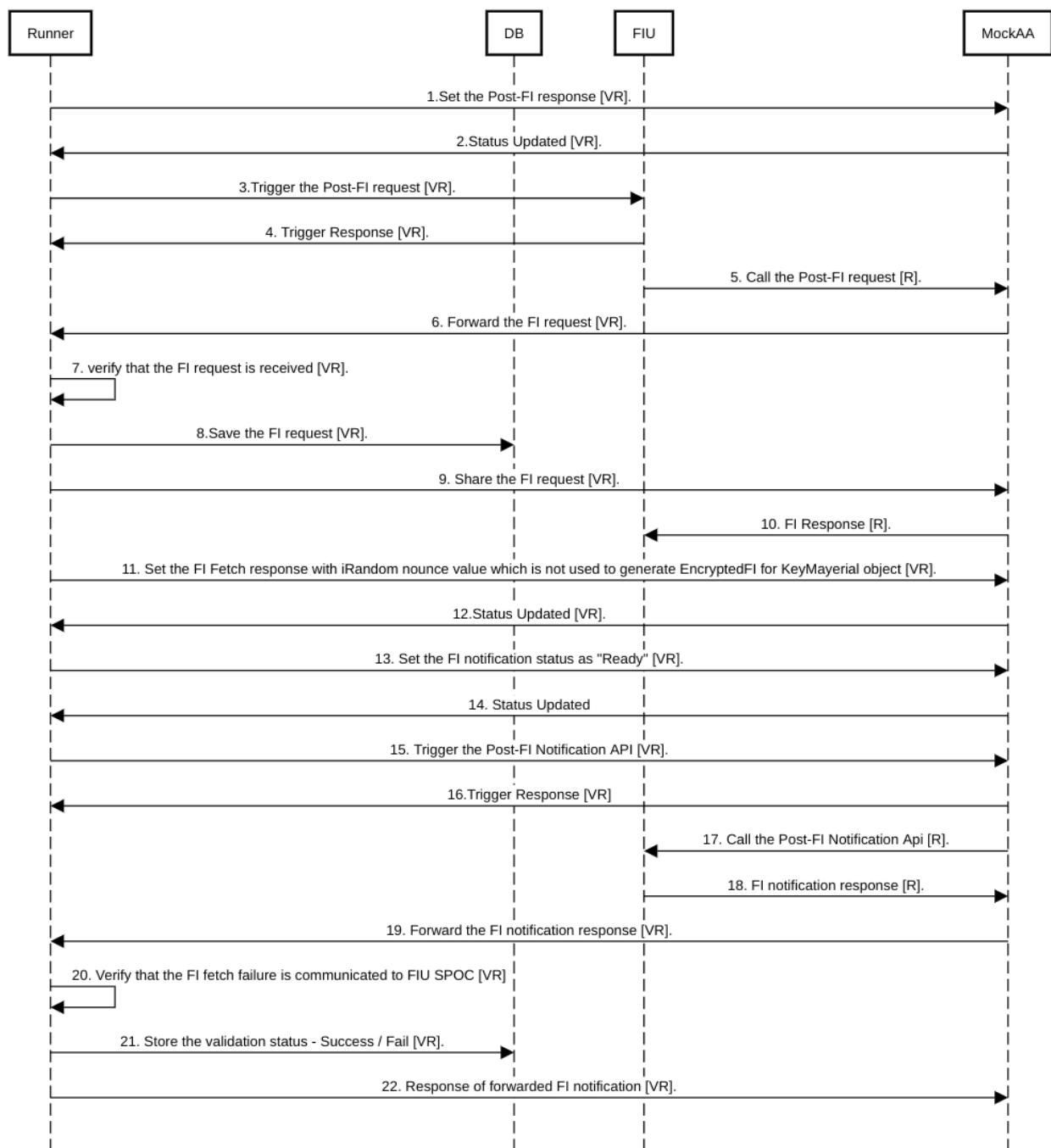
Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the FI fetch failure is communicated to FIU SPOC [VR]

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2025_GET_FIFetchSessionId.feature Scenario 2025_10



FIU_2026_GET_FIFetchSessionId.feature

Scenario 2026_1

title FIU_2026_GET_FIFetchSessionId.feature Scenario 2026_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA: 1. Set the Post-FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the FI Fetch response with invalid JWS
signatureinvalid JWS signature [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:13. Set the FI notification status as "Ready" [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

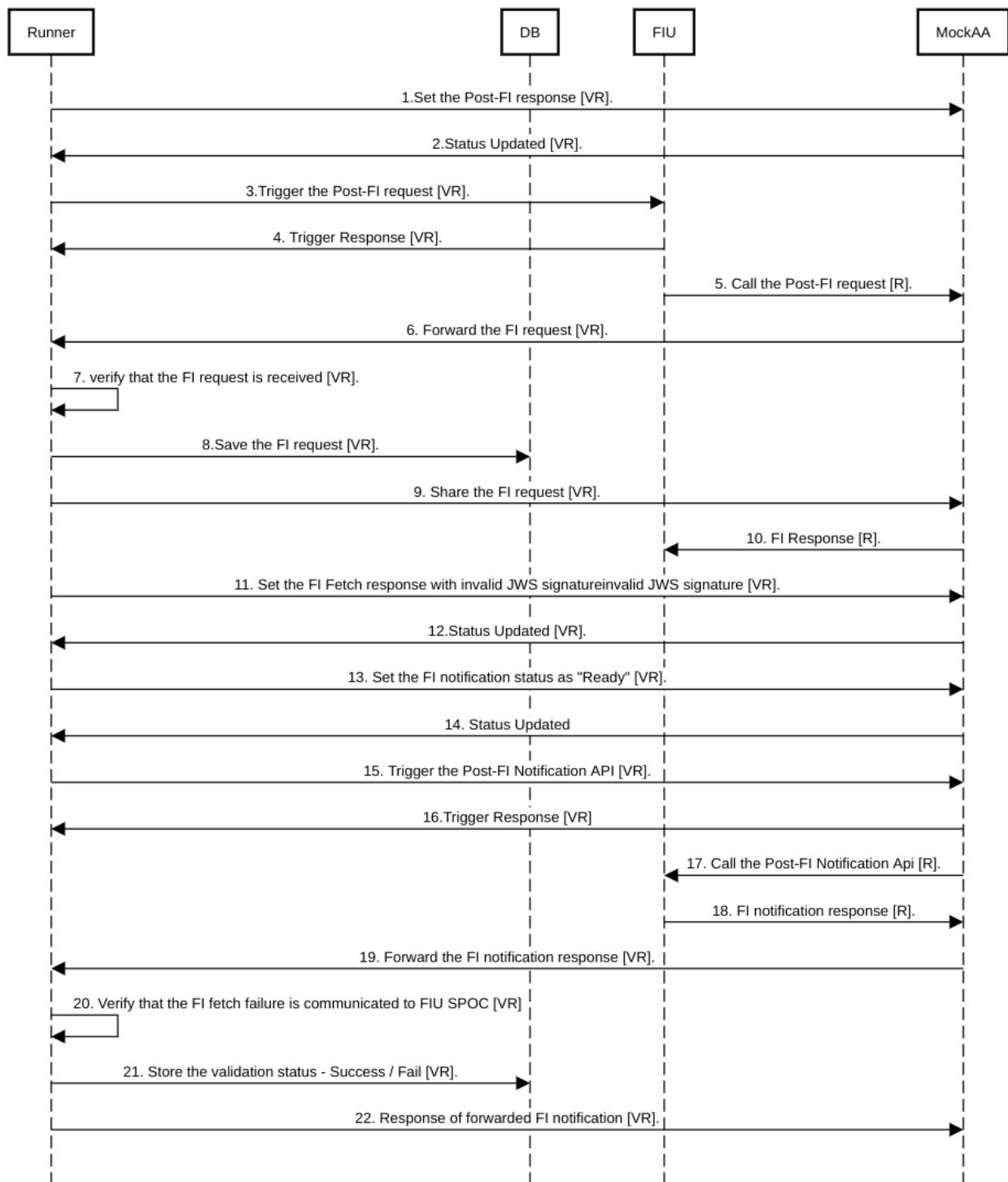
Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the FI fetch failure is communicated to FIU
SPOC [VR]

Runner -> DB:21. Store the validation status - Success / Fail [VR].

Runner->MockAA:22. Response of forwarded FI notification [VR].

FIU_2026_GET _FIFetchSessionId.feature Scenario 2026_1



FIU_2027_POST_FINotification.feature

Scenario 2027_1

title FIU_2027_POST_FINotification.feature Scenario 2027_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA:1. Set the Post FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the Get-FI fetch response [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:12. Set the FI notification status as "Ready" [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the response code 400 is displayed [VR]

Runner->Runner:21. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR]

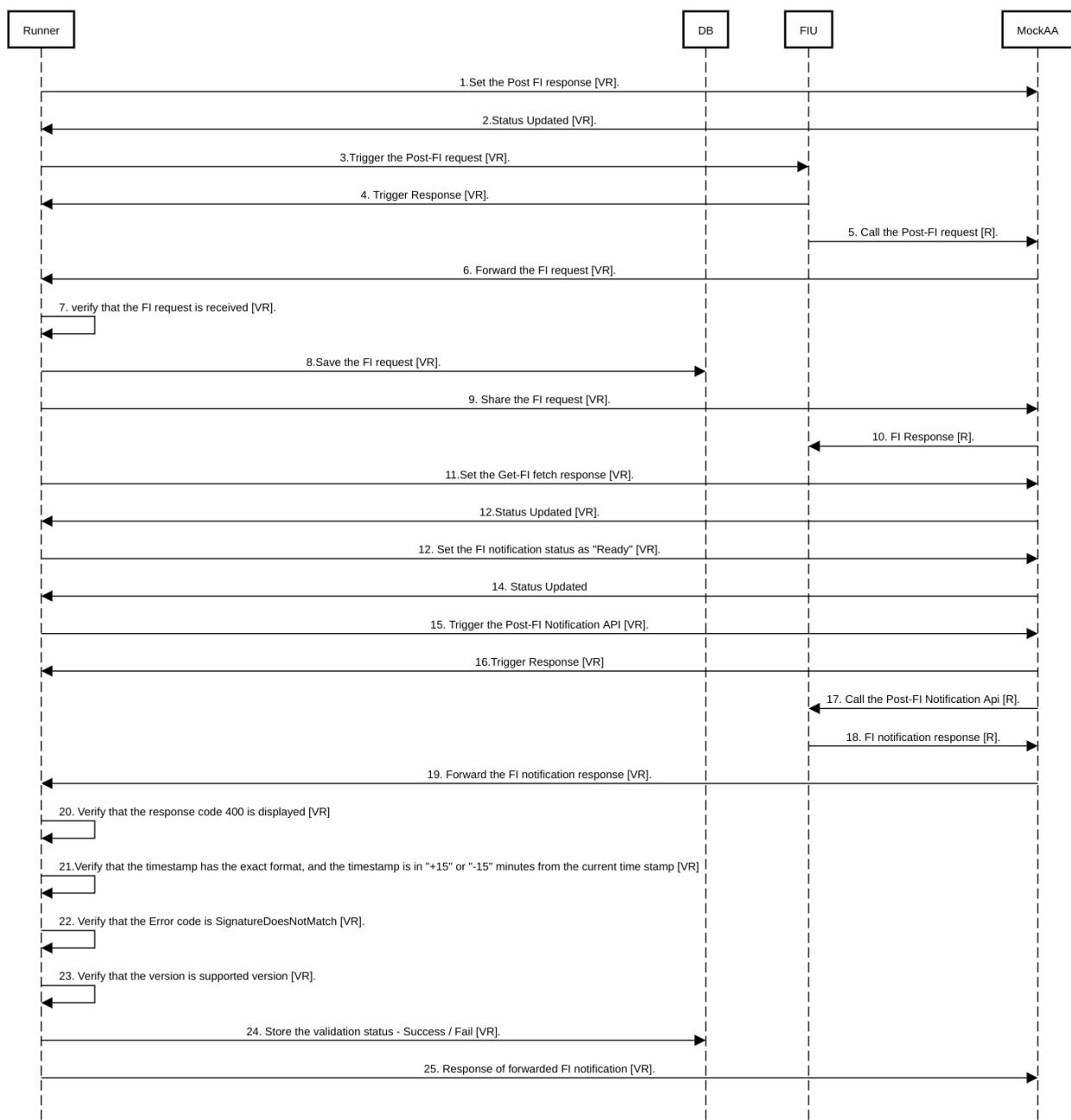
Runner->Runner:22. Verify that the Error code is SignatureDoesNotMatch [VR].

Runner->Runner:23. Verify that the version is supported version [VR].

Runner -> DB:24. Store the validation status - Success / Fail [VR].

Runner->MockAA:25. Response of forwarded FI notification [VR].

FIU_2027_POST_FINotification.feature Scenario 2027_1



FIU_2028_POST_FINotification.feature

Scenario 2028_1

title FIU_2028_POST_FINotification.feature Scenario 2028_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA:1. Set the Post FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the Get-FI fetch response [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:12. Set the FI notification status as "Active" and without the aa_api_key header [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

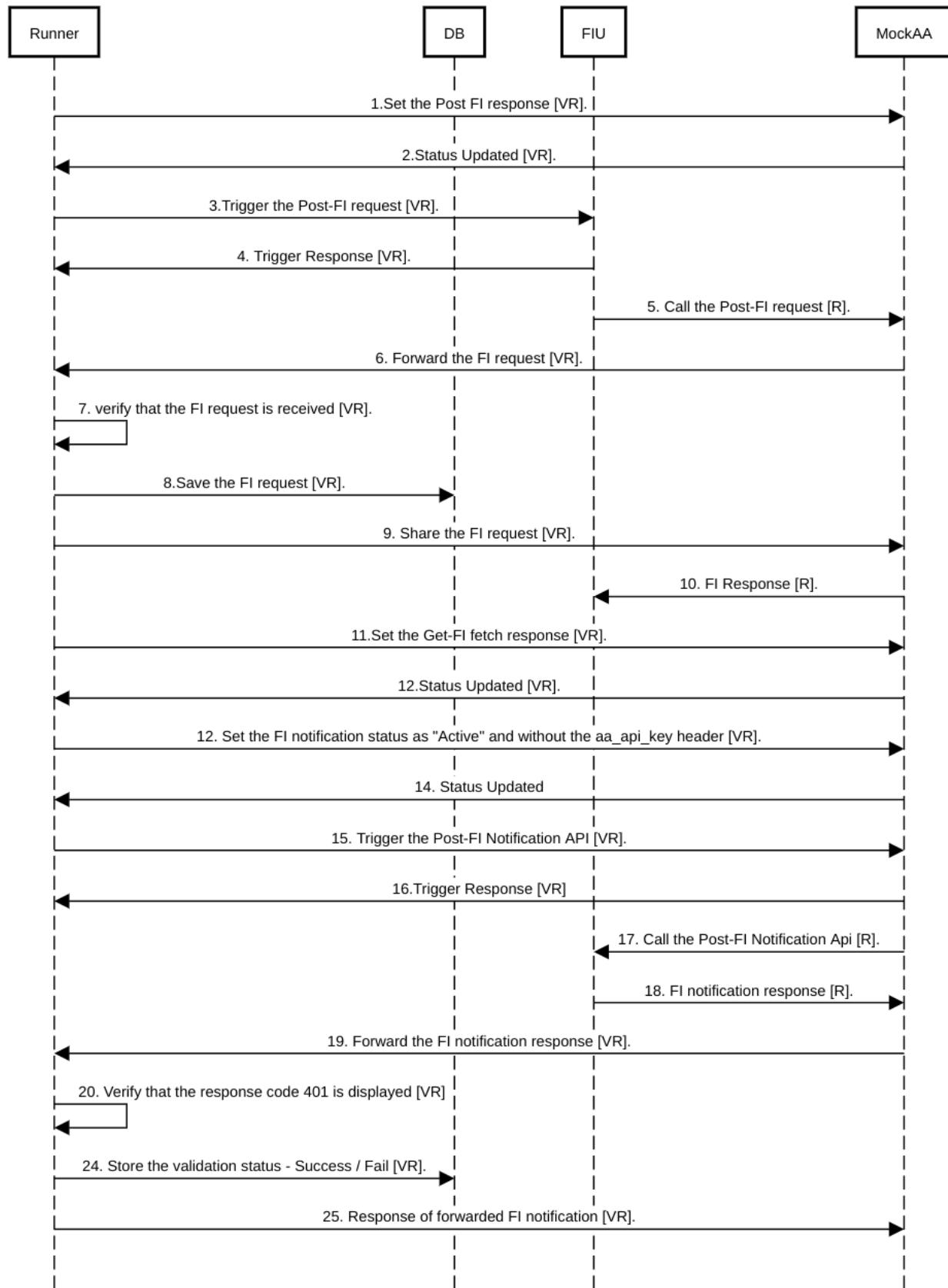
Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the response code 401 is displayed [VR]

Runner -> DB:24. Store the validation status - Success / Fail [VR].

Runner->MockAA:25. Response of forwarded FI notification [VR].

FIU_2028_POST _FINotification.feature Scenario 2028_1



Scenario 2028_2

title FIU_2028_POST_FINotification.feature Scenario 2028_2

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA:1. Set the Post FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the Get-FI fetch response [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:12. Set the FI notification status as "Active" and with aa_api_key as empty string[VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

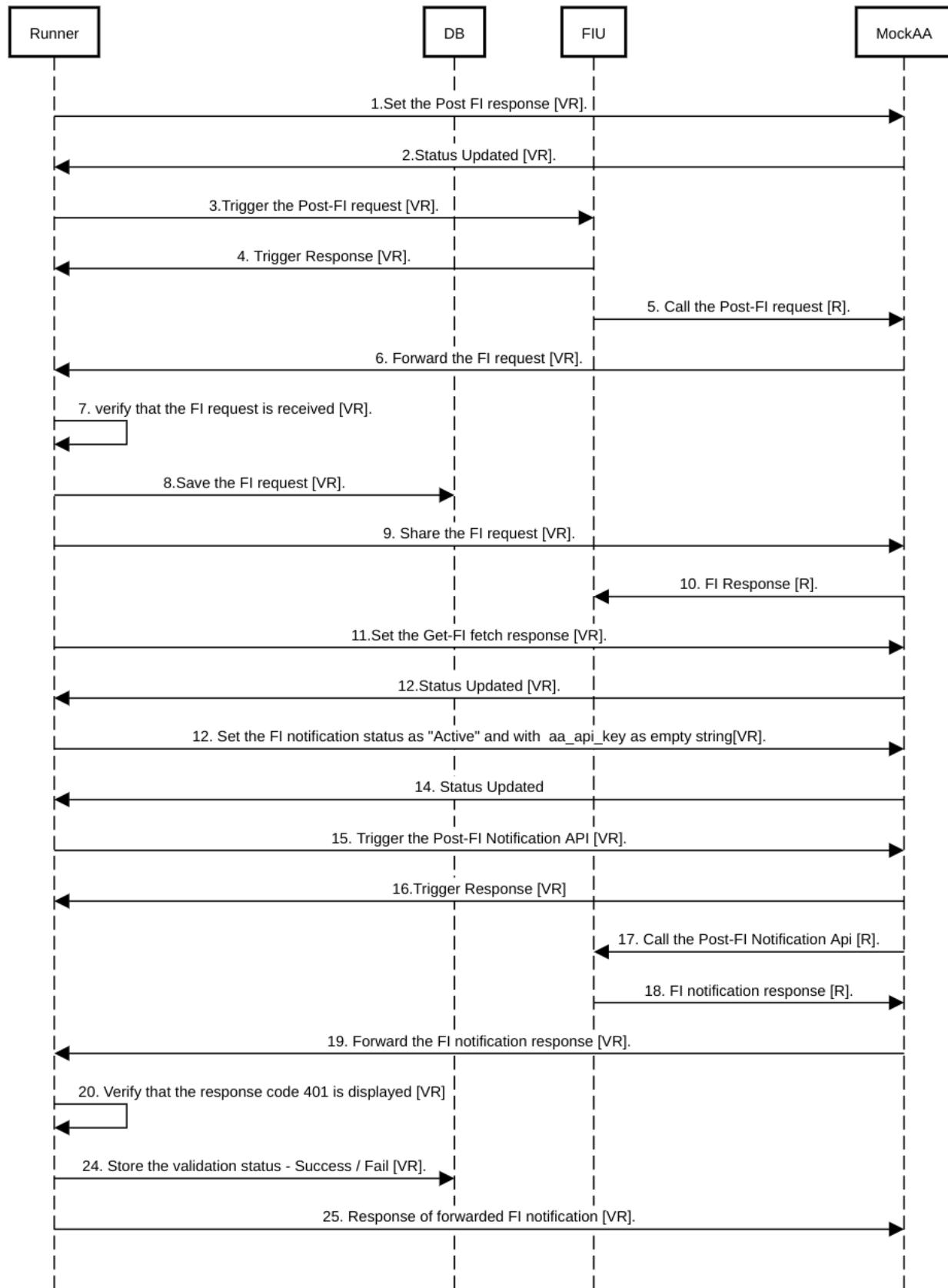
Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the response code 401 is displayed [VR]

Runner -> DB:24. Store the validation status - Success / Fail [VR].

Runner->MockAA:25. Response of forwarded FI notification [VR].

FIU_2028_POST _FINotification.feature Scenario 2028_2



Scenario 2028_3

title FIU_2028_POST_FINotification.feature Scenario 2028_3

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA:1. Set the Post FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the Get-FI fetch response [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:12. Set the FI notification status as "Active" and with aa_api_key as o [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the response code 401 is displayed [VR]

Runner -> DB:24. Store the validation status - Success / Fail [VR].

Runner->MockAA:25. Response of forwarded FI notification [VR].

Scenario 2028_4

title FIU_2028_POST_FINotification.feature Scenario 2028_4

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA:1. Set the Post FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the Get-FI fetch response [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:12. Set the FI notification status as "Active" split the api key with .(dot) and with one character added for that component on each split component [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

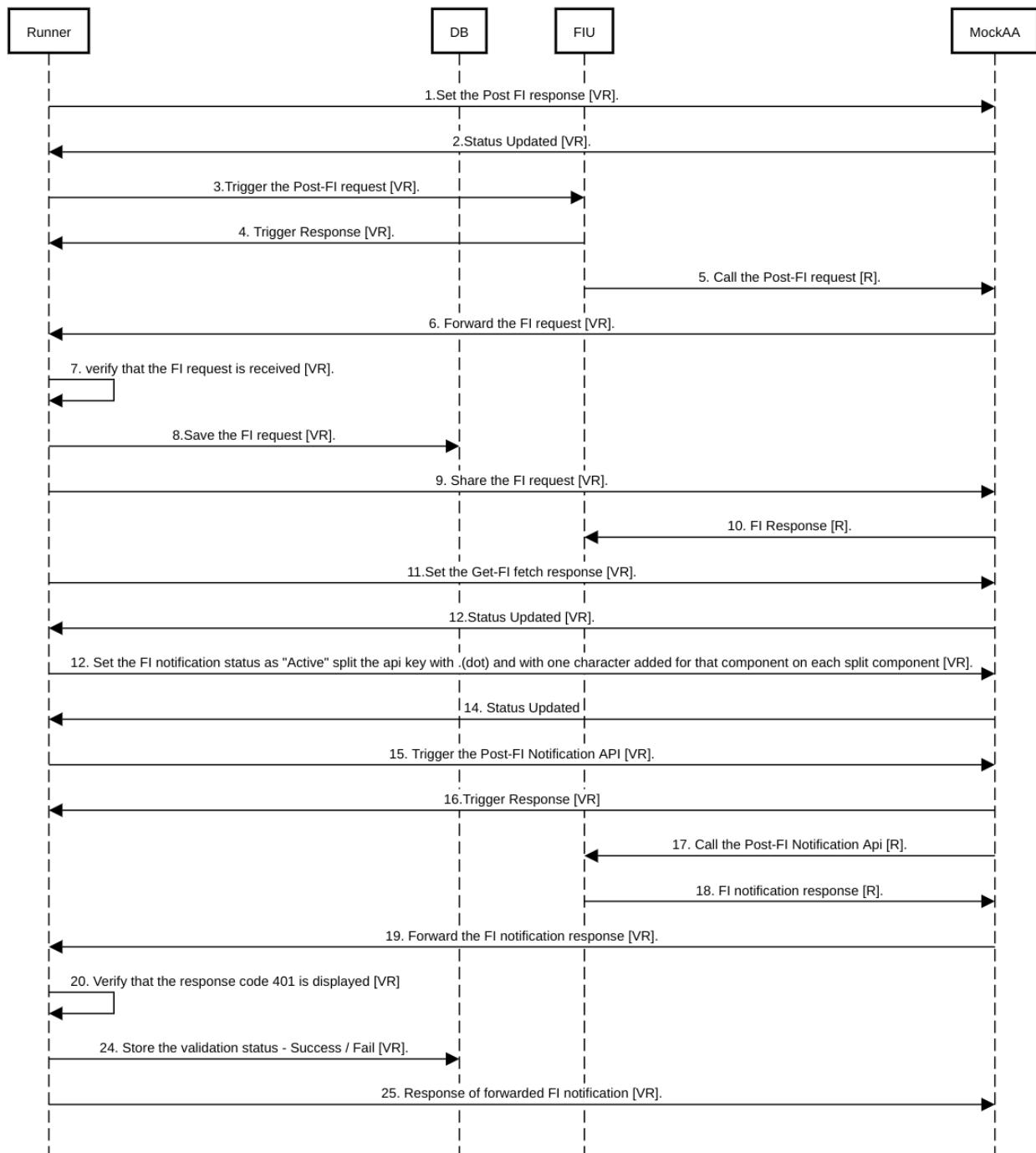
Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the response code 401 is displayed [VR]

Runner -> DB:24. Store the validation status - Success / Fail [VR].

Runner->MockAA:25. Response of forwarded FI notification [VR].

FIU_2028_POST _FINotification.feature Scenario 2028_4



Scenario 2028_4.1

title FIU_2028_POST_FINotification.feature Scenario 2028_4.1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA:1. Set the Post FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the Get-FI fetch response [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:12. Set the FI notification status as "ACTIVE", split the api key with .(dot) and with one character added for that component on each split component [VR].

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

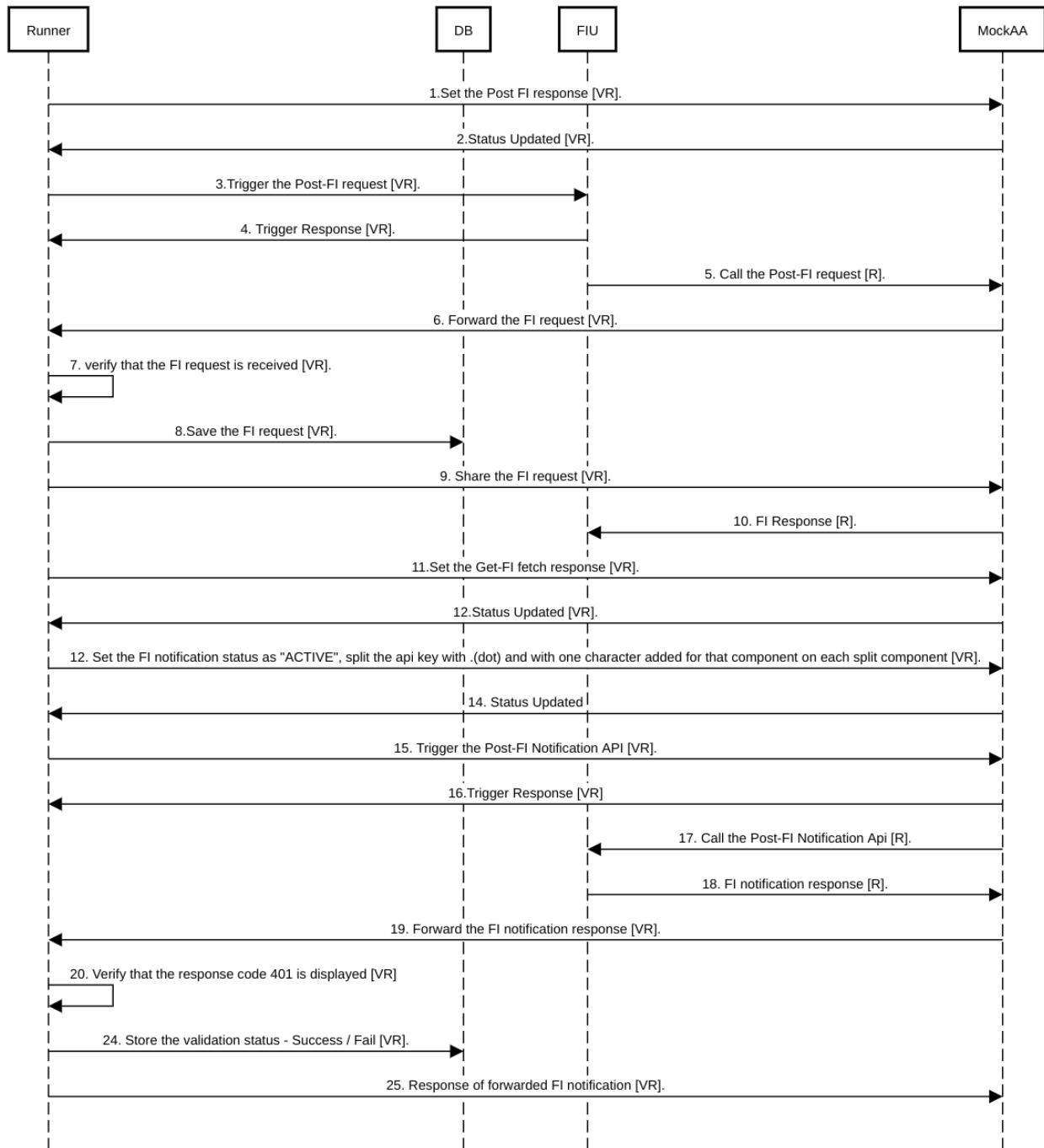
Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the response code 401 is displayed [VR]

Runner -> DB:24. Store the validation status - Success / Fail [VR].

Runner->MockAA:25. Response of forwarded FI notification [VR].

FIU_2028_POST_FINotification.feature Scenario 2028_4.1



FIU_2029_POST_FINotification.feature

Scenario 2029_1

title FIU_2029_POST_FINotification.feature 2029_1

participant Runner

participant DB

participant FIU

participant MockAA

Runner->MockAA:1. Set the Post FI response [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-FI request [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-FI request [R].

MockAA ->Runner:6. Forward the FI request [VR].

Runner->Runner:7. verify that the FI request is received [VR].

Runner -> DB:8. Save the FI request [VR].

Runner->MockAA:9. Share the FI request [VR].

FIU<-MockAA:10. FI Response [R].

Runner->MockAA:11. Set the Get-FI fetch response [VR].

Runner<-MockAA:12. Status Updated [VR].

Runner->MockAA:12. Set the FI notification status as "ACTIVE" with Alternate AA API key.

Runner<-MockAA:14. Status Updated

Runner->MockAA:15. Trigger the Post-FI Notification API [VR].

Runner<-MockAA:16. Trigger Response [VR]

FIU<-MockAA:17. Call the Post-FI Notification Api [R].

FIU->MockAA:18. FI notification response [R].

Runner<-MockAA:19. Forward the FI notification response [VR].

Runner->Runner:20. Verify that the response code 400 is displayed [VR]

Runner->Runner:21. Verify that the timestamp has the exact format, and the timestamp is in "+15" or "-15" minutes from the current time stamp [VR].

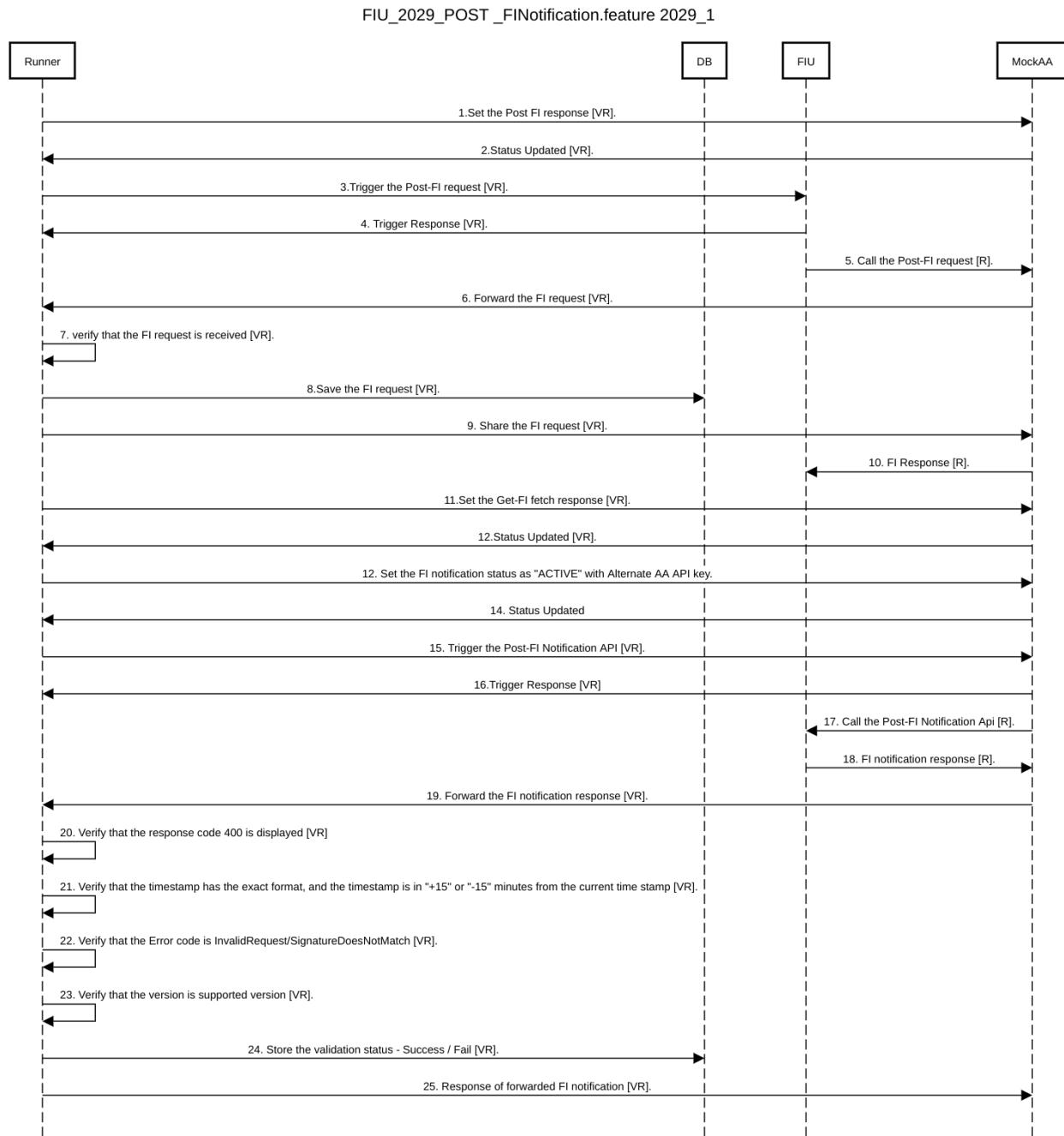
Runner->Runner:22. Verify that the Error code is

InvalidRequest/SignatureDoesNotMatch [VR].

Runner->Runner:23. Verify that the version is supported version [VR].

Runner -> DB:24. Store the validation status - Success / Fail [VR].

Runner->MockAA:25. Response of forwarded FI notification [VR].



FIU_3001_Callback API.feature

Scenario 3001_1

title FIU_3001_Callback API.feature Scenario 3001_1

participant Runner

participant DB

participant FIU

participant MockAA

VR -> Vishwaas Runner call

R -> Rebit Call

Runner->MockAA:1. Set the Post-Consent response Random Customer id [VR].

Runner<-MockAA:2. Status Updated [VR].

Runner->FIU:3. Trigger the Post-Consent API [VR].

Runner<-FIU:4. Trigger Response [VR].

FIU->MockAA:5. Call the Post-Consent API [R].

Runner<-MockAA:6. Forward the consent request [VR].

Runner->Runner:7. verify that the consent request is received [VR].

Runner->Runner:8. Validate that the consent request is for random user id [VR].

Runner -> DB:9.Save the consent request [VR].

Runner->MockAA:10. Share the consent request [VR].

FIU<-MockAA:11. Consent Response [R].

Runner->MockAA:12. Set the consent handle response with "PENDING" Status [VR].

Runner<-MockAA:13. Status Updated [VR].

Runner->MockAA:14. Set the Get-Consent request [VR].

Runner<-MockAA:15. Status Updated [VR].

Runner->MockAA:16. Set consent notification status as "Active" [VR].

Runner<-MockAA:17.Status Updated

Runner->MockAA:18. Trigger the Post-Consent Notification API [VR].

Runner<-MockAA:19.Trigger Response [VR]

FIU<-MockAA:20. Call the Post-Consent Notification Api [R].

FIU->MockAA:21. Consent notification response [R].

Runner->MockAA:22.Set the FI Response for the above consent [VR].

Runner<-MockAA:23. Status Updated
Runner->FIU:24. Trigger the Post-FI request [VR].
Runner<-FIU:25. Trigger Response [VR].
FIU->MockAA:26. Call the Post-FI request [R].
MockAA ->Runner:27. Forward the FI request [VR].
Runner->Runner:28. verify that the FI request is received [VR].
Runner -> DB:29. Save the FI request [VR].
Runner->MockAA:30. Share the FI request [VR].
FIU<-MockAA:31. FI Response [R].
Runner->MockAA:32. Set the Get-FI fetch response [VR].
Runner<-MockAA:33. Status Updated [VR].
Runner->MockAA:34. Trigger the Post-FI Notification API [VR].
Runner<-MockAA:35. Trigger Response [VR]
FIU<-MockAA:36. Call the Post-FI Notification Api [R].
FIU->MockAA:37. FI notification response [R].
Runner<-MockAA:38. Forward the FI notification response [VR].
Runner ->Runner:39. Verify that the success response is received from all the stages [VR].
Runner -> DB:40. Store the validation status - success / fail [VR].
Runner->MockAA:41. Response of forwarded FI notification [VR].

FIU_3001_Callback API.feature Scenario 3001_1



