# **Credit Card Statement Generator**

## **Intern Software Development Assignment**

### **Overview**

In this assignment, you will develop a banking application that generates credit card statements in PDF format. Each intern will be responsible for creating a complete statement generator for a different bank, based on that bank's existing statement format. This project will simulate a real-world development task where you'll work with databases, document generation, and financial data processing.

### **Assignment Objectives**

1. Develop a standalone application that connects to a SQL Server database
2. Extract and process customer credit card data
3. Generate professional PDF statements matching your assigned bank's format
4. Implement proper data validation and error handling
5. Create and execute comprehensive test cases
6. Present a working demo with sample data

### **Bank Assignment**

Each intern will be assigned one of the following banks to implement:

1. Maybank (Malaysia)
2. CIMB (Malaysia/Singapore)
3. HSBC (Global)
4. DBS (Singapore)
5. Standard Chartered (Global)
6. OCBC (Singapore)
7. Public Bank (Malaysia)
8. RHB (Malaysia)

### **Required Features**

#### **1. Database Operations**

* Connect to a SQL Server database
* Create appropriate table schemas for customer data, accounts, transactions, rewards, etc.
* Implement efficient query operations to retrieve statement data

#### **2. PDF Statement Generation**

* Generate statements that closely match your assigned bank's format
* Include all standard sections (customer information, transactions, summary, rewards, etc.)
* Support multi-page statements when needed
* Ensure consistent formatting throughout the document

#### **3. Multi-language Support**

* Implement bilingual support based on your assigned bank's languages
* Ensure proper display of all language characters

#### **4. Data Processing**

* Calculate balances, minimum payments, and due dates correctly
* Format currency values according to regional standards
* Process transaction data correctly (debits, credits, fees, etc.)

#### **5. Testing & Quality Assurance**

* Create and execute at least 15 test cases covering different aspects of your application
* Document test results and bug fixes
* Ensure the application is robust and handles edge cases appropriately

### **Deliverables**

1. **Source Code**: Complete application with comments and documentation
2. **Database Scripts**: SQL scripts for creating and populating the database with test data
3. **Test Cases**: Document with test cases and results
4. **User Guide**: Brief document explaining how to set up and use your application
5. **Sample PDFs**: Generated statements showcasing your application's output
6. **Presentation**: A 10-minute presentation demonstrating your solution

### **Getting Started**

1. **Research Phase (Week 1)**
   * Find your assigned bank's statement sample (Google image search, bank websites, etc.)
   * Analyze the statement format, required sections, and unique features
   * Design your database schema based on the required data
2. **Development Phase (Weeks 2-3)**
   * Set up your development environment
   * Create the database and populate it with test data
   * Develop the core functionality for generating statements
   * Implement PDF generation with proper formatting
3. **Testing Phase (Week 4)**
   * Create and execute test cases
   * Fix any bugs or issues
   * Optimize performance and ensure reliability
4. **Presentation Phase (Final Week)**
   * Prepare your demo with sample data
   * Document your solution and create your presentation
   * Present your work to the team

### **Technical Requirements**

* Programming Language: Your choice (recommended: C#, Java, Python)
* Database: SQL Server (provided)
* PDF Generation: Any appropriate library for your chosen language
* Version Control: Use Git for source code management
* IDE: Your choice

### **Resources Provided**

* Access to SQL Server database instance
* Sample customer and transaction data structure
* Maybank statement sample for reference
* Basic project template (optional)

## **Evaluation Criteria**

Your assignment will be evaluated based on the following criteria:

### **1. Functionality (40%)**

| **Criteria** | **Description** | **Weight** |
| --- | --- | --- |
| Database Operations | Correct implementation of database connections, queries, and data handling | 10% |
| PDF Generation | Accuracy of the generated statements compared to the bank's format | 15% |
| Calculations | Correctness of financial calculations (balances, minimum payments, etc.) | 10% |
| Error Handling | Appropriate handling of errors and edge cases | 5% |

### **2. Technical Implementation (25%)**

| **Criteria** | **Description** | **Weight** |
| --- | --- | --- |
| Code Quality | Clean, well-structured, and documented code | 10% |
| Database Design | Appropriate schema design and query efficiency | 5% |
| Performance | Efficient processing of data and PDF generation | 5% |
| Security | Proper handling of sensitive financial data | 5% |

### **3. Testing (20%)**

| **Criteria** | **Description** | **Weight** |
| --- | --- | --- |
| Test Case Coverage | Comprehensive test cases covering various scenarios | 10% |
| Test Execution | Successful execution of test cases and documentation of results | 5% |
| Bug Resolution | Identification and resolution of issues | 5% |

### **4. Presentation and Documentation (15%)**

| **Criteria** | **Description** | **Weight** |
| --- | --- | --- |
| Documentation | Clear and comprehensive documentation of your solution | 5% |
| Demo | Effective demonstration of your application's capabilities | 5% |
| Knowledge | Understanding of the system and ability to answer questions | 5% |

### **Grading Scale**

* **Excellent (90-100%)**: Exceeds requirements with exceptional quality
* **Good (80-89%)**: Meets all requirements with high quality
* **Satisfactory (70-79%)**: Meets most requirements with acceptable quality
* **Needs Improvement (60-69%)**: Meets some requirements but has significant issues
* **Unsatisfactory (<60%)**: Fails to meet most requirements

## **Additional Guidelines**

### **Finding Statement Samples**

1. Search for "[Bank Name] credit card statement sample" on Google Images
2. Check bank websites for sample statements in their documentation
3. Look for financial education resources that may contain statement examples
4. Ensure you're using the most recent statement format

### **Data Privacy Considerations**

1. Use fictional data for all customer information
2. Never use real account numbers or personal information
3. Mask sensitive portions of account numbers in your output

### **Testing Recommendations**

1. Test with various balance amounts (zero, standard, very large)
2. Test with different transaction volumes (few, many, none)
3. Test bilingual features with different language settings
4. Test edge cases like statement periods crossing month boundaries
5. Test with unusual transaction descriptions and amounts

### **Common Challenges to Prepare For**

1. Accurate replication of statement formatting and layout
2. Handling of multi-page statements with proper pagination
3. Correct calculation of financial values and dates
4. Efficient processing of large transaction datasets
5. Proper display of bilingual content