A black text on a white background

Description automatically generated

**SCHOOL OF ENGINEERING AND TECHNOLOGY ASSIGNMENT FOR THE**

**BACHELOR OF SOFTWARE ENGINEERING (HONS) ACADEMIC SESSION SEPTEMBER 2024**

**SWE3043 SOFTWARE TESTING**

**DEADLINE: Week 12, 13 December 2024 (Friday), 5pm via eLearn (by Group Leader) STUDENT NAME: Syehrran A/L Arulsamy (Leader) STUDENT ID: 22000608**

**STUDENT NAME: Tung Qi Yong STUDENT ID: 22022750**

**STUDENT NAME: Toh Kar Ming STUDENT ID: 21091137**

**STUDENT NAME: James Hwang Qi Leong STUDENT ID: 22004097**

**INSTRUCTIONS TO CANDIDATES**

* This is a group assignment, with FOUR (4) to FIVE (5) members in a group. This assignment will contribute 30% to your final grade. The marking scheme rubric is available on the group assignment descriptions.

|  |
| --- |
| **IMPORTANT**  The University requires students to adhere to submission deadlines for any form of assessment. Penalties are applied in relation to unauthorized late submission of work. |
| * Coursework submitted after the deadline but within 1 week will be accepted for a maximum mark of 40%. * Work handed in following the extension of 1 week after the original deadline will be regarded as a non-submission and marked zero. |
| **Lecturer’s Remark** (Use additional sheet if required)  We **Syehrran A/L Arulsamy (22000608), Tung Qi Yong (22022750), Toh Kar Ming (21091137), James Hwang Qi Leong (22004097)** received the assignment and read the comments **Syehrran(22/12/2024), Qi Yong (22/12/2024), Kar Ming(22/12/2024), James(22/12/2024)** (Signature/date) |

**Academic Honesty Acknowledgement**

“We **Syehrran A/L Arulsamy (22000608), Tung Qi Yong (22022750), Toh Kar Ming (21091137), James Hwang Qi Leong (22004097)** (students’ name) verify that this paper contains entirely our own work. We have not consulted with any outside person or materials other than what was specified (an interviewee, for example) in the assignment or the syllabus requirements. Further, we have not copied or inadvertently copied ideas, sentences, or paragraphs from another student. We realize the penalties *(refer to page 16, 5.5, Appendix 2, page 44 of the student handbook diploma and undergraduate programme)* for any kind of copying or collaboration on any assignment.”

… (Students’ signature / Date)

**Part 1: Test Design**

**Specifications**

**Document Control**

|  |  |
| --- | --- |
| **Document Name** | ATMS Test Design Specification (Iteration 1) |
| **Reference Number** | ATMS\_TDS\_1 |
| **Version** | 1.0.0 |
| **Project Code** | XYZ\_ATMS |
| **Status** | In-use |
| **Date Released** | 22nd of December 2024 |

|  |  |  |
| --- | --- | --- |
| **Name** | **Position** | **Signature** |
| Prepared By*:*  Syehrran A/L Arulsamy | Test Manager  ABC Test-Co | SYEHRRAN |
| Prepared By*:*  Tung Qi Yong | Test Lead  ABC Test-Co | QI YONG |
| Prepared By:  Toh Kar Ming | Test Analyst  ABC Test-Co | KAR MING |
| Prepared By:  James Hwang Qi Leong | Test Analyst  ABC Test-Co | JAMES |

**Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Release Date** | **Section** | **Amendments** |
| 1.0.0 | 22/12/2024 | All | Original document |

**Distribution List**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Version** | **Release Date** | **Controlled Copy No** | **Recipient Name** | **Department** | **Issue Date** | **Return Date** |
| 1.0.0 | 22/12/2024 | 01 |  | ABC Test-Co  QA Dept | 22/12/2024 |  |
| 1.0.0 | 22/12/2024 | 02 |  | ABC Test-Co  Test Team | 22/12/2024 |  |
| 1.0.0 | 22/12/2024 | 03 |  | XYZ Bank | 22/12/2024 |  |

**TABLE OF CONTENT**

[**Document Control 4**](#_Toc356732216)

[**Version History 5**](#_Toc356732217)

[**Distribution List 7**](#_Toc356732218)

[**1.0 Introduction** 7](#_Toc356732220)

[1.1 Purpose 7](#_Toc356732221)

[1.2 Scope 7](#_Toc356732222)

[1.3 References 7](#_Toc356732223)

[**2.0 Test Design 8**](#_Toc356732224)

[2.1 Features To Be Tested 8](#_Toc356732225)

[2.2 Approach Refinements 8](#_Toc356732226)

2.2.1 F005 Withdrawal…………………………………………………………………………..9

2.2.2 F007 Transfer…………………………………………………………………………….18 2.2.2 F008 Balance Inquiry……………………………………………………………............28

[**2.3** **Test Identification**](#_Toc356732227) 33

2.3.1 F005 Withdrawal………………………………………………………………………….33

2.3.2 F007 Transfer…………………………………………………………………………….35 2.3.2 F008 Balance Inquiry……………………………………………………………............36

[**2.4 Feature Pass/Fail Criteria**](#_Toc356732228) **37**

[**2.5 Test Deliverable 3**](#_Toc356732229)**7**

**1.0 INTRODUCTION**

**1.1 Purpose**

This test design specification supports the following objectives:

1. To identify the test items covered
2. To further refine the test approach from Test Plan
3. To define the sources of the information used to prepare the plan
4. To identify the general cases of test to be executed

**1.2 Scope**

This test design specification covers Withdrawal, Transfer, Balance Inquiry features developed in ATMS.

**1.3 References**

The following IEEE standards have been referenced in preparation of this document:

1. IEEE 829-2008 Standard for Software and System Test Documentation

The following documents provide the test basis for the test design:

I. ATMS Software Requirement Specifications (ATMS\_SRS\_1.0)

1. ATMS Software Design Specifications (ATMS\_SDS\_1.0)

**2.0 Test Design**

**2.1 Features to Be Tested**

The following table contains the features to be tested based on ATMS Software Requirement Specification (SRS) for the iteration. Listed together are the function ID and its corresponding functions and estimated risk level.

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature ID** | **Description** | **Risk Level** | **Test level** |
| F005 | Withdrawal Transaction | High | Functional Testing |
| F007 | Transfer | High | Functional  Testing |
| F008 | Balance Inquiry | High | Functional  Testing |

*Table 2.1 Features to Be Tested*

**2.2 Approach Refinement**

Test cases for ATMS will be designed using five techniques which are covered in this section. Techniques are applied based on suitability of techniques according to nature of features.

The techniques that will be applied are as follows:

1. Equivalence Partitioning
2. Boundary Value Analysis
3. Decision Table Testing
4. State Transition Testing
5. Use Case Testing

Nature of features as below:

* F005 Withdrawal – The cash dispenser releases the requested amount after a successful withdrawal transaction.
* F007 Transfer – Successfully transfers the specified amount from the source account to the destination account.
* F008 Balance Inquiry – Successfully displays the account balance on the screen.

**2.2.1 F005 Withdrawal**

Three techniques are identified for designing test for this feature, including:

i. Decision Table Testing

ii. State Transition Testing

iii. Use Case Testing

The decision table will validate combinations of input conditions and their corresponding outcomes for the withdrawal process. State transition testing will assess the ATM system's behavior as users navigate through the various states involved in completing a withdrawal. Lastly, use case testing will focus on verifying the core functionality of the withdrawal feature to ensure it performs as expected for the user.

**2.2.1.1 Decision Table Testing**

|  |  |
| --- | --- |
| **Test Condition ID** | **Test Condition** |
| TCON - 05 - 001 | Valid transaction type |
| TCON - 05 - 002 | Valid account type |
| TCON - 05 - 003 | Valid withdrawal amount |
| TCON - 05 - 004 | Sufficient ATM cash |
| TCON - 05 - 005 | Sufficient account funds |
| TCON - 05 - 006 | Daily withdrawal limit not exceeded |
| TCON - 05 - 007 | Dispense Money |
| TCON - 05 - 008 | Print Receipt |
| TCON - 05 - 009 | Eject Card |
| TCON - 05 - 010 | Display error message |

*Table 2.2.1.1a Withdrawal Decision Table Test Conditions*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Condition** | TCOV - 05 - 001 | TCOV - 05 - 002 | TCOV - 05 - 003 | TCOV - 05 - 004 | TCOV - 05 - 005 | TCOV - 05 - 006 | TCOV - 05 - 007 | TCOV  - 05 - 008 |
| Valid transaction type | N | Y | Y | Y | Y | Y | Y | Y |
| Valid account type | - | N | Y | Y | Y | Y | Y | Y |
| Valid withdrawal amount | - | - | N | Y | Y | Y | Y | Y |
| Sufficient ATM Funds | - | - | - | N | Y | Y | Y | Y |
| Sufficient account funds | - | - | - | - | N | Y | Y | Y |
| Daily withdrawal limit not exceeded | - | - | - | - | - | N | Y | Y |
| **Output** |  | | | | | | | |
| Dispense Money | N | N | N | N | N | N | Y | Y |
| Print Receipt | - | - | - | - | - | - | Y | N |
| Eject Card | N | N | N | N | N | N | Y | Y |
| Display error message | Y | Y | Y | Y | Y | Y | N | N |

*Table 2.2.1.1b Balance Inquiry Decision Table*

**2.2.1.2 State Transition Testing**

A diagram of a company

Description automatically generated

*Figure 2.2.1.2 Withdrawal State Transition* *Diagram*

The test condition for state transition testing is as follows:

i. TCON - 05 - 011 Withdrawal State Transition

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Input**  **State** | **Transaction Type (TT)** | **Invalid Transaction Type (ITT)** | **Account Type (AT)** | **Invalid Account Type (IAT)** | **Select Amount (SA)** | **Invalid Select Amount (ISA)** | **Balance OK (BO)** | **Insufficient Balance (IB)** | **Continue Transition (CoT)** | **Not Continue (NC)** | **Cancel Transaction (CT)** |
| **Choosing Transaction (S1)** | S2/ PT TCO V-  05-  009 | S1/ RPT TCO V-  05-  010 | S1/- | S1/- | S1/- | S1/- | S1/- | S1/- | S1/- | S1/  - | S1/  - |
| **Choosing Account (S2)** | S2/- | S2/- | S3/ PA TCO V-  05-  011 | S2/ RPA TCO V-  05-  012 | S2/- | S2/- | S2/- | S2/- | S2/- | S2/  - | S1/ PS TC OV  -  05-  01  3 |
| **Getting Amount (S3)** | S3/- | S3/- | S3/- | S3/- | S4/ PAm TCO V-  05-  014 | S3/ RPA  m TCO V-  05-  015 | S3/- | S3/- | S3/- | S3/  - | S1/ PS TC OV  -  05-  01  6 |
| **Checking Balance (S4)** | S4/- | S4/- | S4/- | S4/- | S4/- | S4/- | S5/ PDR TCO V-  05-  017 | S6/ RPA  m TCO V-  05-  018 | S4/- | S4/  - | S1/ PS TC OV  -  05-  01  9 |
| **Print Receipt (S5)** | S5/- | S5/- | S5/- | S5/- | S5/- | S5/- | S5/- | S5/- | S1/ PS TCO V-  05-  020 | S7/ EC TC OV  -  05-  021 | S7/  - |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Continuing Session (S6)** | S6/- | S6/- | S6/- | S6/- | S6/- | S6/- | S6/- | S6/- | S1/ PS TCO V-  05-  022 | S7/ EC TC OV  -  05-  023 | S6/  - |
| **Ejecting**  **Card (S7)** | - | - | - | - | - | - | - | - | - | - | - |

## *Table 2.2.1.3a Withdrawal State Table*

|  |  |
| --- | --- |
| **Test Coverage ID** | **Test Coverage** |
| TCOV - 05 - 009 | S1 to S2 with input TT |
| TCOV - 05 - 010 | S1 to S1 with input ITT |
| TCOV - 05 - 011 | S2 to S3 with input AT |
| TCOV - 05 - 012 | S2 to S2 with input IAT |
| TCOV - 05 - 013 | S2 to S1 with input CT |
| TCOV - 05 - 014 | S3 to S4 with input SA |
| TCOV - 05 - 015 | S3 to S3 with input ISA |
| TCOV - 05 - 016 | S3 to S1 with input CT |
| TCOV - 05 - 017 | S4 to S5 with input BO |
| TCOV - 05 - 018 | S4 to S6 with input IB |
| TCOV - 05 - 019 | S4 to S1 with input CT |
| TCOV - 05 - 020 | S5 to S1 with input CoT |
| TCOV - 05 - 021 | S5 to S7 with input NC |
| TCOV - 05 - 022 | S6 to S1 with input CoT |
| TCOV - 05 - 023 | S6 to S7 with input NC |

*Table 2.2.1.3b Withdrawal State Coverage*

**2.2.1.3 Use Case Testing**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC001 | |
| **Use Case** | F005 Withdrawal | |
| **Purpose** | To allow users to perform a cash withdrawal. | |
| **Actor** | User | |
| **Trigger** | The user chooses “Withdrawal” in the Transaction Menu | |
| **Precondition** | ATM is in ‘On’, Valid Card, Valid Pin and at Transaction Menu | |
| **Scenario Name** | **Step** | **Action** |
| **Main Flow** | 1 | The user chooses “Withdrawal” in the Transaction Menu |
| 2 | System requests for the type of account to withdraw from |
| 3 | User chooses the valid type of account to withdraw from |
| 4 | System verifies the type of account to withdraw |
| 5 | System requests for the amount to be withdrawn |
| 6 | The user enters the valid amount of banknotes in ATM |
|  | 7 | System verifies the amount of banknotes in ATM |
| 8 | System verifies the daily withdrawal limit of account |
| 9 | The system verifies amount of money in the account |
| 10 | System dispenses cash equal to the withdrawal amount |
| 11 | System records in log and prints receipt |
| 12 | The system asks whether the user wants to perform another transition |
| 13 | User chooses “No” |
| 14 | System terminates session and ejects card |
| 15 | System enters ‘Idle’ state and displays message “Please insert your card” |
| **Alternate Flow- Invalid Account Type** | 3.1.1 | User chooses invalid type of account to withdraw from |
| 3.1.2 | System displays error message “Invalid account type” |
| 3.1.3 | Back to Main Flow Step 12 |
| **Alternate Flow- Cancel Withdrawal at Account Type**  **Selection** | 3.2.1 | User presses “Cancel” button |
| 3.2.2 | Back to Main Flow Step 12 |
| **Alternate Flow- Invalid Withdrawal Amount (Insufficient**  **Balance in ATM)** | 6.1.1 | The user inputs invalid amount to withdraw (Amount more than balance in ATM) |
| 6.1.2 | The system displays error message “Insufficient cash” |
| 6.1.3 | Back to Main Flow Step 5 |
| **Alternate Flow- Invalid Withdrawal Amount (Insufficient Balance in**  **Account)** | 6.2.1 | The user inputs invalid amount to withdraw (Amount more than the balance in customer’s account) |
| 6.2.2 | The system display error message “Insufficient balance” |
| 6.2.3 | Back to Main Flow Step 5 |
| **Alternate Flow- Invalid Withdrawal Amount (Daily Withdrawal Limit Reached)** | 6.3.1 | User inputs invalid amount to withdraw (Amount more than daily withdrawal limit) |
| 6.3.2 | The system displays error message “Daily withdrawal limit reached” |
| 6.3.3 | Back to Main Flow Step 5 |
| **Alternate Flow- Cancel When Entering Withdrawal**  **Amount)** | 6.4.1 | The user presses “Cancel” button |
| 6.4.2 | Back to Main Flow Step 5 |
| **Alternate Flow- Perform another Transaction** | 13.1 | User chooses “Yes” |
| 13.2 | The system displays the transaction menu and waits for user’s transaction request |
| **Rules** | 1. Withdrawal amount is set at 20, 40, 60, 100, 200 2. Daily withdrawal limit is set at 300 by default | |

*Table 2.2.1.3a Withdrawal Use Case*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Condition ID** | **Test Condition** | **Test Coverage ID** | **Test Coverage** | **Test Data** |
| TCON-05-012 | Main Flow | TCOV-05-024 | Main Flow | Transaction Menu = 1 (Withdrawal), Account Type= 1 (Checking), Amount to withdrawal= 1 ($40), Continue Transaction= 2 (No) |
| TCON-05-013 | Alternative Flow- Invalid Account Type | TCOV-05-025 | Alternative Flow- Invalid Account Type | Invalid Account Type = 0 Invalid Account  Type = 4 |
| TCON-05-014 | Alternative Flow- Cancel Withdrawal At Account Type  Selection | TCOV-05-026 | Alternative Flow- Cancel Withdrawal at Account Type  Selection | Press “Cancel” |
| TCON-05-015 | Alternative Flow- Invalid Withdrawal Amount (Insufficient Balance in  ATM) | TCOV-05-027 | Alternative Flow- Invalid Withdrawal Amount (Insufficient Balance in  ATM) | Withdrawal Amount ($200)> Balance in ATM ($100) |
| TCON-05-016 | Alternative Flow- Invalid Withdrawal Amount (Insufficient  Balance in Account) | TCOV-05-028 | Alternative Flow- Invalid Withdrawal Amount (Insufficient  Balance in Account) | Withdrawal Amount ($200)  > Balance in customer’s account ($100) |
| TCON-05-017 | Alternative Flow- Invalid Withdrawal Amount (Daily Withdrawal  Limit Reached) | TCOV-05-029 | Alternative Flow- Invalid Withdrawal Amount (Daily Withdrawal  Limit Reached) | Withdrawal Amount ($301)  > Daily Limit ($300) |
| TCON-05-018 | Alternative Flow- Cancel When Entering  Withdrawal Amount) | TCOV-05-030 | Alternative Flow- Cancel When Entering  Withdrawal Amount) | Press “Cancel” |
| TCON-05-019 | Alternative Flow- Perform another  Transaction | TCOV-05-031 | Alternative Flow- Perform another  Transaction | Continue Transaction= 1 (Yes) |

*Table 2.2.1.3b Withdrawal Use Case Test Conditions & Coverage*

**2.2.2 F007 Transfer**

Four Techniques are included to design the test for the feature, including:

i. Equivalence Partitioning

ii. Boundary Value Analysis

iii. Decision Table Testing

iv. State Transition Testing

Equivalence Partitioning will divide input data into groups with expected similar outcomes, reducing test cases while covering typical and edge cases. Boundary Value Analysis (BVA) will test values at and around boundaries to identify edge-case issues. Decision Table Testing will ensure all input combinations and actions are systematically evaluated. State Transition Testing will validate the system’s behavior during state changes, ensuring proper handling of valid and invalid transitions.

A diagram of a flowchart

Description automatically generated**2.2.2.1 Equivalence Partitioning**

*Figure 2.2.2.1a Transfer Equivalence Partition & Boundary Value Analysis*

A screenshot of a computer

Description automatically generated

*Figure 2.2.2.1b Transfer Equivalence Partitions*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Condition ID** | **Test Condition** | **Test Coverage ID** | **Test Coverage** | **Example of Test Data** |
| TCON-07-001 | amount = 0.00 | TCOV-07-001 | amount = 0.00  (Invalid) | 0.00 |
| TCON-07-002 | amount > 0.00  amount <= account balance | TCOV-07-002 | amount > 0.00  amount <= account balance  (Valid) | 0.01 |
| TCON-07-003 | amount > account balance | TCOV-07-003 | amount > account balance  (Invalid) | account balance + 0.01 |
| TCON-07-004 | Valid Amount | TCOV-07-004 | Valid Amount  (Valid output) | 10.00 |
| TCON-07-005 | Invalid Amount | TCOV-07-005 | Invalid Amount (Invalid Amount) | 2147483648.00 |

*Table 2.2.2.1b Transfer Equivalence Partitioning Test Conditions & Coverage*

**2.2.2.2 Boundary Value Analysis**

**A screenshot of a phone

Description automatically generated**

*2.2.2.2a Transfer Boundary Value Analysis*

EP1: amount = 0.00

EP2: 0.00 < amount <= account balance

EP3: amount > account balance

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Condition ID** | **Test Condition** | **Test Coverage ID** | **Test Coverage** | **Example of Test Data** |
| TCON-07-001 | amount = 0.00 | TCOV-07-001 | amount = 0.00 | 0.00 |
| TCON-07-006 | amount = 0.01 | TCOV-07-006 | amount = 0.01 | 0.01 |
| TCON-07-007 | amount = account balance | TCOV-07-007 | amount = account balance  (Valid) | account balance |
| TCON-07-008 | amount = account balance + 0.01 | TCOV-07-008 | amount = account balance + 0.01  (Invalid) | account balance + 0.01 |
| TCON-07-009 | amount = 2147483648.00 | TCOV-07-009 | amount = 2147483648.00  (Invalid) | 2147483648.00 |
| TCON-07-004 | Valid Amount | TCOV-07-004 | Valid Amount  (Valid output) | 10.00 |
| TCON-07-005 | Invalid Amount | TCOV-07-005 | Invalid Amount (Invalid Amount) | 2147483648.00 |

*Table 2.2.2.2b Transfer Boundary Value Analysis Test Conditions & Coverage*

**2.2.2.3 Decision Table**

|  |  |
| --- | --- |
| **Test Condition ID** | **Test Condition** |
| TCON-07-010 | Valid Transaction Type |
| TCON-07-011 | Valid Sending Account |
| TCON-07-012 | Valid Receiving Account |
| TCON-07-013 | Display Message |
| TCON-07-014 | Retry |
| TCON-07-015 | Complete Transaction |
| TCON-07-016 | Continue Transaction |
| TCON-07-017 | Eject Card |
| TCON-07-018 | Print Receipt |

*Table 2.2.2.3a Transfer Decision Table Test Conditions*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condition/Output** | **TCOV-07-010** | **TCOV-07-011** | **TCOV-07-012** | **TCOV-07-013** |
| **Valid Transaction Type** | N | Y | Y | Y |
| **Valid Sending Account** | - | N | Y | N |
| **Valid Receiving Account** | - | N | N | Y |
| **Output** |  |  |  |  |
| **Display Message** | Y | Y | Y | Y |
| **Retry** | Y | Y | N | N |
| **Complete Transaction** | N | N | N | Y |
| **Continue Transaction** | Y | Y | Y | N |
| **Print Receipt** | N | N | N | N |
| **Eject Card** | Y | Y | Y | Y |

*Table 2.2.3.3b Transfer Decision Table*

**2.2.2.4 State Transition Testing**

A diagram of a bank

Description automatically generated

*Figure 2.2.2.4 Transfer State Transaction Diagram*

The test condition for state transition testing is as follows:

i. TCON-07-019 Transfer State Transition

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **State**  **Input** | **Transaction Type (TT)** | **Sending Account (SA)** | **Receiving Account (RA)** | **Display Transfer Amount (DTA)** | **Perform Transaction (PT)** | **Invalid Transfer: Same Amount (ITSA)** | **Invalid Transfer: Insufficient available balance (ITIAB)** | **Completed Transaction (CT)** | **Display Receipt (DR)** | **Yes (Y)** | **No (N)** | **Retry (RT)** | **Cancel (C)** | **Invalid Card (IC)** | **Invalid PIN (IP)** |
| Choosing Transaction Type (S1) | S2/DSAO  TCOV-07-014 | S1/- | S1/- | S1/- | S1/- | S1/- | S1/- | S1/- | S1/- | S1/- | S1/- | S1/RTCTT  TCOV-07-015 | S9/EC  TCOV-07-016 | S1/- | S1/- |
| Select Sending Account (S2) | S2/- | S3/DRAO  TCOV-07-017 | S2/- | S2/- | S2/- | S2/- | S2/- | S2/- | S2/- | S2/- | S8/EC  TCOV-07-018 | S2/RTSSA  TCOV-07-019 | S2/- | S2/- | S2/- |
| Select Receiving Account (S3) | S3/- | S3/- | S4/ETA  TCOV-07-020 | S3/- | S3/- | S3/- | S3/- | S3/- | S3/- | S3/- | S8/EC  TCOV-07-021 | S3/RTSRA  TCOV-07-022 | S3/- | S3/- | S3/- |
| Enter Transfer Amount (S4) | S4/- | S4/- | S4/- | S5/PT  TCOV-07-023 | S4/- | S4/- | S4/- | S4/- | S4/- | S4/- | S4/- | S4/- | S8/PAT  TCOV-07-024 | S4/- | S4/- |
| Performing Transaction (S5) | S5/- | S5/- | S5/- | S5/- | S5/- | S8/PAT  TCOV-07-025 | S8/PAT  TCOV-07-026 | S6/CT  TCOV-07-027 | S5/- | S5/- | S5/- | S5/- | S5/- | S8/PAT  TCOV-07-028 | S5/PT  TCOV-07-029 |
| Completed Transaction (S6) | S6/- | S6/- | S6/- | S6/- | S6/- | S6/- | S6/- | S7/PR  TCOV-07-030 | S6/- | S6/- | S6/- | S6/- | S6/- | S6/- | S6/- |
| Print Receipt (S7) | S7/- | S7/- | S7/- | S7/- | S7/- | S7/- | S7/- | S7/- | S8/PAT  TCOV-07-031 | S7/- | S7/- | S7/- | S9/EC  TCOV-07-032 | S7/- | S7/- |
| Perform Another Transaction (S8) | S8/- | S8/- | S8/- | S8/- | S8/- | S8/- | S8/- | S8/- | S8/- | S1/RTCTT  TCOV-07-033 | S9/EC  TCOV-07-034 | S8/- | S8/- | S8/- | S8/- |

*Table 2.2.2.4a Transfer State Table*

|  |  |
| --- | --- |
| **Test Coverage ID** | **Test Coverage** |
| TCOV-07-014 | S1 to S2 with TT |
| TCOV-07-015 | S1 to S1 with RT |
| TCOV-07-016 | S1 to S9 with C |
| TCOV-07-017 | S2 to S3 with SA |
| TCOV-07-018 | S2 to S8 with N |
| TCOV-07-019 | S2 to S2 with RT |
| TCOV-07-020 | S3 to S4 with RA |
| TCOV-07-021 | S3 to S8 with N |
| TCOV-07-022 | S3 to S3 with RT |
| TCOV-07-023 | S4 to S5 with DTA |
| TCOV-07-024 | S4 to S8 with C |
| TCOV-07-025 | S5 to S8 with ITSA |
| TCOV-07-026 | S5 to S8 with ITIAB |
| TCOV-07-027 | S5 to S6 with PT |
| TCOV-07-028 | S5 to S8 with IC |
| TCOV-07-029 | S5 to S5 with IP |
| TCOV-07-030 | S6 to S7 with CT |
| TCOV-07-031 | S7 to S8 with DR |
| TCOV-07-032 | S7 to S9 with C |
| TCOV-07-033 | S8 to S1 with Y |
| TCOV-07-034 | S8 to S9 with C |

*Table 2.2.2.4b Transfer State Coverage*

**2.2.3 F008 Balance Inquiry**

Three techniques are identified for designing test for this feature, including:

1. Decision Table Testing
2. State Transition Testing
3. Use Case Testing

The decision table will be used to test combinations of inputs or conditions and their corresponding outcomes. State transition testing will be used to test the ATMS’s behavior when users are transitioning between states. Use case testing lastly, will be testing the core functionality of the balance enquiry function to ensure that the function works as intended for the user.

**2.2.3.1 Decision Table Testing**

|  |  |
| --- | --- |
| **Test Condition ID** | **Test Condition** |
| TCON-08-001 | Choosing Balance Inquiry Transaction |
| TCON-08-002 | Selecting Account Type |
| TCON-08-003 | Display Balance |
| TCON-08-004 | Error Message for Non-existing Invalid Account |
| TCON-08-005 | Error Message for Out-of-Range Invalid Account |

*Table 2.2.3.1a Balance Inquiry Decision Table Conditions*

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **TCOV-08-001** | **TCOV-08-002** | **TCOV-08-003** |
| Choosing Balance Inquiry Transaction | Y | Y | Y |
| Selecting Account to Enquire | Y | Y | Y |
| **Output** |  | | |
| Display Balance | Y | - | - |
| Error Message for Non-existing Invalid Account | - | Y | - |
| Error Message for Out-of-Range Invalid Account | - | - | Y |

*Table 2.2.3.1b Balance Inquiry Decision Table*

**2.2.3.2 State Transition Testing**

*A screenshot of a black screen

Description automatically generated*

*Figure 2.2.3.2 Balance Inquiry State Transition Diagram*

The test condition for state transition testing is as follows:

1. TCON-08-006 Balance Inquiry State Transition

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Input**  **State** | **Inquiry (I)** | **Cancel (C)** | **Account Exists (AE)** | **Account Doesn’t Exist (AED)** | **Input Outside Option (IOO)** | **Complete Transaction (CT)** | **Yes (Y)** | **No (N)** |
| Choosing Transaction Type (S1) | S2/BI TCOV-08-004 | S5/EC TCOV-08-005 | S1/- | S1/- | S1/- | S1/- | S1/- | S1/- |
| Selecting Amount Type (S2) | S2/- | S2/- | S3/DB TCOV-08-006 | S4/DMNA TCOV-08-007 | S4/DMOR TCOV-08-008 | S2/- | S2/- | S2/- |
| Print Receipt (S3) | S3/- | S3/- | S3/- | S3/- | S3/- | S4/PT TCOV-08-009 | S3/- | S3/- |
| Performing Another Transaction (S4) | S4/- | S4/- | S4/- | S4/- | S4/- | S4/- | S5/RTM TCOV-08-010 | S5/EC TCOV-08-011 |

*Table 2.2.3.2a Balance Inquiry State Table*

|  |  |
| --- | --- |
| **Test Coverage ID** | **Test Coverage** |
| TCOV-08-004 | S1 to S2 with input I |
| TCOV-08-005 | S1 to S5 with input C |
| TCOV-08-006 | S2 to S3 with input AE |
| TCOV-08-007 | S2 to S4 with input AED |
| TCOV-08-008 | S2 to S4 with input IOO |
| TCOV-08-009 | S3 to S4 with input CT |
| TCOV-08-010 | S4 to S5 with input Y |
| TCOV-08-011 | S4 to S5 with input N |

*Table 2.2.3.2b Balance Inquiry State Coverage*

**2.2.3.3 Use Case Testing**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC008 | |
| **Use Case** | F008 Balance Inquiry | |
| **Purpose** | To allow users to perform a balance inquiry | |
| **Actor** | User | |
| **Trigger** | User chooses “Balance Inquiry” in transaction menu. | |
| **Precondition** | ATMS is in “On” state and is at transaction menu. | |
| **Scenario Name** | **Step** | **Action** |
| **Main Flow** | 1 | The system displays the transaction menu and waits for the user’s transaction request. |
| 2 | The user chooses the balance inquiry option. |
| 3 | The system displays the account type options and waits for the user’s account option. |
| 4 | The user chooses an account type. |
| 5 | The account balance chosen is displayed. |
| 6 | The system displays an option for users to print a receipt. |
| 7 | The user chooses to print the receipt. |
| 8 | The system asks if the user would like to perform another transaction. |
| 9 | The user chooses the “No” option. |
| 10 | The system terminates the session and ejects the card. |
| **Alternative Flow – Cancel Transaction Request** | 2.1 | The user chooses the “Cancel” option instead of any transaction options from the transaction menu. |
| 2.2 | The system terminates the session and ejects the card. |
| **Alternative Flow – Account Does Not Exist** | 5.1.1 | The system displays an error message for Non-existing Invalid Account. |
| 5.1.2 | Back to main flow step 8. |
| **Alternative Flow – Input Outside Options** | 5.2.1 | The system displays an error message for Out-of-Range Invalid Account. |
| 5.2.2 | Back to main flow step 8. |
| **Alternative Flow – Performs Another Transaction** | 9.1 | The user chooses the “Yes” option. |
| 9.2 | Back to main flow step 1. |

*Table 2.2.3.3a Balance Inquiry Use Case*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Condition ID** | **Test Condition** | **Test Coverage ID** | **Test Coverage** | **Test Data** |
| TCON-08-007 | Main Flow | TCOV-08-012 | Main Flow | Valid Card (Card 1)  Valid Account Type  (1: Checking & 2: Savings) |
| TCON-08-008 | Alternative Flow – Cancel Transaction Request | TCOV-08-013 | Alternative Flow – Cancel Transaction Request | Press “Cancel” |
| TCON-08-009 | Alternative Flow – Account Does Not Exist | TCOV-08-014 | Alternative Flow – Account Does Not Exist | Valid Card (Card 1)  Invalid Account Type  (3: Money Market) |
| TCON-08-010 | Alternative Flow – Input Outside Options | TCOV-08-015 | Alternative Flow – Input Outside Options | Valid Card  Invalid Account Choice  (Any input except 1 to 3) |
| TCON-08-011 | Alternative Flow – Performs Another Transaction | TCOV-08-016 | Alternative Flow – Performs Another Transaction | Press “Yes” |

*Table 2.2.3.3b Balance Inquiry Use Case Test Conditions & Coverage*

**2.3 Test Identification**

**2.3.1 Withdrawal**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Test Coverage ID** | **Test Condition ID** |
| TC-05-001 | Withdrawal Main Flow | TCOV-05-007 | TCON-05-001 |
| TCON-05-002 |
| TCON-05-003 |
| TCON-05-004 |
| TCON-05-005 |
| TCON-05-006 |
| TCON-05-007 |
| TCON-05-008 |
| TCON-05-009 |
| TCOV-05-009 | TCON-05-011 |
| TCOV-05-011 |
| TCOV-05-017 |
| TCOV-05-020 |
| TCOV-05-024 | TCON-05-012 |
| TC-05-002 | Withdrawal Alternative Flow- Invalid Account Type | TCOV-05-002 | TCON-05-001 |
| TCON-05-002 |
| TCOV-05-012 | TCON-05-011 |
| TCOV-05-025 | TCON-05-012 |
| TC-05-003 | Withdrawal Alternative Flow- Cancel withdrawal at Account Type Selection | TCOV-05-012 | TCON-05-011 |
| TCOV-05-026 | TCON-05-014 |
| TC-05-004 | Withdrawal Alternative Flow- Invalid Withdrawal Amount **(Insufficient Balance in ATM)** | TCOV-05-004 | TCON-05-001 |
| TCON-05-002 |
| TCON-05-003 |
| TCOV-05-027 | TCON-05-015 |
| TC-05-005 | Withdrawal Alternative Flow- Invalid Withdrawal Amount **(Insufficient Balance in Account)** | TCOV-05-005 | TCON-05-001 |
| TCON-05-002 |
| TCON-05-003 |
| TCON-05-004 |
| TCOV-05-018 | TCON-05-011 |
| TCOV-05-028 | TCON-05-016 |
| TC-05-006 | Withdrawal Alternative Flow- Invalid Withdrawal Amount **(Daily Withdrawal Limit Reached)** | TCOV-05-032 | TCON-05-001 |
| TCON-05-002 |
| TCON-05-003 |
| TCON-05-004 |
| TCON05-005 |
| TCOV-05-029 | TCON-05-017 |
| TC-05-007 | Withdrawal Alternative Flow- Cancel When Entering Withdrawal Amount | TCOV-05-016 | TCON-05-011 |
| TCOV-05-030 | TCON-05-018 |
| TC-05-008 | Withdrawal Alternative Flow- Perform another Transaction | TCOV-05-031 | TCON-05-019 |

*Table 2.6.1 Withdrawal Test Identification*

**2.3.2 Transfer**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Test Coverage ID** | **Test Condition ID** |
| TC-07-001 | Verify Transfer: Return Money to Same Account | TCOV-07-010 | TCON-07-010 |
| TCOV-07-011 | TCON-07-011 |
| TCOV-07-012 | TCON-07-012 |
| TCOV-07-013 | TCON-07-013 |
| TC-07-002 | Verify Transfer: Transfer Money to a Different Account | TCOV-07-010 | TCON-07-010 |
| TCOV-07-011 | TCON-07-011 |
| TCOV-07-012 | TCON-07-012 |
| TCOV-07-013 | TCON-07-013  03 |
| TC-07-003 | Verify Transfer Amount (amount > 0.00) | TCOV-07-002 | TCON-07-002 |
| TC-07-004 | Verify Transfer Amount (amount = 0.00) | TCOV-07-001 | TCON-07-001 |
| TC-07-005 | Verify Transfer Amount (amount > account balance) | TCOV-07-003 | TCON-07-003 |
| TC-07-006 | Verify Continue Transaction (input = 1 (Yes)) | TCOV-07-010 | TCON-07-010 |
| TCOV-07-011 | TCON-07-011 |
| TCOV-07-012 | TCON-07-012 |
| TC-07-007 | Verify Continue Transaction (input = 2 (No)) | TCOV-07-013 | TCOV-07-013 |

*Table 2.3.2 Transfer Test Identification*

**2.3.3 Balance Inquiry**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Test Coverage ID** | **Test Condition ID** |
| TC-08-001 | Balance Inquiry Main Flow | TCOV-08-001 | TCON-08-001 |
| TCON-08-002 |
| TCON-08-003 |
| TCOV-08-004 | TCON-08-006 |
| TCOV-08-006 |
| TCOV-08-009 |
| TCOV-08-011 |
| TCOV-08-012 | TCON-08-007 |
| TC-08-002 | Balance Inquiry Alternate Flow - Cancel Transaction Request | TCOV-08-005 | TCON-08-006 |
| TCOV-08-013 | TCON-08-008 |
| TC-08-003 | Balance Inquiry Alternate Flow - Account Does Not Exist | TCOV-08-002 | TCON-08-001 |
| TCON-08-002 |
| TCON-08-004 |
| TCOV-08-007 | TCON-08-006 |
| TCOV-08-014 | TCON-08-009 |
| TC-08-004 | Balance Inquiry Alternate Flow – Input Outside Options | TCOV-08-003 | TCON-08-001 |
| TCON-08-002 |
| TCON-08-005 |
| TCOV-08-008 | TCON-08-006 |
| TCOV-08-015 | TCON-08-010 |
| TC-08-005 | Balance Inquiry Alternative Flow – Performs Another Transaction | TCOV-08-010 | TCON-08-006 |
| TCOV-08-016 | TCON-08-011 |

*Table 2.3.3 Balance Inquiry Test Identification*

**2.4 Feature Pass/ Fail Criteria**

Each feature must satisfy the following criteria in order to pass:

1. Pass all its test cases
2. 0% calculation error if the features involves monetary transaction
3. No deadlock
4. Fulfil its requirement

**2.5 Test Deliverable**

The following documents will be generated by the system test group and will be delivered after test completion.

Test documentation:

1. Test Design Specifications

ii. Test Case Specifications

**Part 1:Test Case**

**Specifications**

**Document Control**

|  |  |
| --- | --- |
| **Document Name** | ATMS Test Design Specification (Iteration 1) |
| **Reference Number** | ATMS\_TDS\_1 |
| **Version** | 1.0.0 |
| **Project Code** | XYZ\_ATMS |
| **Status** | In-use |
| **Date Released** | 22nd of December 2024 |

|  |  |  |
| --- | --- | --- |
| **Name** | **Position** | **Signature** |
| Prepared By*:*  Syehrran A/L Arulsamy | Test Manager  ABC Test-Co | SYEHRRAN |
| Prepared By*:*  Tung Qi Yong | Test Lead  ABC Test-Co | QI YONG |
| Prepared By:  Toh Kar Ming | Test Analyst  ABC Test-Co | KAR MING |
| Prepared By:  James Hwang Qi Leong | Test Analyst  ABC Test-Co | JAMES |

**Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Release Date** | **Section** | **Amendments** |
| 1.0.0 | 22/12/2024 | All | Original document |

**Distribution List**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Version** | **Release Date** | **Controlled Copy No** | **Recipient Name** | **Department** | **Issue Date** | **Return Date** |
| 1.0.0 | 22/12/2024 | 01 |  | ABC Test-Co  QA Dept | 22/12/2024 |  |
| 1.0.0 | 22/12/2024 | 02 |  | ABC Test-Co  Test Team | 22/12/2024 |  |
| 1.0.0 | 22/12/2024 | 03 |  | XYZ Bank | 22/12/2024 |  |

**TABLE OF CONTENT**

[**Document Control 39**](#_Toc356732216)

[**Version History 40**](#_Toc356732217)

[**Distribution List 40**](#_Toc356732218)

[**1.0 Introduction** **42**](#_Toc356732220)

[1.1 Purpose 42](#_Toc356732221)

[1.2 Scope 42](#_Toc356732222)

[1.3 References 42](#_Toc356732223)

[**2.0 Test Case 43**](#_Toc356732224)

[2.1 Environment 43](#_Toc356732225)

[2.2 Test Cases 44](#_Toc356732226)

2.2.1 F005 Withdrawal………………………………………………………………………….44

2.2.2 F007 Transfer…………………………………………………………………………….49 2.2.2 F008 Balance Inquiry……………………………………………………………............55

**1.0 Introduction**

**1.1 Purpose**

This test case specification supports the following objective:

i. To detail the test cases derived from test coverage in Test Design Specification.

**1.2 Scope**

This test case specification covers Withdrawal, Transfer, Balance Inquiry features developed in ATMS.

**1.3 References**

The following IEEE standards have been referenced in preparation of this document:

i. IEEE 829-2008 Standard for Software and System Test Documentation

The following document provides the test basis for this test design:

i. ATMS Software Requirement Specifications, ATMS\_SRS\_1.0

ii. ATMS System Design Specifications, ATMS\_SDS\_1.0

iii. ATMS Test Design Specifications, ATMS\_TDS\_1\_1.0.0

**2.0 Test Cases**

**2.1 Environment**

ATM tests will be run using dummy accounts, where tested transactions will not involve actual monetary transactions. Tests shall be performed on an actual ATM with the following specifications:

|  |  |
| --- | --- |
| **Hardware Name** | Cineo C2040 |
| **Manufacturer** | Wincor Nixdorf |
| **Keyboard** | EPP V6 (ZKA & PCI certified) |
| **Card Processor** | Hybrid motorized, EMV x1 Level 1 certified |

*Table 2.1a ATM Hardware Requirements*

The following dummy ATM cards are needed for testing purposes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Card Number** | **Validity** | **Available Account** | **Account Balance** |
| 0 | Invalid | N/A | N/A |
| 1 | Valid | 1 | 100 |
| 2 | 1000 |
| 2 | Valid | 1 | 100 |
| 3 | 5000 |
| 3 | Valid | 1 | 100 |
| 2 | 1000 |
| 3 | 5000 |

*Table 2.1b: Required Dummy ATM Cards*

**2.2 Test Case**

This section details all test cases designed separated according to features.

**2.2.1 Withdrawal**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | | TC-05-001 | | |
| **Related Feature ID** | | F005 | | |
| **Objective** | | Withdrawal Main Flow | | |
| **Covered Test Coverage Items** | | TCOV-05-007, TCOV-05-009, TCOV-05-011, TCOV-05-017, TCOV-05-020, TCOV-05-024 | | |
| **#** | **Input** | **Expected Result** | **Special Procedural Requirements** | **Intercase Dependency** |
| 1 | Transaction Type: 1 (Withdrawal) | System proceeds to account type selection. | Card 1 with Correct PIN Input.  ATMS at Transaction Menu. | None |
| 2 | Account Type: 2 (Savings) | System displays withdrawal amount selection screen. |
| 3 | Withdrawal Amount: RM100 | System debits RM100, updates balance, prints receipt, and prompts for another transaction. |

*Table 2.2.1.1 Withdrawal Main Flow Test Case*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | | TC-05-002 | | |
| **Related Feature ID** | | F005 | | |
| **Objective** | | Withdrawal Alternative Flow - Invalid Account Type | | |
| **Covered Test Coverage Items** | | TCOV-05-002, TCOV-05-012, TCOV-05-025 | | |
| **#** | **Input** | **Expected Result** | **Special Procedural Requirements** | **Intercase Dependency** |
| 1 | Transaction Type: 1 (Withdrawal) | Displays an error message: "Invalid account type." | Card 1 with Correct PIN Input.  ATMS at Transaction Menu. | None |
| 2 | Account Type: 4 (Invalid) | Prompts user to re-select a valid account. |

*Table 2.2.1.2 Withdrawal Alternative Flow - Invalid Account Type Test Case*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | | TC-05-003 | | |
| **Related Feature ID** | | F005 | | |
| **Objective** | | Withdrawal Alternative Flow - Cancel Withdrawal at Account | | |
| **Covered Test Coverage Items** | | TCOV-05-012, TCOV-05-026, TCOV-05-037 | | |
| **#** | **Input** | **Expected Result** | **Special Procedural Requirements** | **Intercase Dependency** |
| 1 | Transaction Type: 1 (Withdrawal) | System proceeds to account type selection. | Card 1 with Correct PIN Input.  ATMS at Transaction Menu. | None |
| 2 | Account Type: None | Prompts for another transaction. |
| 3 | Cancel pressed at the account selection screen. | System terminates the transaction. |

*Table 2.2.1.3 Withdrawal Alternative Flow - Invalid Account Type Test Case*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | | TC-05-004 | | |
| **Related Feature ID** | | F005 | | |
| **Objective** | | Invalid Withdrawal Amount (Insufficient Balance in ATM) | | |
| **Covered Test Coverage Items** | | TCOV-05-004, TCOV-05-027, TCOV-05-028 | | |
| **#** | **Input** | **Expected Result** | **Special Procedural Requirements** | **Intercase Dependency** |
| 1 | Transaction Type: 1 (Withdrawal) | System proceeds to account type selection. | Card 1 with Correct PIN Input.  ATMS at Transaction Menu. | None |
| 2 | Account Type: 1 (Checking) | System displays withdrawal amount selection screen |
| 3 | Withdrawal Amount: RM200 (ATM funds = RM150) | Displays error message: "Insufficient ATM funds." |

*Table 2.2.1.4 Invalid Withdrawal Amount (Insufficient Balance in ATM)Test Case*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | | TC-05-005 | | |
| **Related Feature ID** | | F005 | | |
| **Objective** | | Invalid Withdrawal Amount (Insufficient Balance in Account) | | |
| **Covered Test Coverage Items** | | TCOV-05-005, TCOV-05-018, TCOV-05-028 | | |
| **#** | **Input** | **Expected Result** | **Special Procedural Requirements** | **Intercase Dependency** |
| 1 | Transaction Type: 1 (Withdrawal) | System proceeds to account type selection. | Card 1 with Correct PIN Input.  ATMS at Transaction Menu. | None |
| 2 | Account Type: 2 (Savings) | System displays withdrawal amount selection screen |
| 3 | Withdrawal Amount: RM200 (Account balance = RM100) | Displays error message: "Insufficient account balance." |

*Table 2.2.1.5 Invalid Withdrawal Amount (Insufficient Balance in ATM)Test Case*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | | TC-05-006 | | |
| **Related Feature ID** | | F005 | | |
| **Objective** | | Invalid Withdrawal Amount (Daily Withdrawal Limit Reached) | | |
| **Covered Test Coverage Items** | | TCOV-05-032, TCOV-05-029 | | |
| **#** | **Input** | **Expected Result** | **Special Procedural Requirements** | **Intercase Dependency** |
| 1 | Transaction Type: 1 (Withdrawal) | System proceeds to account type selection. | Card 1 with Correct PIN Input.  ATMS at Transaction Menu. | None |
| 2 | Account Type: 2 (Savings) | System displays withdrawal amount selection screen |
| 3 | Withdrawal Amount: RM300 (Daily Limit = RM200) | Displays error message: "Daily withdrawal limit reached." Prompts user to re-enter a valid amount. |

*Table 2.2.1.6 Invalid Withdrawal Amount (Daily Withdrawal Limit Reached)**Test Case*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | | TC-05-007 | | |
| **Related Feature ID** | | F005 | | |
| **Objective** | | Cancel When Entering Withdrawal Amount | | |
| **Covered Test Coverage Items** | | TCOV-05-016, TCOV-05-030 | | |
| **#** | **Input** | **Expected Result** | **Special Procedural Requirements** | **Intercase Dependency** |
| 1 | Transaction Type: 1 (Withdrawal) | Displays message: "Transaction canceled." | Card 1 with Correct PIN Input.  ATMS at Transaction Menu. | None |
| 2 | Account Type: 1 (Checking) | Prompts for another transaction. |
| 3 | Cancel pressed at the amount entry screen. |  |

*Table 2.2.1.7 Cancel When Entering Withdrawal Amount)**Test Case*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | | TC-05-008 | | |
| **Related Feature ID** | | F005 | | |
| **Objective** | | Withdrawal Alternative Flow - Perform Another Transaction | | |
| **Covered Test Coverage Items** | | TCOV-05-031 | | |
| **#** | **Input** | **Expected Result** | **Special Procedural Requirements** | **Intercase Dependency** |
| 1 | User chooses "Yes" to perform another transaction after a successful withdrawal. | Redirects to the transaction selection screen. | Card 1 with Correct PIN Input.  ATMS at Transaction Menu. | None |

*Table 2.2.1.8* ***Withdrawal Alternative Flow - Perform Another* *Transaction*** *Test Case*

**2.2.2 Transfer**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | TC-07-001 | | |
| **Related Feature ID** | F007 | | |
| **Objective** | Verify Transfer: Return Money to Same Account | | |
| **Covered Test Coverage Items** | TCOV-07-010, TCOV-07-011, TCOV-07-012, TCO7-07-013 | | |
| **Input** | **Expected Result** | **Special Procedural Requirements** | **Intercase Dependency** |
| Sending Account = 1 (Checking Account)  Receiving Account = 1 (Checking Account) | System displays error message: “Can’t transfer money from an account to itself” | Card 1. | None |

*Table 2.2.2.1 Verify Transfer: Return Money to Same Account Test Case*

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | TC-07-002 | | |
| **Related Feature ID** | F007 | | |
| **Objective** | Verify Transfer: Transfer Money to a Different Account | | |
| **Covered Test Coverage Items** | TCOV-07-010, TCOV-07-011, TCOV-07-012, TCOV-07-013 | | |
| **Input** | **Expected Result** | **Special Procedural Requirements** | **Intercase Dependency** |
| Sending Account = 1 (Checking Account)  Receiving Account = 2(Savings Account) | System prints transaction receipt and prompt message “Would you like to do another transaction (1) Yes/ (2) No” | Card 1. | None |

*Table 2.2.2.2 Verify Transfer: Transfer Money to a Different Account Test Case*

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | TC-07-003 | | |
| **Related Feature ID** | F007 | | |
| **Objective** | Verify Transfer Amount (amount > 0.00) | | |
| **Covered Test Coverage Items** | TCOV-07-002 | | |
| **Input** | **Expected Result** | **Special Procedural Requirements** | **Intercase Dependency** |
| Amount > 0.00 | System print transaction receipt and prompt message “Would you like to do another transaction (1) Yes/ (2) No” | Card 1. | None |

*Table 2.2.2.3 Verify Transfer Amount (amount > 0.00) Test Case*

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | TC-07-004 | | |
| **Related Feature ID** | F007 | | |
| **Objective** | Verify Transfer Amount (amount = 0.00) | | |
| **Covered Test Coverage Items** | TCOV-07-001 | | |
| **Input** | **Expected Result** | **Special Procedural Requirements** | **Intercase Dependency** |
| Amount = 0.00 | System prints transaction receipt and prompt message “Would you like to do another transaction (1) Yes/ (2) No” | Card 1. | None |

*Table 2.2.2.4 Verify Transfer Amount (amount = 0.00) Test Case*

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | TC-07-005 | | |
| **Related Feature ID** | F007 | | |
| **Objective** | Verify Transfer Amount (amount > account balance) | | |
| **Covered Test Coverage Items** | TCOV-07-003 | | |
| **Input** | **Expected Result** | **Special Procedural Requirements** | **Intercase Dependency** |
| Amount > account balance | System displays message “Insufficient available balance. Would you like to do another transaction (1) Yes/ (2) No” | Card 1. | None |

*Table 2.2.2.5 Verify Transfer Amount (amount > account balance) Test Case*

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | TC-07-006 | | |
| **Related Feature ID** | F007 | | |
| **Objective** | Verify Continue Transaction (input = 1 (Yes)) | | |
| **Covered Test Coverage Items** | TCOV-07-010, TCOV-07-011, TCOV-07-012 | | |
| **Input** | **Expected Result** | **Special Procedural Requirements** | **Intercase Dependency** |
| Input = 1 (Yes) | System returns to the transaction type menu and allows the user to select their desired transaction. | Card 1. | None |

*Table 2.2.2.6 Verify Continue Transaction (input = 1 (Yes)) Test Case*

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | TC-07-007 | | |
| **Related Feature ID** | F007 | | |
| **Objective** | Verify Continue Transaction (input = 2 (No)) | | |
| **Covered Test Coverage Items** | TCOV-07-013 | | |
| **Input** | **Expected Result** | **Special Procedural Requirements** | **Intercase Dependency** |
| Input = 2 (No) | System ejects the card. | Card 1. | None |

*Table 2.2.2.7 Verify Continue Transaction (input = 2 (No)) Test Case*

**2.2.3 Balance Inquiry**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | | TC-08-001 | | |
| **Related Feature ID** | | F008 | | |
| **Objective** | | Balance Inquiry Main Flow | | |
| **Covered Test Coverage Items** | | TCOV-08-001, TCOV-08-004, TCOV-08-006, TCOV-08-009, TCOV-08-011, TCOV-08-012 | | |
| **#** | **Input** | **Expected Result** | **Special Procedural Requirements** | **Intercase Dependency** |
| 1 | Choose “4: Balance Inquiry” menu option. | System displays options of accounts to inquire from. | Card 1 with Correct PIN Input.  ATMS at Transaction Menu. | None |
| 2 | Account Type: 1 | Displays account balance. | None |
| 3 | Select “Take Receipt” button. | The balance logs disappear. | None |
| 4 | Continue Transaction = 2 | Card is ejected. | None |

*Table 2.2.3.1: Balance Inquiry Main Flow Test Case*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | | TC-08-002 | | |
| **Related Feature ID** | | F008 | | |
| **Objective** | | Balance Inquiry Alternate Flow - Cancel Transaction Request | | |
| **Covered Test Coverage Items** | | TCOV-08-005, TCOV-08-013 | | |
| **#** | **Input** | **Expected Result** | **Special Procedural Requirements** | **Intercase Dependency** |
| 1 | Select “Cancel” button. | Card is ejected. | Any card with any PIN.  ATMS at Transaction Menu. | None |

*Table 2.2.3.2: Balance Inquiry Alternate Flow - Cancel Transaction Request Test Case*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | | TC-08-003 | | |
| **Related Feature ID** | | F008 | | |
| **Objective** | | Balance Inquiry Alternate Flow - Account Does Not Exist | | |
| **Covered Test Coverage Items** | | TCOV-08-002, TCOV-08-007, TCOV-08-014 | | |
| **#** | **Input** | **Expected Result** | **Special Procedural Requirements** | **Intercase Dependency** |
| 1 | Choose “4: Balance Inquiry” menu option. | System displays options of accounts to inquire from. | Card 1 with Correct PIN Input.  ATMS at Transaction Menu. | None |
| 2 | Account Type: 3 | System displays error message “Invalid Account Type”. | None |
| 3 | Continue Transaction = 2 | Card is ejected. | None |

*Table 2.2.3.3: Balance Inquiry Alternate Flow - Account Does Not Exist Test Case*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | | TC-08-004 | | |
| **Related Feature ID** | | F008 | | |
| **Objective** | | Balance Inquiry Alternate Flow – Input Outside Options | | |
| **Covered Test Coverage Items** | | TCOV-08-003, TCOV-08-008, TCOV-08-015 | | |
| **#** | **Input** | **Expected Result** | **Special Procedural Requirements** | **Intercase Dependency** |
| 1 | Choose “4: Balance Inquiry” menu option. | System displays options of accounts to inquire from. | Card 1 with Correct PIN Input.  ATMS at Transaction Menu. | None |
| 2 | Account Type: 4 | System displays error message “Invalid Input”. | None |
| 3 | Continue Transaction = 2 | Card is ejected. | None |

*Table 2.2.3.4: Balance Inquiry Alternate Flow - Input Outside Options Test Case*

**Part 2: Static Testing Techniques – Reviews**

**TABLE OF CONTENT**

[**1.0 Type of Reviews 59**](#_Toc356732216)

[**2.0 Testing Principles 60**](#_Toc356732217)

[**3.0 Why are Review Useful 61**](#_Toc356732218)

[**4.0 What Kinds of Problems Do You Think Reviews Would Locate? 62**](#_Toc356732219)

**5.0 What Kinds of Problems Might Reviews Not Locate?** [**63**](#_Toc356732220)

**6.0 How Did Our Group Implement It** [**65**](#_Toc356732224)

**7.0 Team Findings** [**66**](#_Toc356732224)

**8.0 Appendix A** [**68**](#_Toc356732224)

**9.0 References** [**71**](#_Toc356732224)

**1.0 Type of Reviews Used to Increase the Quality of Work Product (Appendix A).**

The type of **reviews applied** to increase the work quality of Appendix A is **walkthrough** and **technical reviews**. In the **technical review** done, as seen through comments labelled “[TR]” in Appendix A, primarily targeted the technical correctness, completeness and consistency of the document. For example, one comment pointed out the incorrect application of a formula in a section describing an algorithm. The reviewer suggested recalculating the results based on the correct equation, ensuring technical accuracy. Another “[TR]” comment flagged an undefined range for the self-selected amount which had no minimum and maximum limits. This highlight allowed the avoidance of potential errors or system abuse experienced by future users. By addressing all the “[TR]” comments, the work product will become technically correct and aligned with professional expectations. It helps to reduce errors and ambiguities, making the document and software more reliable for decision-making or implementation purposes. The technical review ultimately ensures that the work output after being corrected will meet high standards of accuracy, contributing to its acceptance and usability in its intended context.

In the **walkthrough review** done, as seen through comments labelled “[WT]” in Appendix A, focused on improving the software’s and the document’s usability and clarity. For example, one “[WT]” comment noted that a section explaining how a control module can be used to display expiration time, should include visual indicators such as colour coded messages. The reviewer suggested this so that it can be easier for users to quickly identify the status of their ticket or card. This will help to enhance readability without compromising the technical content, ensuring the ease of usage of the system by a broader audience. By incorporating all the walkthrough comments, the document was refined to better cater to user needs. These changes help to improve the software’s usability by providing clearer guidance and enhancing  user experience, while also ensuring that the document’s instructions were intuitive and easy to follow for a diverse audience.

Together, the technical review and walkthrough, ensures that the content would be accurate, complete and in line with professional standards while making sure that the delivery of the document as well will be clear, coherent and readable. These reviews gave feedback that will allow the work product to be transformed into a high-quality document that is both technically sound and easily comprehensible, demonstrating the importance of integrating both review approaches in a comprehensive review process.

**2.0 Testing Principle Followed When Using Reviews in Early Stages of the Development Process.**

The reviews conducted during the document’s developmental process reflect key testing principles, particularly **defect prevention** and **early detection of defects**, which are essential for improving the quality and efficiency of the final work product.

The key principle of **defect prevention** puts emphasis on avoiding issues rather than resolving them later in the development lifecycle which will stack up on cost. Several potential flaws in the document such as inaccuracies in technical content, incomplete information, and unclear structure were identified and addressed proactively through the use of conducting Technical Reviews (TR) and Walkthroughs (WT) in the early stages. For example, the technical review flagged a calculation logic, asking to ensure that the calculation for reducing the balance should include edge cases such as insufficient funds or system error. By rectifying these issues in the documentation before it is finalized, it prevents future downstream issues such as incorrect conclusions or misinterpretation by stakeholders. Research confirms as well that addressing defects early reduces costs and enhances quality as well since early-stage reviews are more cost-effective and mitigates risks (Devi, 2012). To summarize, the defect prevention approach highlights the value of addressing the root cause of initial errors at the inception stage which can ensure a more streamlined and error-free development process.

The early review process also aligns with the testing principle of **detecting defects as soon as possible**. Walkthrough comments are used to improve the document’s flow and readability, ensuring that issues like vague transitions and formatting inconsistencies in the software requirements can be identified at an early stage. An early intervention like this could potentially minimize the cascading effects of these issues since fixing them at a later stage in the process after the requirement has been fully implemented can be time-consuming and impact other sections of the software and document as well. For instance, as seen in the walkthrough feedback in Appendix A, improving the clarity by including visual indicators such as colour coded messages to make it easier for users to quickly identify the status of their ticket or card, could potentially ensure that subsequent sections cater to thoughtful suggestions to improve the clarity of matters which would reduce the overall risk of confusion for users later during deployment. This can be supported through a study by Sherif and Kelly (1992), that shows that early intervention improves overall efficiency and enhances readability and user experience.

By focusing on defect prevention and early detection, the review process helps to optimize cost and effort by addressing issues early, which is significantly less costly than fixing them later. Early reviews identified major technical inaccuracies and minor user interface/experience issues, which will help in enabling a cost-effective improvement process that enhanced the software’s quality and reduced later-stage corrections. This approach helped to suggest improvements in the development of a high-quality and accurate software and document respectively, emphasizing the importance of early-stage reviews as a key quality assurance medium.

**3.0 Why Are Reviews Useful?**

Reviews are essential in the ticket machine project as they enable the early identification of potential issues, reducing risks and costs during the development process. By thoroughly examining system requirements and design, stakeholders can detect inconsistencies, ambiguities, and omissions before these issues escalate into costly challenges during coding, implementation, or testing phases. For example, reviews validate workflows like ticket purchasing and travel card recharging, ensuring they are logically structured and meet user expectations. They also confirm adherence to critical usability and accessibility standards, making interfaces intuitive, consistent, and inclusive for a diverse range of users. These standards are particularly vital for ticket machines, which must cater to individuals with varying levels of technical proficiency and physical abilities.

Moreover, reviews promote collaboration among project stakeholders by fostering clear communication and alignment on the system’s objectives. Developers, designers, and users can share insights to ensure their expectations are reflected in the system’s requirements and design. This collaborative approach minimizes misunderstandings and ensures that the system evolves into a robust, user-friendly, and cost-efficient product.

Addressing potential pitfalls early in the lifecycle through reviews reduces the likelihood of costly rework later. By identifying issues upfront, teams can focus on building a stable and functional system that aligns with user expectations, avoids unnecessary delays, and stays within budget. Additionally, reviews act as a safeguard against misaligned features, allowing stakeholders to verify that the system delivers its intended value to users. Ultimately, reviews serve as a foundation for quality assurance, ensuring that the system is reliable, accessible, and user-centric from the outset.

Another critical benefit of reviews is that they provide a structured mechanism for quality control. By breaking down the system into manageable components during the review process, stakeholders can systematically evaluate each element’s functionality and alignment with the overall project goals. This granular approach allows for targeted improvements, ensuring that each aspect of the system is optimized. Moreover, by involving diverse stakeholders in the review process, including developers, testers, and end-users, teams can benefit from multiple perspectives, which enhances the overall robustness of the system (Pressman & Maxim, 2020).

Reviews also play a vital role in maintaining project momentum and direction. During complex development processes like those involved in building ticket machines, it is easy to stray from the original objectives due to evolving requirements or unforeseen challenges. Reviews provide an opportunity to realign the project with its initial goals, ensuring that any changes are deliberate and contribute positively to the system’s success. This iterative evaluation process ensures that the system remains on track and prevents scope creep, which can lead to delays and increased costs.

Finally, reviews help ensure compliance with regulatory and industry standards, particularly in projects where public accessibility and data security are paramount. By thoroughly evaluating the system’s adherence to these standards, reviews reduce the risk of penalties or reputational damage, providing an additional layer of protection for both the system and its users (Gillies, 2011).

**4.0 What Kinds of Problems Do You Think Reviews Would Locate?**

Reviews are particularly effective at identifying high-level issues related to requirements, design, and usability. They can uncover gaps or ambiguities in specifications, such as missing rules for handling invalid cards or insufficient details about discount eligibility. If these issues remain unresolved, they could result in user confusion or functional inconsistencies. Reviews are also valuable for detecting design flaws, including improper state transitions or errors in handling edge cases, such as repeated transaction cancellations. Additionally, inconsistencies between system modules, like mismatches in how the ticket machine and control module calculate or display balances, are often identified during design reviews.

Another area where reviews are beneficial is in identifying usability issues. They help pinpoint confusing workflows, poorly worded error messages (e.g., “Unknown card”), or unclear instructions that may undermine user confidence. Accessibility concerns are another focus, ensuring that the system accommodates users with disabilities. For instance, a review might highlight a lack of intuitive navigation or insufficient error recovery mechanisms, both of which can be addressed to improve inclusivity.

Policy and workflow validation is another key aspect where reviews can provide value. By examining how the system implements operational policies, reviews can expose inefficiencies or misalignments in processes like ticket purchasing, card recharging, or balance checking. For example, they can verify whether the system properly supports multiple trips within an hour for valid cards or ensures discounts are correctly applied for groups such as children or students.

Reviews also play an important role in detecting communication gaps between design and implementation teams. For example, if developers misinterpret the intended functionality of a specific feature, such as how expiration times should be calculated, this issue can be flagged during the review process. Catching such problems early helps ensure that the final product aligns with the original design, avoiding costly corrections later.

By addressing these areas, reviews make a significant contribution to improving the system’s robustness, usability, and alignment with user needs. They allow potential issues to be resolved proactively, reducing the risk of disruptions during later stages of development.

**5.0 What Kinds of Problems Might Reviews Not Locate?**

Despite their strengths, reviews have limitations and are not capable of addressing every type of issue. For example, implementation-specific problems such as syntax errors, logic flaws, or database misconfigurations typically arise during coding and fall outside the scope of reviews. These issues are best detected through rigorous testing and debugging during the implementation phase.

Reviews also struggle to identify dynamic system behaviors that occur during runtime. Performance issues, like the system’s inability to handle high transaction volumes during peak hours, are difficult to predict through static reviews. For instance, a ticket machine that operates efficiently under normal conditions might encounter delays and user frustration when processing simultaneous transactions during busy periods. Addressing such challenges requires stress testing and performance simulations.

Hardware-related problems present another challenge for reviews. Issues such as card reader malfunctions, printer jams, or hardware overheating in extreme temperatures can only be uncovered through physical testing in real-world conditions. These factors lie beyond the capabilities of traditional review processes.

Unpredictable user behavior is another area where reviews fall short. Scenarios such as users repeatedly ancelling transactions or entering invalid data rapidly can lead to unexpected system behavior. Identifying and addressing these edge cases often requires exhaustive testing to replicate real-world interactions.

Integration and compatibility issues may also go unnoticed during reviews. While the system might work perfectly in isolation, problems can arise when it interfaces with external components like third-party payment systems or database management platforms. For instance, the software might experience unexpected errors when communicating with a payment gateway, highlighting the importance of thorough integration testing.

Security vulnerabilities are another area that reviews cannot fully address. Identifying issues like weak encryption, authentication loopholes, or risks of data exposure requires specialized security assessments beyond the typical review process. Failing to address these vulnerabilities can leave the system exposed to significant threats.

Lastly, reviews are unable to identify challenges related to the physical environment. Factors such as screen glare in outdoor settings, insufficient weatherproofing, or difficulties in operating the system during extreme temperatures often require on-site testing. Ensuring that the system performs reliably in its intended environment demands real-world validation.

**6.0 How Did Our Group Implement It**

Our group implemented the review process for Appendix A through a structured approach utilizing technical reviews (TR) and walkthroughs (WT) to address both technical and usability concerns. This process ensured that the document was both accurate and user-friendly, in line with the project requirements and quality standards.

The first step in the implementation was dividing Appendix A into logical sections such as the ticket machine module, control module, and ticket pricing rules. Each section was assigned to specific team members to focus on either technical aspects or user experience. This division allowed for a thorough evaluation of the document from both perspectives.

In the technical review phase, team members concentrated on identifying ambiguities and inconsistencies in the technical specifications. For instance, one significant issue was the absence of minimum and maximum limits for the self-selected recharge amount in the ticket machine module. This concern was highlighted with the comment “[TR] Define a range for the self-selected amount, such as minimum and maximum limits, to avoid potential errors or system abuse” (Boehm, 1981). Similarly, the control module’s calculation logic for reducing the balance of a travel card was found to be insufficient in handling edge cases, such as transactions involving insufficient funds. This was flagged with “[TR] Ensure the calculation logic for reducing the balance includes edge cases such as insufficient funds or system error. Implement error handling and transaction logging for debugging.”

The walkthrough phase focused on evaluating usability and accessibility. The team identified areas where the document could be improved to enhance user experience. For example, error messages such as “Unknown card” were found to lack actionable guidance for users, leading to the suggestion “[WT] Suggest providing actionable options on the error screen, such as ‘Try Again’ or ‘Contact Support,’ to improve the user experience” (Nielsen, 1993). Another key usability issue was the absence of visual indicators for ticket statuses, which was addressed with “[WT] Recommend including visual indicators such as color-coded messages to make it easier for users to quickly identify the status of their ticket or card” (Nielsen, 1993).

Feedback from both technical reviews and walkthroughs was systematically annotated in Appendix A using “[TR]” and “[WT]” labels. This structured approach enabled effective communication of findings and streamlined the rework process. After incorporating these recommendations, the team conducted a final review to ensure all changes were implemented correctly, resulting in a robust and user-centric document.

**7.0 Team Findings**

The review process identified several critical issues in Appendix A, categorized into technical and usability concerns, which were addressed through targeted recommendations. These findings were crucial in transforming the document into a comprehensive and accurate system specification.

**Technical Issues Identified**

One of the primary technical issues was the lack of defined recharge limits in the ticket machine module. Without these limits, the system was at risk of misuse, such as users inputting excessively large or small amounts. The recommendation “[TR] Define a range for the self-selected amount, such as minimum and maximum limits, to avoid potential errors or system abuse” addressed this gap effectively (Boehm, 1981). Additionally, the calculation logic for reducing the balance of the travel card was flagged as insufficiently robust. The team suggested refining the logic to include edge cases like insufficient funds or system errors, as noted in “[TR] Ensure the calculation logic for reducing the balance includes edge cases such as insufficient funds or system error. Implement error handling and transaction logging for debugging.”

Another technical concern was the ambiguity in discount application rules. For example, the system did not clearly specify how discounts would apply when multiple conditions overlapped, such as a student purchasing a ticket on a public holiday. The team recommended defining a precedence hierarchy for discounts to eliminate confusion and ensure consistent application.

**Usability Issues Identified**

From a usability perspective, one of the most critical issues was the lack of actionable error messages. Messages like “Unknown card” were too vague and could confuse users. The walkthrough team proposed improvements, including “[WT] Suggest providing actionable options on the error screen, such as ‘Try Again’ or ‘Contact Support,’ to improve the user experience” (Nielsen, 1993).

Visual feedback was another area of improvement. The lack of color-coded indicators for ticket statuses made it difficult for users to interpret information quickly. This led to the recommendation “[WT] Recommend including visual indicators such as color-coded messages to make it easier for users to quickly identify the status of their ticket or card” (Nielsen, 1993). Additionally, certain windows lacked intuitive navigation options, such as a “Back” button, which could confuse users. The suggestion “[WT] Ensure that this window includes a back option when performing a backward action” was made to address this.

Finally, the team proposed adding localized tooltips or brief guides for first-time users, as noted in “[WT] Add localized tooltips or brief guides for first-time users to make the option more intuitive and accessible for diverse demographics.”

**Highlighted Improvements**

The recommendations addressed in the reviews resulted in significant improvements:

* Defined recharge limits were introduced to prevent misuse and errors.
* Calculation logic in the control module was refined to account for edge cases.
* Error messages were enhanced with actionable options, improving user clarity and guidance.
* Visual indicators and navigation options were integrated, making the system more intuitive.
* Accessibility features such as localized tooltips were added to cater to a broader audience.

**(2960 words)**

**8.0 Appendix A**

**Appendix A:**

**A Ticket Machine - System Specification Development model:** The V – model **Requirements:**

The ticket machine module:

The traveller shall be informed about

* Payment information
* Ticket prices
* Ticket conditions

The traveller shall be able to buy

* A rechargeable travel card
* A week pass
* A 21-day pass

A traveller with an existing travel card shall be able to

* Recharge the travel card with a self-selected amount of money
* Check the balance of the card The control module:

A traveller with a rechargeable travel card should be able to

* Register a trip, if the trip has not already been registered within the last hour. When the trip is registered, the system shall reduce the balance of the card with the cost of a trip.
* Get a ticket valid for one hour after registration. It should be possible to make multiple trips within one hour of registration.
* Get information about expiration time.
* Get information about the balance of the card.

All travellers shall be informed about

* Whether the ticket is valid or not
* The ticket expiration date

Both the ticket machine and the control module should be accessible to everyone and the user- friendliness is therefore important. Both modules must be

* Understandable
* Intuitive to use
* Consistent in behaviour and design
* Reliable
* Robust
* Fault tolerant

**Ticket prices and discounts:**

Ticket prices and discounts will be set in the administration module. For test purposes you can use the following ticket prices:

|  |  |  |  |
| --- | --- | --- | --- |
| **Category\Ticket Type** | **Single trip\*** | **A week pass** | **A 21-day pass** |
| **Children** | RM1.50 | RM10.00 | RM15.00 |
| **Students** | RM2.50 | RM15.50 | RM25.00 |
| **Adults** | RM3.00 | RM25.50 | RM35.00 |

\*The price for a single trip using a travel card

The discounts only apply to travellers with a travel card.

|  |  |  |  |
| --- | --- | --- | --- |
| **Day \ Category** | **Children** | **Students** | **Adults** |
| Monday to Friday | 0% | 0% | 0% |
| Saturday to Sunday | 100% | 50% | 25% |
| Public holidays | 100% | 60% | 30% |

**User interaction – The ticket machine**

The ticket machine module will be described as the interaction between the user and the system:

**State 1:** Start window:

The traveller can choose one of two options:

1. Buy a ticket or a travel card. The system goes to state 2.
2. Recharge an existing travel card. The system goes to state 3.

**State 2:** Window for choosing the type of ticket or travel card: The traveller can choose one of three options:

1. Travel card. The system goes to state 5.
2. A week card. The system goes to state 6.
3. 21-days card. The system goes to state 6.

**State 3:** Window for reading a travel card.

The traveller holds the travel card in front of the card reader so that the card is read. If the system is recognizing the card, the system goes to state 5. If the system does not recognize it or the card, for any reason, is not valid, the system goes to state 4.

**State 4:** Window for displaying an error message.

The system displays the error message “Unknown card” on the screen. After a few seconds, the system goes back to the start window, state 1.

**State 5:** Window for making deposits to the travel card

The traveller enters the amount he/she wants to deposit to the card. The system goes to state 6.

**State 6:** Confirmation window.

Information about what is purchased and what is payable is displayed on the screen. The traveller can choose one of two options:

1. The traveller confirms the purchase, by pressing the confirm button. The system goes then to state 7.
2. Cancel: The traveller cancels the transaction by pushing the cancel button. The system goes the back to the start screen, state 1.

**State 7:** Payment window:

The following message is displayed on the screen: “Insert you credit card”.

Although not mentioned in the description above, all windows will have a cancel button.

**User interaction – The control module**

The control module will be described as the interaction between the user and the system. Travelers with a week or 21-day pass does not need to register the trip. However, the control module can be used to display expiration time.

**State 1:** Window for reading a ticket or a travel card.

The traveller holds the ticket or the travel card in front of the card reader so that it is read. If the system is recognizing the ticket or the travel card, the system goes to state 2. If the system, for any reason, does not recognize the ticket or the travel card, the system goes to state 5.

**State 2:** The system checks the expiration time:

For 7-day or 21-day pass: If the time is before the expiration date, the system goes to state 4. If not, the system goes to state 5.

For travel cards: If it is less than one hour since the last registration, the system goes to state 4. If there is one hour or more than one hour since the last registration, the system goes to state 3.

**State 3:** The system checks the balance of the travel card.

If the balance is above or equal the cost of the trip, the balance will be reduced by the cost for the trip. The system then goes to state 4.

If the balance is below the cost of the trip, the system goes to state 5.

**State 4:** Information window.

The system will show information about expiration time for both tickets and travel cards. For traveling cards, the cost of the trip and the balance of the card will also be displayed. After a few seconds, the system goes back to state 1.

**State 5:** Error message window.

The system displays an error message on the screen with the following message: “The card or ticket is not valid. You must pay before entering”.

After a few seconds, the system goes back to state 1.

**9.0 References**

Devi, T. (2012). Improving Quality of Software through Formal Inspection. International Journal of Engineering Research and Application (IJERA), 2(1), 552-557.

Gillies, A. (2011). *Software quality: Theory and management* (3rd ed.). Cengage Learning EMEA.

Karna, K. N. (1984). Software engineering economics. Proceedings of the IEEE, 72(6), 751-751. <https://doi.org/10.1109/proc.1984.12932>

Nielsen, J. (1994). *Usability Engineering*. Morgan Kaufmann.

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.

Sherif, Y. S., & Kelly, J. C. (1992). Improving software quality through formal inspections. Microelectronics Reliability, 32(3), 423–431. <https://doi.org/10.1016/0026-2714(92)90072-s>

**Part 3: Reflection**

**1. Tung Qi Yong**

In this assignment, I was in charge of playing different roles for each process. For part 1, I acted as a Software Tester, specifying test design and test case for the Automated Teller Machine System (ATMS). For part 2, I will be a Software Tester and a Usability Analyst to implement the review of the work product in Appendix A. Each role required different responsibilities and perspectives to implement the software testing process.

In Part 1 of this assignment, my main task was to apply two black box test techniques to the ATMS specifically on the Feature 007 Transfer, the test techniques are Equivalence Partitioning and Boundary Value Analysis. For example, during the test design specification process, I categorised the user input for transfer amount into three situations: when the amount value is zero as invalid, the amount value below the account balance as valid, and the amount value exceeds the account balance as invalid. In short, completing the task in part 1 of this assignment enhanced my skill in applying black-box test techniques. It also made me realise the importance of structured test case design for identifying potential defects early.

On the other hand, part 2 of the assignment is about reviewing the ticket machine system by implementing two review techniques, which are technical review and walkthrough. During the first process, I played the role of software developer to conduct the technical review for the word product in Appendix A. I focused on the technical implementation, logic for key features, and feasibility concerns. For example, the logic for the balance reduction for travel cards is not fully specified in the work product. The system should also handle errors like insufficient amounts to ensure the system is robust enough. Next, in the walkthrough review, I used a usability analyst perspective to evaluate the work product’s accessibility and user experience. For example, I suggested the system should implement the screen reader feature to make sure the system can be accessed by people with visual impairment. The main issue faced in this process is about to balance the technical aspect and user-centric design considerations. However, the review process is implemented under a group of reviewers with different roles. Thus, the issue can be easily solved by collecting the opinions of different groupmates and brainstorming solutions.

In conclusion, this assignment provides deep experience in the software testing field by implementing the test and reviewing the specific work product. I improved my ability to design effective tests using different black box test techniques. In part 2, I gained a deeper understanding of software review processes and received the experience of collaborating with others to review the work product. The three different roles I played also helped me to improve my critical thinking skills and use different perspectives to test the work product. Last but not least, This assignment not only enriched my skills in software testing and review but also expanded my understanding of how the actual process is implemented, especially the workflow for reviewing and implementing the test with a group of people.

**2.** **Syehrran A/L Arulsamy**

In this assignment, I took on distinct roles for each part of the project, which allowed me to gain a well-rounded understanding of software testing and review processes. For Part 1, I acted as a Software Tester, designing test cases and test specifications for the withdrawal functionality in the Automated Teller Machine System (ATMS). For Part 2, I assumed the roles of a Domain Expert and an End-User Representative during the review process for Appendix A. These roles required me to approach the assignment from both technical and user-centric perspectives, enabling me to refine my analytical and collaborative skills.

In Part 1, my primary responsibility was to develop the Test Design Specifications and Test Case Specifications for the ATMS withdrawal feature using three specific testing techniques: Decision Table Testing, State Transition Testing, and Use Case Testing. I categorized user input conditions and outcomes to validate the system's behavior under various scenarios. For example, I identified test conditions such as valid transaction types, valid account types, and sufficient account balances to ensure the system processed withdrawals correctly. One key aspect was analyzing state transitions, such as ensuring the system responded appropriately to errors like invalid account types or insufficient funds. By documenting these conditions and corresponding outputs in a decision table, I ensured comprehensive coverage of valid and invalid cases.

In the State Transition Testing, I mapped the withdrawal process's various states, including transaction initiation, account selection, amount validation, and receipt printing. This approach highlighted critical interactions between states and helped identify potential flaws, such as improper handling of invalid inputs. For instance, I tested edge cases where users exceeded their daily withdrawal limits or attempted to withdraw amounts greater than the ATM’s available cash. These scenarios reinforced the importance of systematically evaluating both positive and negative pathways to ensure system robustness. Completing this part of the assignment deepened my understanding of structured testing methodologies and emphasized the value of precise documentation in ensuring alignment with system requirements.

In Part 2, I transitioned to the roles of a Domain Expert and an End-User Representative to conduct reviews of Appendix A using Technical Review (TR) and Walkthrough (WT) techniques. As a Domain Expert, I evaluated the technical implementation of the ticket machine system, focusing on correctness and feasibility. For example, I identified that the recharge feature lacked defined minimum and maximum limits, which could lead to system abuse or errors. My recommendation to establish these limits ensured that the feature aligned with domain-specific requirements and reduced potential risks. Additionally, I highlighted gaps in the balance deduction logic, such as its inability to handle edge cases like insufficient funds. This role allowed me to approach the system with a real-world perspective, ensuring technical solutions were both practical and effective.

As an End-User Representative, my focus shifted to improving usability and accessibility. I assessed the system’s user interface and interaction flow, recommending enhancements to improve the overall user experience. For instance, I proposed the inclusion of color-coded indicators to help users quickly identify ticket statuses and actionable error messages, such as “Try Again” or “Contact Support,” to guide users through issues effectively. I also suggested adding tooltips for first-time users to make the system more intuitive and accessible to a diverse user base. Balancing technical and user-centric considerations was a challenge, but collaborative discussions within the group allowed us to address these concerns comprehensively.

Throughout this assignment, I faced challenges in understanding and documenting the detailed testing and review processes. For instance, in Part 1, I initially struggled to map the test design specifications to the withdrawal feature’s functionality. However, consistent feedback and collaboration with my group helped me overcome these hurdles. Similarly, in Part 2, balancing the technical accuracy of the system with user-centric design required a nuanced approach. Group discussions played a crucial role in resolving conflicts and refining our recommendations.

This assignment has significantly enhanced my understanding of software testing and review methodologies. In Part 1, I gained practical experience in applying black-box testing techniques to identify potential vulnerabilities and ensure system reliability. In Part 2, I developed a deeper appreciation for the role of reviews in improving software quality. My roles as a Domain Expert and End-User Representative allowed me to explore both technical and user-focused perspectives, emphasizing the importance of collaboration and diverse viewpoints in delivering high-quality systems.

In conclusion, this assignment provided me with invaluable insights into the workflow and challenges of software testing and review. By taking on distinct roles, I was able to develop critical thinking skills and approach problems from multiple angles. The knowledge and skills gained through this experience will undoubtedly be instrumental in future software development and testing projects.

**3. James Hwang Qi Leong**

I am James Hwang Qi Leong, and I am a Software Tester responsible for completing tasks assigned by our team leader, Syehrran. In this project, my main role was to handle Part 1: Test Design, focusing on state transition testing, decision table testing, and creating test cases for the F007: Transfer functionality in the ATMS project. For Part 2: Static Testing Techniques – Reviews, I worked on evaluating the usefulness of reviews, identifying the kinds of problems reviews can locate, and understanding their limitations. These activities helped me develop a better understanding of software quality assurance and how testing ensures a system’s success.

In Part 1, I focused on designing and documenting test cases for the F007: Transfer functionality in the ATMS project. My tasks involved applying state transition testing and decision table testing to verify the system’s behavior under different scenarios. Using state transition testing, I analyzed how the system moved between various states, such as choosing the transaction type, selecting sending and receiving accounts, entering the transfer amount, and completing the transaction. My main goal was to ensure the system handled these transitions correctly and identify potential issues, such as errors with invalid accounts, insufficient balances, or improper card ejections. Referring to the Transfer State Transition Diagram, I validated important paths like retry mechanisms, cancellation, and edge-case scenarios, such as returning to the transaction menu after an invalid entry. This helped me understand how the system’s states interact and emphasized the importance of verifying both valid and invalid pathways to ensure reliability and robustness.

Using decision table testing, I systematically tested how the system responded to various combinations of inputs and outputs for the F007: Transfer functionality. For instance, I developed test cases to check valid and invalid transfer amounts, invalid recipient accounts, and cases where users attempted to transfer money to the same account. The Transfer Decision Table provided a clear framework for documenting conditions like valid transaction types, account validation, and expected outputs. Outputs such as displaying error messages, retrying, completing transactions, or ejecting cards were thoroughly tested. This approach helped me uncover gaps in decision-making logic and confirm that the system handled all scenarios appropriately. Additionally, I created test cases to verify scenarios like transferring amounts exceeding the account balance or continuing a transaction after an error. Writing test cases in a structured format ensured they were aligned with system requirements and made it easier to trace and test specific scenarios.

In Part 2, I conducted reviews of the ticket machine specification through technical reviews and walkthroughs. My role was to assess whether reviews were useful, identify problems reviews could uncover, and understand the types of issues reviews might not detect. Reviews proved essential for catching issues early in the development process. For example, I found unclear instructions for handling invalid travel cards and vague rules for ticket pricing on weekends. If unresolved, these issues could have caused confusion for users and developers. Reviews also fostered collaboration among team members, helping us refine expectations and improve the design.

Reviews were effective at uncovering high-level issues in requirements, design, and usability. They revealed gaps in specifications, such as missing error recovery steps and inconsistencies in applying discounts. Usability issues, like poorly worded error messages (“Unknown card”), were also addressed. These efforts improved the system’s clarity and ensured it was accessible to diverse users. However, reviews do have limitations. They cannot uncover implementation-specific bugs, such as coding errors or performance issues, which require dynamic testing. For example, reviews cannot predict how the system will perform under high transaction volumes during peak hours. Similarly, hardware issues, like printer malfunctions or overheating, require physical testing. These limitations show that while reviews are useful, they must be complemented with other testing methods to ensure the system’s success.

This project taught me how structured testing and reviews work together to improve system reliability. Through state transition testing and decision table testing in Part 1, I developed a systematic approach to identifying and addressing potential vulnerabilities in the system. Part 2 enhanced my critical thinking skills by teaching me how to evaluate specifications and identify ambiguities early. These activities also highlighted the importance of collaboration, clear documentation, and attention to detail in ensuring software quality.

Overall, this project provided valuable hands-on experience in software testing and reviews. I gained practical knowledge in designing comprehensive test cases, conducting reviews, and understanding the role of quality assurance in creating reliable and user-friendly systems. These skills will be invaluable in future software development and testing projects.

**4. Toh Kar Ming**

For Part 1 of this assignment, I undertook the role of a software tester for Company ABC to develop test design specifications and test case specifications for the Automated Teller Machine System (ATMS). My specific responsibility was the balance inquiry (F008) function. For Part 2, I contributed by drafting the essay sections on “What type of reviews did your group use to increase the quality of work product (Appendix A)?” and “Which testing principle is followed when using reviews in the early stages of the development process?”.

Initially, in executing my Part 1 responsibilities, I faced significant confusion regarding the flow of the documentation as well as expectations for the assignment. Gaining clarity required considerable time, and my group members and I frequently sought guidance from Dr. Lee to understand the requirements better. However, this process proved to be a valuable learning experience. The primary challenge for me was grasping the structure and purpose of the formal documentation and its alignment with the ATMS system’s functionality. Over time, through persistent inquiries and thorough analysis, I gained a comprehensive understanding of the documentation framework and developed a strong appreciation for its systematic arrangement. For instance, the systematic compilation of multiple test conditions and coverages and the further mapping of these combinations to the respective test cases in the test identification section of the test design specification really improved my understanding of how organized and traceable testing processes ensures comprehensive coverage and alignment with system requirements. Despite the initial learning curve, once I was able to visualize the entire flow of the documentation and structure, I could efficiently complete my tasks, having recognized how each component interconnected. This assignment provided me with significant insights into the formal documentation of testing procedures, which I believe will serve as a foundational skill in my future career in software development.

In Part 2 of the assignment, I was responsible for addressing the essay questions “What type of reviews did your group use to increase the quality of the work product (Appendix A)?” and “Which testing principle is followed when using reviews in the early stages of the development process?” This involved analysing the review techniques selected for this section, that is the technical review and walkthrough and evaluating their qualities and impact on the work product after their application. The task required critical thinking to assess the application static testing techniques and to apply theoretical knowledge to practical scenarios. As I analysed the output following the application of these review methods to Appendix A, I found myself reflecting on potential improvements that might have been overlooked due to the inherent limitations of the selected review techniques as well. This made me understand why there was so many different types of review methods in the first place and how each of them is recommended for different scenarios. This process overall, deepened my understanding of the testing principle “Early testing saves time and cost,” as it emphasized the value of identifying defects early in the software development lifecycle through reviews. Additionally, this experience enhanced my analytical and collaborative skills, allowing me to better appreciate the systematic nature of static testing techniques. Overall, I gained valuable insights into how reviews can improve software quality, as well as a nuanced understanding of the strengths and weaknesses of technical reviews and walkthroughs.