



ANASTASIA LABS

Proof of Achievement - Milestone 1

Payment Subscription Smart Contract

Project Number 1100025

Project Manager Jonathan Rodriguez

Contents

1. Introduction	1
2. Test Suite Details	2
2.1. Test Execution Results	2
3. Managing Recurring Payments Tests	3
3.1. Test Case: Initiating a Subscription (succeed_initiate_subscription) .	4
3.2. Test Case: Terminate Subscription (succeed_terminate_subscription)	5
3.3. Test Case: Extend Subscription (succeed_extend_subscription)	6
3.4. Test Case: Unsubscribe (succeed_unsubscribe)	7
3.5. Test Case: Withdrawing Subscription Fees by Merchant (succeed_merchant_withdraw)	8
3.6. Test Case: Withdrawing Subscription Fees by Subscriber (succeed_subscriber_withdraw)	9
4. User Workflow for Managing Recurring Payments	10
5. Conclusion	11

Payment Subscription Smart Contract

1. Introduction

This document presents comprehensive evidence of the successful implementation and testing of the Payment Subscription Smart Contract addressing the effortless management of recurring payments

Each section provides detailed insights into the functionality, security, and usability of the smart contract, demonstrating its readiness for real-world application.

Our rigorous testing suite demonstrates the contract's ability to manage recurring payments effectively and with ease.

2. Test Suite Details

The test suite for the Payment Subscription Smart Contract consists of thirteen critical test cases, each designed to verify specific aspects of the contract's functionality.

2.1. Test Execution Results

```
Testing ...

payment_subscription/tests/account_multi_validator
PASS [mem: 347249, cpu: 137903361] succeed_create_account
PASS [mem: 206854, cpu: 79875639] succeed_delete_account
PASS [mem: 477264, cpu: 179987133] succeed_update_account
PASS [mem: 289679, cpu: 112928610] succeed_remove_account
4 tests | 4 passed | 0 failed

payment_subscription/tests/payment_multi_validator
PASS [mem: 719170, cpu: 277679281] succeed_initiate_subscription
PASS [mem: 358259, cpu: 134424664] succeed_terminate_subscription
PASS [mem: 886708, cpu: 338587008] succeed_extend_subscription
PASS [mem: 712513, cpu: 273599426] succeed_unsubscribe
PASS [mem: 765639, cpu: 289918106] succeed_merchant_withdraw
PASS [mem: 598455, cpu: 229728929] succeed_subscriber_withdraw
6 tests | 6 passed | 0 failed

payment_subscription/tests/service_multi_validator
PASS [mem: 416801, cpu: 163250864] success_create_service
PASS [mem: 560773, cpu: 210655896] success_update_service
PASS [mem: 596384, cpu: 230611796] success_remove_service
3 tests | 3 passed | 0 failed

Summary 13 checks, 0 errors, 0 warnings
```

Figure 1: All Payment Subscription Tests

This test validates the contract's ability to initiate a new subscription. It demonstrates:

- Correct setup of subscription parameters
- Proper creation of the Payment Datum
- Accurate handling of inputs and outputs
- Successful minting of the Payment NFT

3. Managing Recurring Payments Tests

This process comprises of six checks:

- succeed_initiate_subscription
- succeed_terminate_subscription
- succeed_extend_subscription
- succeed_unsubscribe
- succeed_merchant_withdraw
- succeed_subscriber_withdraw

3.1. Test Case: Initiating a Subscription (succeed_initiate_subscription)

```

Testing ...
payment_subscription/tests/payment_multi_validator
PASS [mem: 9148552, cpu: 480309676] succeed_initiate_subscription
- with traces
  test: Initiating a New Subscription
  -----
  Step 1: Setting up the subscription
  -----
  Service Currency Symbol:
  | h'8FA726C3C491658188E6FF558CB1A1C4F8FDC2E9F26A9A16F48B8A6'
  Account Currency Symbol:
  | h'F83D635C7C873D189E9BF4A6A4B4F25190D296FFD8D9AC605A218'
  Subscription Fee: (lowface)
  | 1000000000
  Subscription Period: (days)
  | 30
  Penalty Fee: (lowface)
  | 1000000
  Step 2: Creating Payment Datum
  -----
  Service NFT:
  | h'000643808136752529784C15E638DA2A27F81C08C9C8B92277913ABDC40A86D4'
  Account NFT:
  | h'0000E14081A4028049AFC47FB302ED59459582CB0D0545B7C6AC22504E98492'
  Subscription Start:
  | 1000000
  Subscription End:
  | 2593000000
  Step 3: Preparing Inputs
  -----
  Account Input:
  | 121([ 121([ 121([ h'EE155AC9C4029207ACB6A8FFB9C9CDD273C81648FF1149EF368CEA6' ]), 1 ]), 121([ 121([ 121([ h'F83D635C7C873D189E9BF4A6A4B4F25190D296FFD8D9AC605A218' ]), 121([ ]), [ h'': 4000000 ] ), h'F83D635C7C873D189E9BF4A6A4B4F25190D296FFD8D9AC605A218' : [ h'0000E14081A4028049AFC47FB302ED59459582CB0D0545B7C6AC22504E98492' : 1 ] ) ], 123([ 121([ ]), 122([ ]))])
  Service Input:
  | 121([ 121([ 121([ h'EE155AC9C4029207ACB6A8FFB9C9CDD273C81648FF1149EF368CEA6' ]), 1 ]), 121([ 121([ 121([ h'8FA726C3C491658188E6FF558CB1A1C4F8FDC2E9F26A9A16F48B8A6' ]), 122([ ]), [ h'': 4000000 ] ), h'8FA726C3C491658188E6FF558CB1A1C4F8FDC2E9F26A9A16F48B8A6' : [ h'000643808136752529784C15E638DA2A27F81C08C9C8B92277913ABDC40A86D4' : 1 ] ) ], 123([ 121([ 121([ h'': h' ]), 1000000000, 121([ h'': h' ]), 1000000, 2592000000, 10, 2000000, 122([ ]))]), 122([ ]))])
  Step 4: Preparing Outputs
  -----
  User Output:
  | 121([ 121([ 121([ h'888D757A05B8EDF3720B3F0CF6C962A660B61A265F6418E1FFED' ]), 122([ ]), [ h'': 1000000000 ] ), h'F83D635C7C873D189E9BF4A6A4B4F25190D296FFD8D9AC605A218' : [ h'0000E14081A4028049AFC47FB302ED59459582CB0D0545B7C6AC22504E98492' : 1 ] ) ], 123([ 121([ ]), 122([ ]))])
  Payment Output:
  | 121([ 121([ 121([ h'873E4FE9E41E92A9118BA3EC3FF4782EFC80F244FB75C879F8A432' ]), 122([ ]), [ h'': 1000000000 ] ), h'873E4FE9E41E92A9118BA3EC3FF4782EFC80F244FB75C879F8A432' : [ h'015F478F81A300F82B183E9858E3A877725785678981507D8920A190AAA720EA' : 1 ] ) ], 123([ 121([ 121([ h'000643808136752529784C15E638DA2A27F81C08C9C8B92277913ABDC40A86D4' , h'0000E14081A4028049AFC47FB302ED59459582CB0D0545B7C6AC22504E98492' , 121([ h'': h' ]), 1000000000, 1000000, 25921000000, 2592000000, 1000000000, 10, 500000, 121([ h'': h' ]), 100002000000 ]), 122([ ]))])
  Step 5: Execution Result
  -----
  Subscription Successfully Initiated!
  -----
  Test Completed!
  token name
  | h'0000E14081A4028049AFC47FB302ED59459582CB0D0545B7C6AC22504E98492'
  token quantity
  | 1
  token name
  | h'015F478F81A300F82B183E9858E3A877725785678981507D8920A190AAA720EA'
  token quantity
  | 1
  -----
  tests | 1 passed | 0 failed
  Summary | check | 0 errors | 0 warnings
  
```

Figure 2: Succeed Initialize Subscription Test

This test validates the contract's ability to initiate a new subscription. It demonstrates:

- Correct setup of subscription parameters
- Proper creation of the Payment Datum
- Accurate handling of inputs and outputs
- Successful minting of the Payment NFT

tion)

```

Testing ...
Payment_subscription/tests/payment_multi_validator
PASS [mm: 1016935, cpu: 5301512491] succeed terminate_subscription
- with traces
  Test Terminating a Subscription
  -----
  Step 1: Subscription Details
  -----
  Service Currency Symbol:
  h'BF4726C3C149165810866FF558C81A1CA40FC2E9F26A9A16F48BAE*'
  Account Currency Symbol:
  h'F83605C7CDB573D10E98F4A64A84F25198D2968D8D89C685A218*'
  Payment Currency Symbol:
  h'47314FE4E41E9249118BA3C53FF4782EFC8CF244F875C879F8A32*'
  Original Subscription Start:
  1000000
  Original Subscription End:
  2593000000
  Termination Time (mid-subscription):
  1297000000
  Step 2: Calculating Refund and Penalty
  -----
  Total Subscription Time:
  2592000000
  Time Elapsed:
  1296000000
  Original Payment Amount:
  1000000000
  Refund Amount:
  500000000
  Penalty Applied:
  1000000
  Step 3: Processing Termination
  -----
  Payment NFT to be burned:
  h'0A402B849AFDC47B302E9549592ACB0D545B7C6422504E984922B3068A*'
  Refund Output:
  121([ _ 121([ _ 121([ _ h'F83605C7CDB573D10E98F4A64A84F25198D2968D8D89C685A218' ] , 122([ ] ) , 122([ ] ) , 121([ _ 121([ _ 121([ _ h'BF4726C3C149165810866FF558C81A1CA40FC2E9F26A9A16F48BAE' ] , 122([ ] ) , 122([ ] ) , 121([ _ 121([ _ 121([ _ h'' : 5000000000 ] , 123([ _ 121([ ] ) , 122([ ] ) ) ) ) ) )
  121([ _ 121([ _ 121([ _ h'BF4726C3C149165810866FF558C81A1CA40FC2E9F26A9A16F48BAE' ] , 122([ ] ) , 122([ ] ) , 121([ _ 121([ _ 121([ _ h'' : 10000000 ] , 123([ _ 121([ _ 121([ _ h'000014A081E1B381AB4A909C668F292174345482D81D06AA29520E004D837057' , h'000014A0815F47F8F1A308F02B1B3E9858E3A877725785678981507F
  820A10' ] , 121([ _ h'', h'', h'', 1000000000 ] ) , 122([ ] ) ) ) ) )
  Step 4: Verifying Transaction
  -----
  Transaction Inputs:
  [ _ 121([ _ 121([ _ 121([ _ h'E155ACDC4029207ACD6A8F18C9CCD273C81648F1149F368EAE*' ] , 122([ ] ) , 122([ _ 121([ _ 121([ _ h'47314FE4E41E9249118BA3C53FF4782EFC8CF244F875C879F8A32' ] , 122([ ] ) , 122([ ] ) , 121([ _ 121([ _ 121([ _ h'' : 4000000 ] , h'F83605C7CDB573D10E98F4A64A84F25198D2968D8D89C685A218' ] , 122([ ] ) , 122([ ] ) , 121([ _ 121([ _ 121([ _ h'BF4726C3C149165810866FF558C81A1CA40FC2E9F26A9A16F48BAE' ] , 122([ ] ) , 122([ ] ) , 121([ _ 121([ _ 121([ _ h'01A402B849AFDC47B302E9549592ACB0D545B7C6422504E984922B3068A' : 1 ] , 123([ _ 121([ _ 121([ _ h'000014A081E1B381AB4A909C668F292174345482D81D06AA29520E004D837057' , h'000014A0815F47F8F1A308F02B1B3E9858E3A877725785678981507F02B0A10' ] , 121([ _ h'', h'', h'', 1000000000, 1000000, 2593000000, 2592000000, 1000000000, 1, 5000000, 121([ _ h'', h'', h'', 1000000, 20000000 ] ) , 122([ ] ) ) ) ) ) )
  Transaction Outputs:
  [ _ 121([ _ 121([ _ 121([ _ h'F83605C7CDB573D10E98F4A64A84F25198D2968D8D89C685A218' ] , 122([ ] ) , 122([ ] ) , 121([ _ 121([ _ 121([ _ h'' : 5000000000 ] , 123([ _ 121([ ] ) , 122([ ] ) , 121([ _ 121([ _ 121([ _ h'BF4726C3C149165810866FF558C81A1CA40FC2E9F26A9A16F48BAE' ] , 122([ ] ) , 122([ ] ) , 121([ _ 121([ _ 121([ _ h'' : 1000000 ] , 123([ _ 121([ _ 121([ _ h'000014A081E1B381AB4A909C668F292174345482D81D06AA29520E004D837057' , h'000014A0815F47F8F1A308F02B1B3E9858E3A877725785678981507F02B0A10' ] , 121([ _ h'', h'', h'', 1000000000 ] ) , 122([ ] ) ) ) ) ) )
  Burned Tokens:
  [ _ h'', h'', h'' : 0 ] , h'47314FE4E41E9249118BA3C53FF4782EFC8CF244F875C879F8A32' : [ _ h'01A402B849AFDC47B302E9549592ACB0D545B7C6422504E984922B3068A' : - 1 ] )
  Step 5: Execution Result
  -----
  Subscription Successfully Terminated
  -----
  Test completed!

```

1 tests | 1 passed | 0 failed

Summary 1 check, 0 errors, 0 warnings

Figure 3: Succeed Terminate Subscription Test

This test verifies the contract's ability to handle early termination, applying appropriate refunds and penalties.

3.3. Test Case: Extend Subscription (succeed_extend_subscription)

```

Testing ...
payment_subscription/tests/payment_multi_validator
PASS [mem: 12026479, cpu: 678032546] succeed extend_subscription
- with traces
  test: Extending an Existing Subscription
  -----
  Step 1: Current Subscription Details
  -----
  Service Currency Symbol:
  h'8FA726C3C1491658108E6F550CB1A1C4F8DC29F26A0A16F48B8E'
  Account Currency Symbol:
  h'FB3D635C7C8573D1B9E98F4A6A04F25190D29B6F08D94C695A218'
  Current Subscription Start:
  1000000
  Current Subscription End:
  259200000
  Current Subscription Fee: (lovelace)
  100000000
  Step 2: Extension Details
  -----
  Extension Period: (days)
  30
  New Subscription End:
  518500000
  Additional Fee for Extension: (lovelace)
  10000000
  Step 3: Updating Payment Datum
  -----
  Original Payment Datum:
  121([_ h'00064380013672529784C15E638DA2A27B1C08C9C8B9227791348DC40A80D4', h'000DE1A0015F478F1A308F82B1B3E958E3A07772578567898150708920A190', 121([_ h'', h'']), 100000000, 1000000, 259200000, 259200000, 10000000, 1, 500000, 121([_ h'', h'']), 1000000, 200000
  Updated Payment Datum:
  121([_ h'00064380013672529784C15E638DA2A27B1C08C9C8B9227791348DC40A80D4', h'000DE1A0015F478F1A308F82B1B3E958E3A07772578567898150708920A190', 121([_ h'', h'']), 110000000, 1000000, 518500000, 259200000, 10000000, 2, 500000, 121([_ h'', h'']), 1000000, 200000
  Step 4: Verifying Transaction
  -----
  Transaction Inputs:
  [ 121([_ 121([_ 121([_ h'EE155ACE9C40292074C8A6F8C9CDD273C81648F1149EF368CEAE', 1)), 121([_ 121([_ 121([_ h'FB3D635C7C8573D1B9E98F4A6A04F25190D29B6F08D94C695A218', 122([_ ])), { h'': { h'': 4000000 }, h'FB3D635C7C8573D1B9E98F4A6A04F25190D29B6F08D94C695A218': { h'000DE1A0015F478F1A308F82B1B3E958E3A07772578567898150708920A190': 1 } }, 122([_ 121([_ ])), 122([_ ])), { h'': { h'': 4000000 }, h'FB3D635C7C8573D1B9E98F4A6A04F25190D29B6F08D94C695A218': { h'000DE1A0015F478F1A308F82B1B3E958E3A07772578567898150708920A190': 1 } }, 122([_ 121([_ 121([_ h'00064380013672529784C15E638DA2A27B1C08C9C8B9227791348DC40A80D4', h'000DE1A0015F478F1A308F82B1B3E958E3A07772578567898150708920A190', 122([_ h'', h'']), 100000000, 1000000, 259200000, 259200000, 10000000, 1, 500000, 121([_ h'', h'']), 1000000, 2000000))), 122([_ ]))]
  Transaction Outputs:
  [ 121([_ 121([_ 121([_ h'642206114F534B29A02970824406F9F21DE30CA5CE080A587C4A02', 122([_ ])), { h'': { h'': 7000000 }, 123([_ 121([_ ])), 122([_ ])), 121([_ 121([_ 121([_ h'873E4FE9E41E9249118BA3E53F74782FC8C8F244F875C87F8BA32', 122([_ ])), { h'': { h'': 120000000 }, h'873E4FE9E41E9249118BA3E53F74782FC8C8F244F875C87F8BA32': { h'00064380013672529784C15E638DA2A27B1C08C9C8B9227791348DC40A80D4': 1 } }, 122([_ 121([_ 121([_ h'00064380013672529784C15E638DA2A27B1C08C9C8B9227791348DC40A80D4', h'000DE1A0015F478F1A308F82B1B3E958E3A07772578567898150708920A190', 122([_ h'', h'']), 110000000, 1000000, 518500000, 259200000, 10000000, 2, 500000, 121([_ h'', h'']), 1000000, 2000000))), 122([_ ]))]
  Reference Inputs:
  [ 121([_ 121([_ 121([_ h'8FA726C3C1491658108E6F550CB1A1C4F8DC29F26A0A16F48B8E', 1)), 121([_ 121([_ 121([_ h'8FA726C3C1491658108E6F550CB1A1C4F8DC29F26A0A16F48B8E', 122([_ ])), { h'': { h'': 4000000 }, h'8FA726C3C1491658108E6F550CB1A1C4F8DC29F26A0A16F48B8E': { h'00064380013672529784C15E638DA2A27B1C08C9C8B9227791348DC40A80D4': 1 } }, 122([_ 121([_ 121([_ h'00064380013672529784C15E638DA2A27B1C08C9C8B9227791348DC40A80D4', h'000DE1A0015F478F1A308F82B1B3E958E3A07772578567898150708920A190', 122([_ h'', h'']), 100000000, 1000000, 259200000, 259200000, 10000000, 1, 500000, 121([_ h'', h'']), 1000000, 2000000))), 122([_ ]))]
  Step 5: Execution Result
  -----
  Subscription Successfully Extended!
  -----
  Test Completed!
  Token Name:
  h'01A0208049FDC47FB302E5945958DC6D054587C6AC22504E984922830968A'
  Token Quantity:
  1
  -----
  Summary 1 check, 0 errors, 0 warnings
  
```

Figure 4: Succeed Extend Subscription Test

This test demonstrates the contract's ability to extend an existing subscription, showcasing the flexibility offered to subscribers. It shows:

- Accurate calculation of the new subscription end date
- Correct fee adjustment for the extension
- Proper updating of the Payment Datum
- Successful execution of the extension transaction

3.4. Test Case: Unsubscribe (succeed_unsubscribe)

```

- testing .....
- payment_subscription/Tests/payment_multi_validator =====
PASS (mm: 10944241, cpu: 566452077) - success_unsubscribe
- with traces
| Test: Unsubscribing from a Service
| -----
| Step 1: Current Subscription Details
| -----
| Original Subscription Fee: (lowcase)
| 1000000000
| Subscription period: (days)
| 30
| Unsubscribe details:
| Time elapsed: (days)
| 15
| Refund Amount: (lowcase)
| 500000000
| Penalty Fee: (lowcase)
| 100000000
| Refunded to user:
| 100000000
| Penalty retained:
| 1000000
| Step 2: Unsubscribe Process
| -----
| Time of Unsubscription:
| 1000000
| Refund Amount:
| 500000000
| Penalty Amount:
| 1000000
| Step 3: Verifying Outputs
| -----
| Refund Output:
| 121([ _ 122([ _ h'F'B3D635C7CB573D1B9E9BF4A6A4B4F25190D2906FDB0C9AC605A218' ]), 122([ ])), [ _ h'': 600000000 } ], 123([ _ 121([ ])), 122([ ]))
| Penalty Output:
| 121([ _ 121([ _ h'8784FE49E41E924911BBA3C53FF4782EFCB0F244F875C9F9BA32' ]), 122([ ])), [ _ h'': 101000000 ], [ _ h'8784FE49E41E924911BBA3C53FF4782EFCB0F244F875C9F9BA32' : [ _ h'01A02B8049FDC47F830E259459582CB0054587C6AC22504E984922830968A' : 1 ] ] ), 1
23([ _ 121([ _ h'000E1400136752529784C15E63B0A2A27F81C00C8B9227913A0C40A80D4' ], 121([ _ h'': h''), 100000000)), 122([ ]))
| Step 4: Validating Transaction
| -----
| Transaction Inputs:
| [ _ 121([ _ 121([ _ h'EE155AC19C40920207ACB0AFBF8C0CDD273C8168F31149F36BC1A6E' ]), 1)), 121([ _ 121([ _ 122([ _ h'F'B3D635C7CB573D1B9E9BF4A6A4B4F25190D2906FDB0C9AC605A218' ]), 122([ ])), [ _ h'': 40000000 ], [ _ h'F'B3D635C7CB573D1B9E9BF4A6A4B4F25190D2906FDB0C9AC605A218' : 1 ] ] ), 123([ _ 121([ _ h'000E1400136752529784C15E63B0A2A27F81C00C8B9227913A0C40A80D4' : 1 ] ), 123([ _ 121([ _ h'8784FE49E41E924911BBA3C53FF4782EFCB0F244F875C9F9BA32' ]), 122([ ])), [ _ h'': 1000000000 ], [ _ h'8784FE49E41E924911BBA3C53FF4782EFCB0F244F875C9F9BA32' : [ _ h'01A02B8049FDC47F830E259459582CB0054587C6AC22504E984922830968A' : 1 ] ], 123([ _ 121([ _ h'000E1400136752529784C15E63B0A2A27F81C00C8B9227913A0C40A80D4' ], 121([ _ h'': h''), 100000000, 259200000, 100000000, 1, 500000, 121([ _ h'': h''), 1000000, 2000000)), 122([ ])))]
| Transaction Outputs:
| [ _ 121([ _ 121([ _ 122([ _ h'F'B3D635C7CB573D1B9E9BF4A6A4B4F25190D2906FDB0C9AC605A218' ]), 122([ ])), [ _ h'': 600000000 } ], 123([ _ 121([ ])), 122([ ])), 121([ _ 121([ _ h'8784FE49E41E924911BBA3C53FF4782EFCB0F244F875C9F9BA32' ]), 122([ ])), [ _ h'': h'8784FE49E41E924911BBA3C53FF4782EFCB0F244F875C9F9BA32' : [ _ h'01A02B8049FDC47F830E259459582CB0054587C6AC22504E984922830968A' : 1 ] ], 123([ _ 121([ _ h'000E1400136752529784C15E63B0A2A27F81C00C8B9227913A0C40A80D4' ], 121([ _ h'': h''), 100000000)), 122([ ])))]
| Minted Tokens:
| [ _ h'': [ _ h'': 0 ], [ _ h'8784FE49E41E924911BBA3C53FF4782EFCB0F244F875C9F9BA32' : [ _ h'01A02B8049FDC47F830E259459582CB0054587C6AC22504E984922830968A' : 1 ] ] )
| Step 5: Execution Result
| -----
| Unsubscription Successfully Processed!
| -----
| Test Completed!
| token name
| token quantity
| 1

```

Figure 5: Succeed Unsubscribe Test

This test verifies the contract's ability to process an unsubscription. It demonstrates:

- Accurate calculation of refund and penalty amounts
- Proper distribution of funds (refund to subscriber, penalty to designated UTxO)
- Correct burning of the Payment NFT

3.5. Test Case: Withdrawing Subscription Fees by Merchant (succeed_merchant_withdraw)

```

Testing ...
payment_subscription/tests/payment_multi_validator
PASS [mem: 49/1365, cpu: 25589/1473] succeed_merchant_withdraw
- with traces
  test: Withdrawing Subscription Fees
  -----
  Step 1: Current Contract State
  -----
  Service Currency Symbol:
  h'BA726C3C1A9165B108E6FF590CB1A1C4F0DC2E9F26A9A16F48B8E'
  Payment Currency Symbol:
  h'873E4F9E41E9249118BA3EC53FF4782EFC80F244F875C879F8A432'
  Subscription Start:
  1000000
  Subscription End:
  2593000000
  Total Subscription Fee: (lovelace)
  1000000000
  Last Claimed:
  1000000
  Current Time:
  5185000000
  Step 2: Withdrawal Calculation
  -----
  Time Elapsed: (days)
  60
  Actual Withdrawal: (lovelace)
  200000000
  Step 3: Verifying Outputs
  -----
  Merchant Output:
  121([ 121([ 121([ h'BA726C3C1A9165B108E6FF590CB1A1C4F0DC2E9F26A9A16F48B8E' ]), 122([ ])), [ h'': 200000000 ], h'BA726C3C1A9165B108E6FF590CB1A1C4F0DC2E9F26A9A16F48B8E': [ h'000DE1400136752529784C15E638DA2A27F81C00C9C8B92277913A8DC40A86D4': 1 ]
), 122([ 121([ ]), 122([ ]))
  Remaining Payment Output:
  121([ 121([ 121([ h'873E4F9E41E9249118BA3EC53FF4782EFC80F244F875C879F8A432' ]), 122([ ])), [ h'': 900000000 ], h'873E4F9E41E9249118BA3EC53FF4782EFC80F244F875C879F8A432': [ h'01A8028049AFDC47FB302E59459582CB6D0545B7C6AC22504E984922830668A': 1 ]
), 122([ 121([ h'000DE1400136752529784C15E638DA2A27F81C00C9C8B92277913A8DC40A86D4', h'000DE14001547F8F1A300F82B1B3E9858E3AB7772578567898150708920A19D', 121([ h'', h'']), 800000000, 1000000, 2593000000, 2593000000, 1000000000, 1, 5185000000, 121([ h'', h'']), 1000
000, 200000000 ]), 122([ ]))
  Step 4: Updating Payment Datum
  -----
  Original Last Claimed:
  1000000
  Updated Last Claimed:
  5185000000
  Step 5: Execution Result
  -----
  Withdrawal Successfully Processed!
  -----
  Test Completed!
  -----
tests | 1 passed | 0 failed
Summary 1 check, 0 errors, 0 warnings

```

Figure 6: Succeed Unsubscribe Test

This test confirms the contract's ability to process withdrawals of subscription fees by a merchant. It shows:

- Correct calculation of withdrawable amounts based on elapsed time
- Proper distribution of funds to the merchant
- Accurate updating of the Payment Datum with new 'last claimed' time

3.6. Test Case: Withdrawing Subscription Fees by Subscriber (succeed_subscriber_withdraw)

```

Testing ...
  payment_subscription/tests/payment_multi_validator
PASS [mem: 12258809, cpu: 6372695650] succeed subscriber_withdraw
  • with traces
  | Test: Withdrawal from Inactive Service
  | -----
  | Step 1: Current Contract State
  | -----
  | Service Active Status:
  | 121({})
  | Payment Amount:
  | 100000000
  | Step 2: Withdrawal Process
  | -----
  | Refund Amount:
  | 100000000
  | Step 3: Verifying Outputs
  | -----
  | User Output:
  | [ 121([ 121([ _ h'F83D635C7CB573D1B9E9BFF4A64AB4F25190D29B6FD80B94C605A218' ]), 122({})), { _ h'': 1100000000 }, h'F83D635C7CB573D1B9E9BFF4A64AB4F25190D29B6FD80B94C605A218': { _ h'000E140015F47F8F1A3D0F
F82B1B3E985E3AB77725785678981507D8920A190': 1 } }, 123([ 121({})), 122({})))
  | Payment Output:
  | 121([ 121([ _ h'873E4FE9E41E924911BBA3EC53FF4782EFC8CF0244F7B5C879F8AA32' ]), 122({})), { _ h'': 1000000000 }, h'873E4FE9E41E924911BBA3EC53FF4782EFC8CF0244F7B5C879F8AA32': { _ h'01A4028049AFC47F830ED
59459582CB6D054587C6A22504E984922830968A': 1 } }, 123([ 121([ _ h'000E1400136752529784C15E6380A2A27F81C00C9CB892277913A8DC40A86D4', h'000E140015F47F8F1A3D0FB2B1B3E985E3AB77725785678981507D8920A190', 121([ _ h'', h'')],
100000000, 1000000, 2592000000, 2592000000, 10000000, 1, 1000000, 121([ _ h'', h'')], 1000000, 2000000))), 122({})))
  | Step 4: Validating Transaction
  | -----
  | Transaction Inputs:
  | [ 121([ 121([ 121([ _ h'EE15SAC9EC40292074CB6AFF8CC00D273CB168F1149E736CEA6E' ]), 11), 121([ 121([ 121([ _ h'F83D635C7CB573D1B9E9BFF4A64AB4F25190D29B6FD80B94C605A218' ]), 122({})), { _ h'': 40000000 },
h'F83D635C7CB573D1B9E9BFF4A64AB4F25190D29B6FD80B94C605A218': { _ h'000E140015F47F8F1A3D0FB2B1B3E985E3AB77725785678981507D8920A190': 1 } }, 123([ 121([ 121([ _ h'000E140015F47F8F1A3D0FB2B1B3E985E3AB77725785678981507D8920A190': 1 } }, 123([ 121([ 121([ _ h'873E4FE9E41E924911BBA3EC53FF4782EFC8CF0244F7B5C879F8AA32' ]), 122({})), { _ h'': 10040000000 }, h'873E4FE9E41E924911BBA3EC53FF4782EFC8CF0244F7B5C879F8AA32': { _ h'01A4028049AFC47F830ED59459582CB6D054587C6A22504E984922830968A': 1 } }, 123([ 121([ _ h'000E1400136752529784C15E6380A2A27F81C00C9CB892277913A8DC40A86D4', h'000E140015F47F8F1A3D0FB2B1B3E985E3AB77725785678981
507D8920A190', 121([ _ h'', h'')], 1000000000, 1000000, 2593000000, 2592000000, 100000000, 1, 1000000, 121([ _ h'', h'')], 1000000, 2000000))), 122({})))]]
  | Transaction Outputs:
  | [ 121([ 121([ 121([ _ h'F83D635C7CB573D1B9E9BFF4A64AB4F25190D29B6FD80B94C605A218' ]), 122({})), { _ h'': 1100000000 }, h'F83D635C7CB573D1B9E9BFF4A64AB4F25190D29B6FD80B94C605A218': { _ h'000E140015F47F8F1A
3D0FB2B1B3E985E3AB77725785678981507D8920A190': 1 } }, 123([ 121([ 121([ _ h'873E4FE9E41E924911BBA3EC53FF4782EFC8CF0244F7B5C879F8AA32' ]), 122({})), { _ h'': 1000000000 }, h'873E4FE9E41E924911BBA3EC53FF4782EFC8CF0244F7B5C879F8AA32': { _ h'01A4028049AFC47F830ED59459582CB6D054587C6A22504E984922830968A': 1 } }, 123([ 121([ _ h'000E1400136752529784C15E6380A2A27F81C00C9CB892277913A8DC40A86D4', h'000E1400
15F47F8F1A3D0FB2B1B3E985E3AB77725785678981507D8920A190', 121([ _ h'', h'')], 1000000000, 1000000, 2593000000, 2592000000, 100000000, 1, 1000000, 121([ _ h'', h'')], 1000000, 2000000))), 122({})))]]
  | Tokens:
  | { _ h'': { _ h'': 0 }, h'873E4FE9E41E924911BBA3EC53FF4782EFC8CF0244F7B5C879F8AA32': { _ h'01A4028049AFC47F830ED59459582CB6D054587C6A22504E984922830968A': -1 } }
  | Step 5: Execution Result
  | -----
  | Withdrawal from Inactive Service Successfully Processed!
  | -----
  | Test Completed!

```

Figure 7: Succeed Unsubscribe Test

This test verifies the contract's ability to process withdrawals of subscription fees by a subscriber when the service becomes inactive. It demonstrates:

- Correct identification of an inactive service
- Full refund of the subscription amount to the subscriber
- Proper burning of the Payment NFT
- Accurate updating of the Payment UTxO

4. User Workflow for Managing Recurring Payments

The following outlines the user workflow for managing recurring payments:

1. **Initiate Subscription:**

- User selects a service and subscription period
- Smart contract mints a Payment NFT and locks the subscription fee
- User receives confirmation of successful subscription

2. **Extend Subscription:**

- User chooses to extend their subscription
- Smart contract calculates additional fee and new end date
- User approves the extension
- Contract updates the Payment Datum with new details

3. **Unsubscribe:**

- User requests to end their subscription
- Contract calculates refund and penalty amounts
- User receives refund, minus any applicable penalties
- Payment NFT is burned, ending the subscription

4. **Merchant Withdrawal**

- Merchant can withdraw accrued fees at any time
- Contract calculates withdrawable amount based on elapsed time
- Remaining funds stay locked until the next withdrawal or end of subscription

5. **Subscriber Withdrawal**

- Subscriber can withdraw remaining funds if the service becomes inactive
- Contract verifies the inactive status of the service
- Full remaining subscription amount is refunded to the subscriber
- Payment NFT is burned, finalizing the withdrawal

This workflow demonstrates the ease with which users can manage their recurring payments, from initiation to termination, directly from their wallets.

5. Conclusion

The Payment Subscription Smart Contract demonstrates robust functionality and ease of use. Through comprehensive testing and thoughtful implementation, it effectively manages recurring payments, allowing users to initiate, extend, and terminate subscriptions directly from their preferred wallet applications.

These features collectively ensure that the contract meets the needs of both service providers and subscribers, offering a secure and user-friendly solution for managing subscription-based services on the Cardano blockchain.