We Cheat Em Bank

Testing Plan | Milestone 2

Cameron Seppi, Vadim Pidoshva, Jessica Adams, Trajan Clark

TEAM 10, CS-4230, Dr. Jingpeng Tang

Table of Contents

Scope

Account Management

Test Descriptions:

- Verify that the system generates a unique account number for each new customer
- Verify a customer has exactly one savings account at a time.

Priority: #1
Dependencies: None
Is This Being Tested?: Yes, Statement Coverage Testing

Monthly Transactions

Test Descriptions:

- Verify that the customer can only initiate a loan up to the proper number of times in a month.
- Verify that the customer can only make a payment on a loan up to the proper number of times in a month.
- Verify that the customer can only deposit to their savings account up to the proper number of times in a month.
- Verify that the customer can only withdraw from their savings account up to the proper number of times in a month.

Priority: #1
Dependencies: None
Is This Being Tested?: Yes, Statement Coverage Testing

Monthly Change

Test Descriptions:

- Verify that interest is calculated on loans and savings.
- Verify that interest is added to the accounts.
- Verify that statements generated for each customer show the beginning balances, account activity, interest, and ending balances.

Priority: #1
Dependencies: None
Is This Being Tested?: Yes, Statement Coverage Testing

Initiate a loan

Test Descriptions:

- Verify the minimum loan amount is $500.
- Verify the maximum loan amount is $50,000.
- Verify that the user can't take out more than 3 loans.
- Verify that the interest rates have an increment of 0.25%.
- Verify that the interest rate matches the inputted rate.

Priority: #1
Dependencies: None
Is This Being Tested?: Yes,

Loan Payment

Test Descriptions:

- Verify that the late fee of $50 is added to the loan balance if the minimum payment is not made.
- Verify that the minimum payment listed in the bill is correct.
- Verify that a payment can't be higher than the loan balance.
- Verify that the loan account closes when the balance is paid.
- Verify that a new loan can be made once the loan account is closed if the max number of loans was reached previously.

Priority: #1
Dependencies: None
Is This Being Tested?: Yes,

Savings Account

Test Descriptions:

- Verify that you can deposit money into your savings account.
- Verify that you can withdraw money from your savings account.
- Verify that the savings rate is ¼ of the inputting loan rate.
- Verify that accounts do not have fractions of cents and rounding is applied

Priority: #1
Dependencies: None
Is This Being Tested?: Yes,

Testing Strategies

Equivalence Class Testing

Strategy One - Equivalence Class Testing

- **Strategy Type**: Equivalence Class Testing
- **White/Black Box**: Black Box
- **Technique**: Manual Testing

Entry Criteria:

- The user must be able to perform all core operations of the WeCheatEm Bank from start to finish. These include initiating loans, making payments on loans, depositing from savings, and withdrawing from savings. The system should also handle file inputs/outputs and display the current balances.

Exit Criteria:

- 80% of tests pass with no critical bugs.
- The system operates as expected under normal conditions (e.g. all functions are working as defined).

Command	Equivalence Classes	Valid Input	Invalid Input
Account Creation	Valid: Account numbers are automatically generated.	System generates a unique account number for each customer.	Duplicate account numbers generated for different customers.
	Valid: One savings account per customer	A customer has exactly one active savings account.	No savings account or more than one savings account for a customer.
	Invalid: Duplicate account number	-	System allows non-unique account numbers.
	Invalid: Multiple savings accounts	-	System allows multiple savings accounts per customer.
Monthly Transactions	Valid: Actions occur within limits	Each monthly action (loan initiation, deposit, etc.) can be repeated if business rules allow.	Actions exceed the permitted frequency or quantity.
	Invalid: Excessive action frequency	-	Actions are taken more times than allowed within a month.
Monthly Change	Valid: Interest calculated and added correctly	Interest is calculated for both loans and savings, and added to accounts.	Interest calculations are incorrect, or interest is not added properly.
	Valid: Accurate statements	Statements display beginning balances, activities, interest, and ending balances correctly.	Statements show incorrect or incomplete data.
	Invalid: Incorrect interest calculations	-	Interest not calculated correctly or not applied to accounts.
	Invalid: Inaccurate statements	-	Statements fail to show correct account activity or details.

Initiate a Loan	Valid: Loan amount within limits	Loan amounts are $500 or more, and up to $50,000.	Loan amount below $500 or above $50,000.
	Valid: Proper loan count and interest increments	Customers can have up to 3 loans, with interest in 0.25% increments matching the input.	More than 3 loans allowed, or interest set not in 0.25% increments.
	Invalid: Loan amount out of range	-	Loan amount below minimum or above maximum limits.
	Invalid: Exceeding loan limits	-	More than 3 loans are permitted.
	Invalid: Incorrect interest rate	-	Interest rate does not match input or isn't in the correct increment.
Loan Payment	Valid: Late fee and payment limits	$50 late fee added if minimum payment is missed, and payment does not exceed loan balance.	Payment above loan balance or late fee not applied when required.
	Valid: Loan closure on zero balance	Loan account closes automatically when balance is paid.	Loan remains open after balance is fully paid.
	Invalid: Overpayment	-	System allows payment higher than the loan balance.
	Invalid: Incorrect late fee application	-	$50 late fee not applied when minimum payment isn’t met.
Savings Account	Valid: Deposit, withdrawal, and interest rate	Deposits and withdrawals are within account limits, and savings interest is ¼ of loan rate.	Deposits or withdrawals exceed limits, or savings rate isn't ¼ of loan rate.
	Valid: Proper rounding of cents	Rounding is applied to avoid fractions of cents in transactions.	Fractions of cents exist in the accounts, or incorrect rounding is used.
	Invalid: Exceeded deposit/withdrawal limits	-	Exceeding deposit/withdrawal limits set

			by the system.
	Invalid: Incorrect savings interest calculation	-	Savings interest rate is not proportional to the loan rate (¼).
	Invalid: Rounding errors	-	Incorrect rounding leading to fractions of cents in balances.

Boundary Value Testing

Strategy Two – Boundary Value Testing

- **Strategy Type**: **Boundary Value Testing**
- **White/Black Box**: Black Box
- **Technique**: Manual Testing

Entry Criteria

1. All boundary values and edge cases for customer accounts, loans, and savings activities have been identified and documented.
2. The system's input constraints, such as loan amounts, interest rates, and monthly activities, are clearly defined.

Exit Criteria

1. All test cases have been executed.
2. Results confirm that the system handles all boundary values and edge cases as expected.

Command	Expected Input	Description	Expected Outcome
Account Management	Valid: Create a new account for a customer	Verify that the system generates a unique account number for each new customer	Each new customer gets a unique account number, no duplicates exist.
	Invalid: Create multiple accounts with the same name	Verify that the system enforces uniqueness for customer accounts	The system returns an error indicating that the account name already

			exists.
	Invalid: Create a second savings account for the same customer	Verify that a customer can have only one savings account at a time	The system returns an error indicating the customer can only have one savings account.
Monthly Transactions	Valid: Execute transactions within the monthly limit	Verify that transactions are processed within the proper limits	All transactions are executed successfully.
	Invalid: Exceed the allowed monthly transaction limit	Verify that exceeding the monthly transaction limit triggers a restriction	The system returns an error indicating the transaction limit has been reached.
Monthly Change	Valid: Calculate interest on a positive balance	Verify that interest is correctly calculated on loans and savings for a positive balance	The correct interest is added to the account balance.
	Invalid: Calculate interest on a negative or zero balance	Verify that interest is handled correctly for zero or negative balances	The interest is $0.00, with no changes to the balance.
	Valid: Generate a monthly statement for an active account	Verify that monthly statements include all required details	The statement shows beginning balance, transactions, interest, and ending balance.
	Invalid: Generate a statement for a closed account	Verify that statements cannot be generated for closed accounts	The system returns an error stating the account is closed.
Initiate a Loan	Valid: Apply for a loan with the minimum amount ($500)	Verify that the minimum loan amount requirement is enforced	The loan application is approved for $500.
	Invalid: Apply for a loan just below the minimum ($499.99)	Verify that the system rejects loans below the minimum amount	The system returns an error indicating it is below the minimum limit.
	Valid: Apply for a loan with the maximum amount ($50,000)	Verify that the system supports the maximum loan amount	The loan application is approved for $50,000.
	Invalid: Apply for a loan just above the maximum ($50,000.01)	Verify that the system rejects loans above the maximum amount	The system returns an error indicating it is above the maximum

			limit.
	Invalid: Request a fourth loan when three are already active	Verify that the system enforces a maximum of 3 active loans	The system returns an error stating the maximum number of loans has been reached.
	Invalid: Apply for a loan with a non-0.25% increment interest rate	Verify that interest rates must be set in 0.25% increments	The system returns an error stating that the interest rate must be in increments of 0.25%.
Loan Payment	Valid: Make a loan payment that covers the minimum amount	Verify that the system accepts payments meeting the minimum requirement	The payment is accepted and reduces the loan balance.
	Invalid: Miss the minimum payment deadline	Verify that a late fee is correctly applied when the minimum payment is not made	The system adds a $50 late fee to the loan balance.
	Invalid: Make a payment that exceeds the remaining loan balance	Verify that payments cannot exceed the loan's outstanding balance	The system rejects the payment as it is more than the outstanding balance.
	Valid: Pay off the remaining loan balance exactly	Verify that the loan account closes upon full payment	The loan is closed and marked as fully paid.
	Valid: Apply for a new loan after closing a previous one	Verify that a new loan can be initiated after closing an existing one	The system approves the new loan as the customer is now eligible again.
Savings Account	Valid: Deposit a positive amount into a savings account	Verify that deposits are correctly added to the savings account	The deposit is successfully added to the savings account.
	Invalid: Deposit a negative amount	Verify that the system rejects negative deposits	The system returns an error indicating deposits must be positive.
	Valid: Withdraw a valid amount from a savings account	Verify that withdrawals are correctly deducted if sufficient funds are available	The withdrawal is successful if sufficient funds are available.

	Invalid: Withdraw an amount that exceeds the balance	Verify that the system rejects withdrawals that exceed the current balance	The system returns an error indicating insufficient funds.
	Valid: Check savings rate as ¼ of the loan rate	Verify that the savings rate is ¼ of the inputted loan rate	The system calculates the correct savings interest rate.
	Valid: Ensure no fractional cents in balances	Verify that account balances are rounded correctly to avoid fractions of cents	Balances are rounded correctly, avoiding fractional cents.
	Invalid: Check if a balance shows a fractional cent	Verify that the system prevents fractional cents in the balance	The system adjusts the balance to remove any fractions of a cent.

Timeline

Task 1: Look over SRS and understand requirements.

- **Starts**: 11/11/2024
- **Precedes**: Testing plan

Task 2: Create Testing Plan

- **Starts**: 11/20/2024
- **Precedes**: Manual testing

Task 3: Start Manual testing

- **Starts**: 11/26/2024
- **Precedes**: Finalization

Task 4: Finalize testing & prepare documentation

- **Starts**: 12/02/2024
- **Ends**: 12/05/2024

Contingency Buffer

- **Additional Time**: A buffer of **2–5 days** can be allocated throughout the testing process to accommodate any unforeseen delays or issues.

- **Reevaluation**: At the end of each week, assess the progress against the planned schedule. If critical delays are anticipated, reallocate tasks or adjust the timeline accordingly.

Risks

Several risks could affect the testing process and project timeline:

- **Dependencies:** If the team we are testing does not have a testable program then it is impossible Black box test.
- **Communication:** If the team we are testing does not communicate in a timely fashion it can lead to delays and resentment.
- **Resource Availability:** Team members may have conflicting commitments, impacting testing coverage.
- **Complexity of the System:** The intricate banking system could lead to misunderstanding of the requirements.

Testing Log

Test Cases

1. Account Management

Test Case 1.1

Verify that the system generates a unique account number for each new customer
Input: Create an account for 2 different customers.
Expected Outcome: The account numbers don't match.

Test Case 1.2

Verify a customer has exactly one savings account at a time

Input: Attempt to create a 2nd savings account.
Expected Outcome: A 2nd savings account is not created.

---

2. Monthly Transactions

Test Case 2.1

Verify that the customer can only initiate a loan up to the proper number of times in a month.
Input: Attempt to initiate a loan more than the proper number of times in a month.
Expected Outcome: The system won't allow this action.

Test Case 2.2

Verify that the customer can only make a payment on a loan up to the proper number of times in a month.
Input: Attempt to make a payment on a loan more than the proper number of times in a month.
Expected Outcome: The system won't allow this action.

Test Case 2.3

Verify that the customer can only deposit to their savings account up to the proper number of times in a month
Input: Attempt to deposit to your savings account more than the proper number of times in a month.
Expected Outcome: The system won't allow this action.

Test Case 2.4

Verify that the customer can only withdraw from their savings account up to the proper number of times in a month
Input: Attempt to withdraw from your savings account more than the proper number of times in a month.
Expected Outcome: The system won't allow this action.

---

3. Monthly Change

Test Case 3.1

Verify that interest is calculated on loans and added to the loan balance
Input: Take out a loan and don't make a payment on it for a month.
Expected Outcome: The amount owed on the loan for the next month matches what is expected for added interest.

Test Case 3.2

Verify that interest is calculated on savings and added to the savings balance
Input: Have some money in savings at the end of a month.
Expected Outcome: The amount in savings for the next month matches what is expected for added interest.

Test Case 3.3

Verify that the bank statement shows beginning balances, activity, interest, and ending balances in a way that the user can understand
Input: Call up the bank statement.
Expected Outcome: The bank statement shows all necessary information.

---

4. Initiate a Loan

Test Case 4.1

Verify the minimum loan amount is $500
Input: Attempt to take out a loan with less than $500.
Expected Outcome: You are unable to take out the loan.

Test Case 4.2

Verify the maximum loan amount is $50,000
Input: Attempt to take out a loan with more than $50,000.
Expected Outcome: You are unable to take out the loan.

Test Case 4.3

Verify that the user can't take out more than 3 loans
Input: Attempt to take out a 4th loan.
Expected Outcome: You are unable to take out the loan.

Test Case 4.4

Verify that the interest rates have an increment of 0.25%
Input: Don't make a payment on a loan for a year and then call up the bank statement.
Expected Outcome: The interest rate for the loan in the next year has gone up by 0.25%.

Test Case 4.5

Verify that the interest rate matches the inputted rate
Input: Call up the bank statement.
Expected Outcome: The interest rate on the bank statement matches the inputted rate.

---

5. Loan Payment

Test Case 5.1

Verify that the late fee of $50 is added to the loan balance if the minimum payment is not made
Input: Don't pay the minimum on the loan and let the next month pass.
Expected Outcome: The bank statement shows that a $50 late fee was added to the loan balance.

Test Case 5.2

Verify that the minimum payment listed in the bill is correct
Input: Initiate a loan and call up the bank statement.
Expected Outcome: The minimum payment listed in the bank statement is correct for the money in the loan.

Test Case 5.3

Verify that a payment can’t be higher than the loan balance
Input: Attempt to pay more money on a loan than is in the loan balance.
Expected Outcome: The system won’t allow action.

Test Case 5.4

Verify that the loan account closes when the balance is paid
Input: Pay off a loan.
Expected Outcome: The loan account closes.

Test Case 5.5

Verify that a new loan can be made once the loan account is closed if the max number of loans was reached previously
Input: Have 3 loans, pay one off and attempt to initiate a new loan.
Expected Outcome: The loan initiation is successful.

---

6. Saving Account

Test Case 6.1

Verify that you can deposit money into your savings account
Input: Attempt to deposit money into savings.
Expected Outcome: The money deposited shows up in savings in the bank statement.

Test Case 6.2

Verify that you can withdraw money from your savings account
Input: Attempt to withdraw money from savings.
Expected Outcome: The money withdrawn doesn’t show up in savings in the bank statement.

Test Case 6.3

Verify that the savings rate is ¼ of the inputting loan rate
Input: Call up the bank statement.
Expected Outcome: The savings interest rate is ¼ of the loan rate on the bank statement.

Test Case 6.4

Verify that accounts do not have fractions of cents and rounding is applied
Input: Attempt to deposit money into savings with fractions of cents.
Expected Outcome: The account doesn’t show fractions of cents and rounding is applied.

Testing Records

1. Account Management

- **Test Case 1.1**: Verify that the system generates a unique account number for each new customer.
  Verified: Failed (12/04/2024)
- **Test Case 1.2**: Verify a customer has exactly one savings account at a time.
  Verified: Passed (12/04/2024)

2. Monthly Transactions

- **Test Case 2.1**: Verify that the customer can only initiate a loan up to the proper number of times in a month.
  Verified: Failed (12/04/2024)
- **Test Case 2.2**: Verify that the customer can only make a payment on a loan up to the proper number of times in a month.
  Verified: Failed (12/04/2024)
- **Test Case 2.3**: Verify that the customer can only deposit to their savings account up to the proper number of times in a month.
  Verified: Failed (12/04/2024)
- **Test Case 2.4**: Verify that the customer can only withdraw from their savings account up to the proper number of times in a month.
  Verified: Failed (12/04/2024)

3. Monthly Change

- **Test Case 3.1**: Verify that interest is calculated on loans and added to the loan balance.
  Verified: Passed (12/04/2024)
- **Test Case 3.2**: Verify that interest is calculated on savings and added to the savings balance.
  Verified: Passed (12/04/2024)
- **Test Case 3.3**: Verify that the bank statement shows beginning balances, activity, interest, and ending balances in a way that the user can understand.
  Verified: Failed (12/04/2024)

4. Initiate a Loan

- **Test Case 4.1**: Verify the minimum loan amount is $500.
  Verified: Passed (12/04/2024)
- **Test Case 4.2**: Verify the maximum loan amount is $50,000.
  Verified: Passed (12/04/2024)
- **Test Case 4.3**: Verify that the user can’t take out more than 3 loans.
  Verified: Failed (12/04/2024)
- **Test Case 4.4**: Verify that the interest rates have an increment of 0.25%.
  Verified: Can’t test (12/04/2024)
- **Test Case 4.5**: Verify that the interest rate matches the inputted rate.
  Verified: Passed (12/04/2024)

5. Loan Payment

- **Test Case 5.1**: Verify that the late fee of $50 is added to the loan balance if the minimum payment is not made.
  Verified: Failed (12/04/2024)
- **Test Case 5.2**: Verify that the minimum payment listed in the bill is correct.
  Verified: Failed (12/04/2024)
- **Test Case 5.3**: Verify that a payment can’t be higher than the loan balance.
  Verified:Can’t test (12/04/2024)
- **Test Case 5.4**: Verify that the loan account closes when the balance is paid.
  Verified: Can’t test (12/04/2024)
- **Test Case 5.5**: Verify that a new loan can be made once the loan account is closed if the max number of loans was reached previously.
  Verified: Can’t test (12/04/2024)

6. Saving Account

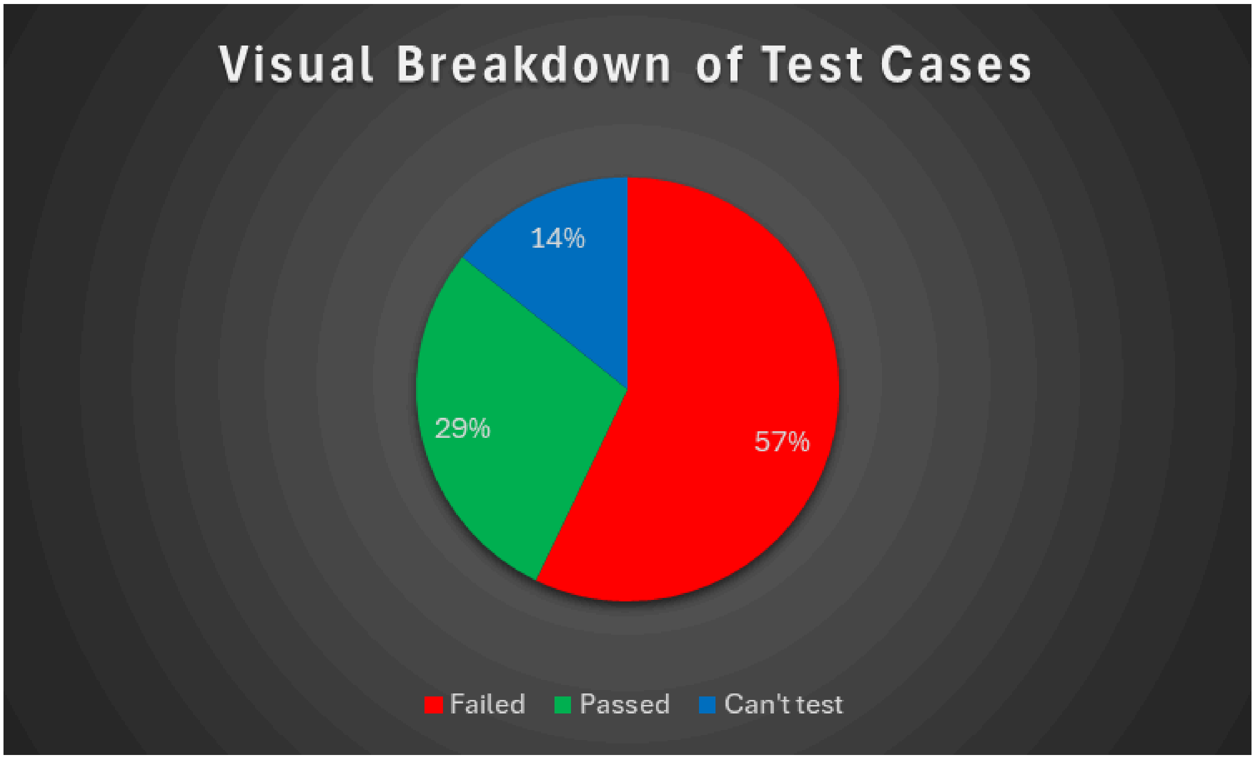
- **Test Case 6.1**: Verify that you can deposit money into your savings account.
  Verified: Failed (12/04/2024)
- **Test Case 6.2**: Verify that you can withdraw money from your savings account.
  Verified: Failed (12/04/2024)
- **Test Case 6.3**: Verify that the savings rate is ¼ of the inputting loan rate.
  Verified: Passed (12/04/2024)
- **Test Case 6.4**: Verify that accounts do not have fractions of cents and rounding is applied.
  Verified: Failed (12/04/2024)

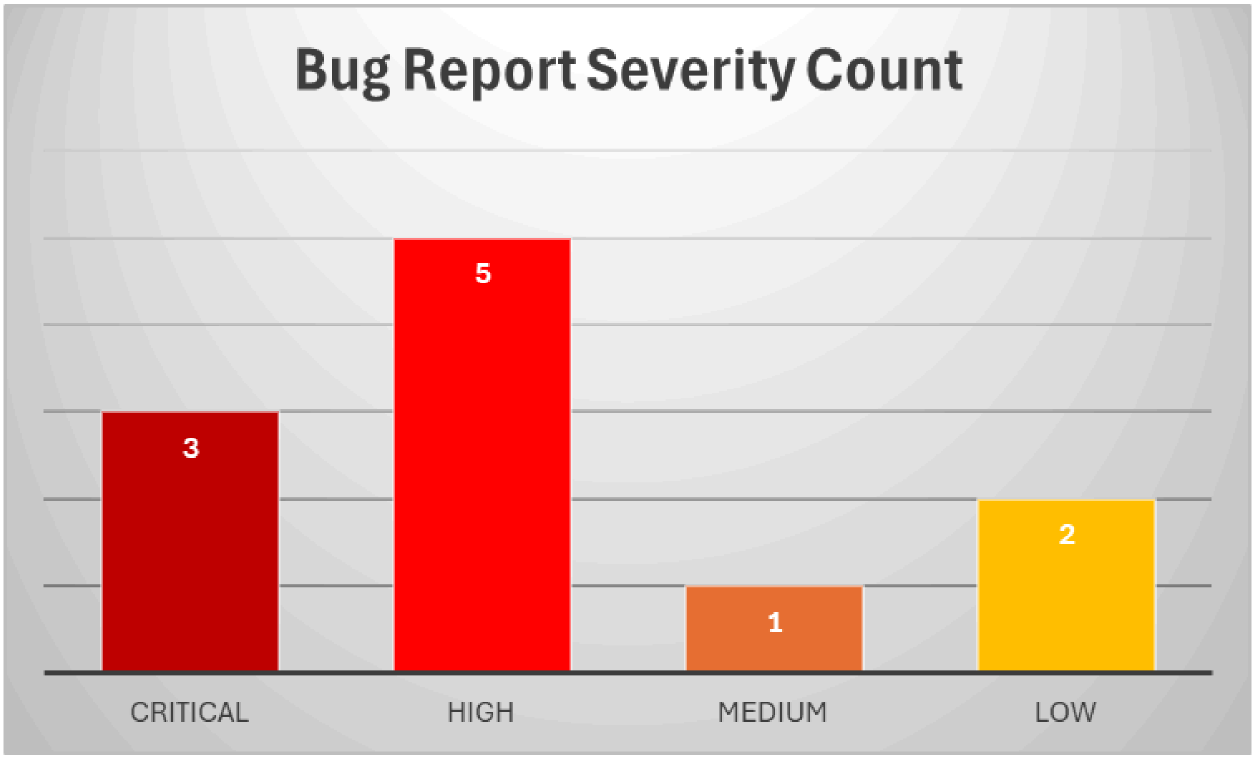
Bug Report

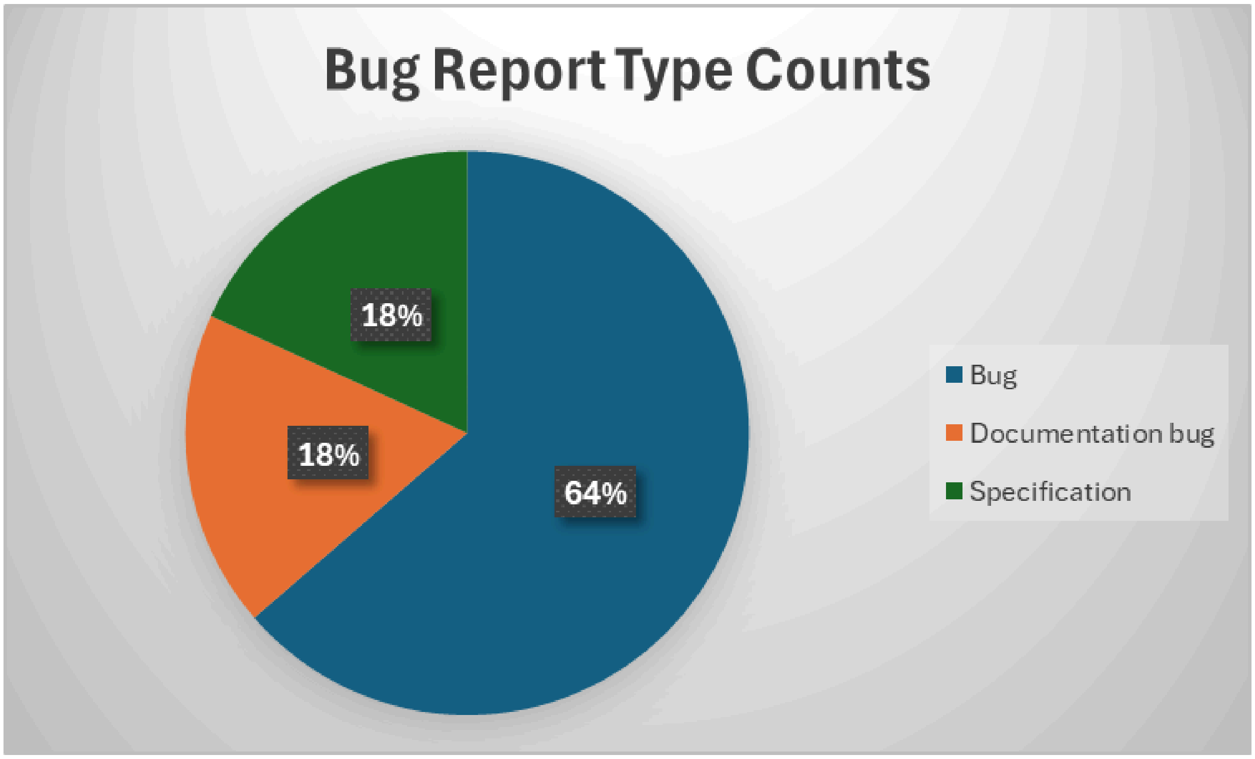
Type	Severity	Description
Bug	High	No user input option.
Documentation bug	High	README fails to state compiler version of program.
Specification	Low	Failed to specify number of times actions can be completed in a month.
Specification	Low	Failed to specify method of rounding.
Documentation bug	Low	README includes unnecessary code definitions.
Bug	Medium	Inclusion of extra loan interest rates (more than one loan interest rate).
Bug	Critical	Failure to include loan payments.

Bug	Critical	Failure to include the ability to withdraw from savings.
Bug	Critical	Failure to include the ability to deposit to savings.
Bug	High	Incorrect loan calculation: A newly created loan accrues interest for the first month.
Bug	High	Ambiguous time passage: Loans are calculated for two months while only a single month is calculated for a savings account.
Bug	High	Bank statement is not very easily readable and lacks context.
Bug	High	Failure to include a system for monthly time passage for more than 2 months.

Metrics and Graphs




Bug Report Severity Count
3
5
1
2
CRITICAL
HIGH
MEDIUM
LOW


Bug Report Type Counts
18%
18%
64%
Bug
Documentation bug
Specification