**Acceptance Criteria for EasyPay Merchant & Utility Bill Payment:**

| 1 | The system must deduct a 1% service charge from the customer's balance for each merchant payment. |
| --- | --- |
| 2 | If the 1% service charge is less than 5 TK, a minimum fee of 5 TK must be applied. |
| 3 | If a customer makes a transaction greater than 5000 TK, they must receive a 10% cashback of the transaction amount. |
| 4 | If a customer makes a transaction greater than 10,000 TK, they must receive a 20% cashback. |
| 5 | The maximum cashback amount must not exceed 3000 TK, even if 20% of the transaction is greater than this limit. |
| 6 | No cashback should be applied for utility bill payments but service charge 1% deducted. |
| 7 | The system must ensure the correct amount is deducted from the customer's balance without any cashback calculations. |
| 8 | If a transaction fails, no service charge should be deducted, and no cashback should be granted. |

**Acceptance Criteria for EasyPay Loan Feature**

|  |  |
| --- | --- |
| 1 | A customer can apply for a loan only if their balance is less than 100 TK. |
| 2 | The maximum loan amount a customer can apply for is 20,000 TK. |
| 3 | If the loan is repaid within 30 days from the loan initiation date, no interest will be charged. |
| 4 | If repayment is delayed beyond 30 days, a daily compound interest of 1.8% will be applied to the remaining amount. |
| 5 | A customer can apply for a new loan only if they have repaid at least 50% of the remaining outstanding loan. |
| 6 | The loan amount should be credited to the customer’s account immediately after approval |
| 7 | Loan repayments must be automatically deducted from the customer’s balance when funds are available. |
| 8 | Customers must receive reminders before the 30-day repayment deadline. |
| 9 | If a customer tries to apply for a loan with more than 100 TK balance, the request must be denied. |
| 10 | If a customer applies for a loan without repaying 50% of their existing loan, the request must be denied. |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **Priority Order** | **Feature ID** | **Title** |
| 1 | 4 | Doctors Appointment |
| 2 | 6 | Generate Patient ID Card |
| 3 | 3 | Bill Collection |
| 4 | 5 | Serial Management |
| 5 | 1 | Process Patient Test Report |
| 6 | 2 | |  | | --- | |  |  |  | | --- | | Deliver Patient Report Online | |

Doctors Appointment (Feature 4) is independent, so it is executed first.

Generate Patient ID Card (Feature 6) depends on doctors’ appointment (Feature 4), so it comes next. Bill Collection (Feature 3) depends on patient ID generation (Feature 6), meaning it must be done afterward. Serial Management (Feature 5) depends on bill collection (Feature 3), so it follows. Process Patient Test Report (Feature 1) depends on both bill collection (Feature 3) and serial management (Feature 5). Deliver Patient Report Online (Feature 2) is the last step, as it depends on the test report being processed (Feature 1).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **For merchant & utility bills.**   |  |  |  |  | | --- | --- | --- | --- | | **SL** | **CHECK POINT** | **YES** | **NO** | | **01** | Does the system deduct a 1% service charge from the customer's balance for each merchant payment? |  |  | | **02** | If the 1% service charge is less than 5 TK, does the system apply a minimum fee of 5 TK? |  |  | | **03** | If a customer makes a transaction greater than 5000 TK, does the system provide a 10% cashback? |  |  | | **04** | If a customer makes a transaction greater than 10,000 TK, does the system provide a 20% cashback? |  |  | | **05** | Does the system ensure that the maximum cashback amount does not exceed 3000 TK? |  |  | | **06** | Does the system apply cashback to utility bill payments? |  |  | | **07** | Does the system deduct a 1% service charge for utility bill payments? |  |  | | **08** | If a transaction fails, does the system deduct the service charge? |  |  | | **09** | If a transaction fails, does the system grant cashback? |  |  | | **For Loan** | | | | | **1** | Does the system log all transactions with details of amount, service charge, and cashback (if applicable)? |  |  | | **2** | Can a customer apply for a loan only if their balance is less than 100 TK? |  |  | | **3** | Is the maximum loan amount a customer can apply for 20,000 TK? |  |  | | **4** | If the loan is repaid within 30 days from the loan initiation date, will no interest be charged? |  |  | | **5** | If the repayment is delayed beyond 30 days, is a daily compound interest of 1.8% applied to the remaining amount? |  |  | | **6** | Can a customer apply for a new loan only after repaying at least 50% of the remaining outstanding loan? |  |  | | **7** | Is the loan amount credited to the customer’s account immediately after approval? |  |  | | **8** | Are loan repayments automatically deducted from the customer’s balance when funds are available? |  |  | | **9** | If a customer tries to apply for a loan with more than 100 TK balance, will the request be denied? |  |  | | **10** | If a customer applies for a loan without repaying 50% of their existing loan, will the request be denied? |  |  |        |  |  | | --- | --- | | **Static testing** | **Sanity testing** | | Testing done without executing the code, focusing on reviewing documents and code structure. | A quick, focused test to verify that a specific feature or bug fix works correctly. | | Early in the development phase . | After receiving a build with minor changes or bug fixes. |  |  |  | | --- | --- | | **Retest** | **Regression test** | | To check whether a specific defect has been fixed. | To check if new changes have affected existing functionalities. | | Only focuses on failed test cases. | Covers previously tested functionalities. | | Retesting a fixed login bug. | Testing login, signup, dashboard after adding a new feature. | |

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Acceptance criteria** | **Acceptance testing** |
| A set of predefined conditions that must meet to be accepted. | Actual test done to verify if the software meets user’s requirements. |
| Defined before development starts. | Perform before releasing the product. |
| Example: user must receive OTP for login. | Example:manually checking if OTP is received on login attempt. |

|  |  |
| --- | --- |
| **Decision table testing** | **Decision testing** |
| Uses a table-based approach to test different input conditions and their outcomes. | Focuses on testing decision-making structures like if-else and switch statements. |
| Complex business logic with multiple conditions | Ensuring all decision points in code are executed. |
| Testing insurance premium calculations with different inputs. | Checking if all "if-else" branches in code execute correctly. |

|  |  |
| --- | --- |
| **State transition test** | **Statement coverage** |
| Testing system behavior based on different states and transitions. | Ensuring every line of code executes at least once. |
| Focus on Input conditions and state changes. | Focus on executing all statements in the code. |
| example: ATM card state changes (Inserted → PIN Entered → Approved). | example: Checking if all lines in a function run during tests. |

|  |  |
| --- | --- |
| **Data testing** | **Database testing** |
| Validating data integrity and correctness. | Checking database structure, schema, and queries. |
| ensuring correct data storage and retrieval. | Verifying indexes, triggers, constraints, and queries. |
| Checking if order details save correctly. | Ensuring a foreign key constraint works. |

|  |  |
| --- | --- |
| **Test planning** | **Test strategy** |
| A detailed document describing test scope, resources, schedule, and deliverables. | A high level document defining the overall testing approach. |
| It is used on project level only. | It is used on an organization level. |
| “We will test the login feature in Sprint 2 using Selenium”. | "We follow an automation-first approach for all projects." |

|  |  |
| --- | --- |
| **Unit testing** | **Component testing** |
| Testing individual function or method. | Testing a complete module with dependencies. |
| Single function or class. | Multiple related units together. |
| example: Testing a "calculateTotal()" function. | Example: testing a shopping cart module with add ,remove and checkout features. |