Angel Gomez

CS 420

Professor Yu

11/12/14

Car Loan Interest Rate Tool

1.Preliminary User Stories

* I would want to get personalized interest rates based on my credit score
* I will also want the system to tell me if I have any positive or negative equity on my current car
* I want to compare different loan terms so that I can choose the best option.
* I want to view what banks or credit unions I am eligible for

2. Functional Requirements Based on User Stories

* The system would be able to allow the user to input their price range and term range.
* Gather interest rate for each individual customer
* System will show if a customer has had late payments and bad reputation
* System should be able to display the comparison of interest depending on the lender
* System can give you personalized advice on how to improve credit.

3. Preliminary Use Case

Use Case 1: Calculate Loan Payment

Actors: User, System

Precondition: User has entered all necessary inputs.

Series of events

User inputs car price, down payment, loan tenure, and credit score.

The system fetches the interest rate based on the credit score.

System calculates loan amount and monthly payments.

System displays the breakdown of payments to the user.

Postcondition: User sees the detailed loan calculations.

Use Case 2: Compare Loan Options

Actors: User, System

Precondition: User has completed at least one loan calculation.

Series of events

User clicks on "Compare Loan Options."

System retrieves interest rates for various banks or lenders.

System displays a comparison table with options.

Postcondition: User views the comparison and selects the best loan option.

4. Preliminary Nonfunctional Requirements

* Layout should be clear and accessible to the customer 24/7
* System should be able to accurately display the calculations
* System should display between what credit cores have the chance of acceptance

-----------------------------------------------------------------------------------------

**Architectural Design**

Will consist of 3-tier architecture

This will have a UI that is accessible to everyone including the users, based on their needs.

It will also have an application layer that will consist of input, it will perform calculations and fetch the interest rates that are needed for the loans.

Will be built using python but not limited to Java or other languages.

Updated Use Cases

Addition to Use Case 1

* User will open the app and navigate to the calculator page
* Users will input car price, down payment and loan tenure Aswell as credit score.
* System retrieves the corresponding interest rates based on their credit scores
* System will calculate all the factors and display the results to the user

Addition to Use Case 2

* Options to help the user compare loans and interest rates as well as show them the difference if they decide to have a co-signer.
* System will also display approval percentages from each union for better application judgement
* Users will select the most suitable option, and the system will help with moving forward.

**Sequence Diagram for calculating loan payments**

User --> UI: Inputs car price, down payment, loan tenure, and credit score.

UI --> Backend: Sends input data.

Backend --> Database: Fetches interest rate based on credit score.

Backend: Calculates loan amount, monthly payment, and total payment.

Backend --> UI: Sends calculated results.

UI --> User: Displays loan calculations.

**Sequence Diagram for comparing loans**

User --> UI: Clicks on "Compare Options."

UI --> Backend: Requests lender rates.

Backend --> Database: Retrieves interest rates for multiple lenders.

Backend --> UI: Sends comparison data.

UI --> User: Displays lender comparison table.

Modified Nonfictional Requirements

Performance:

The system shall process loan calculations within 500ms for up to 100 concurrent users.

Comparison queries should be completed in under 2 seconds, even with 1,000 lender entries.

Usability:

The system should feature a mobile-friendly UI for use on various devices.

Users will be able to complete a loan calculation in less than 5 steps.

Scalability:

The system shall support up to 10,000 concurrent users with minimal response delays.

Security:

User inputs (credit score) shall be securely transmitted using TLS encryption.

Backend APIs will restrict access to authorized clients only.

Reliability:

System uptime should be 99.95%, with a failover mechanism in case of backend failures.

Maintainability:

Loan formulas and interest rate tables should be easily updated in the database.

Class Diagram

**Describe the User Interface**

The interface will be created with a clean layout that is easy to understand and navigate through.

Compatibility will be across both desktop and mobile devices

The user interface will be vivid and colorful and easy on the eye for those who are navigating through it.

**Key Components**

Home Screen: Introduction to the tool and quick navigation to the loan calculator.

Loan Calculator Screen: Input form for car price, down payment, loan tenure, and credit score.

Results Screen: Displays the loan amount, monthly payment, total payment, and a breakdown of payments.

Comparison Screen: A table showing interest rates, monthly payments, and total payments for various lenders.

Settings and Help: Optional screen for adjusting preferences or viewing FAQs.

--------------------

**User interface Screens**

**Screen 1: Home Screen**

Purpose: Entry point for the tool.

Key Elements:

* Title: "Car Loan Interest Rate Tool"
* Description: Brief text explaining the tool's purpose.
* Button: “Start Loan Calculation.”

**Screen 2: Loan Calculator Screen**

Purpose: Collect user inputs for loan calculation.

Key Elements:

Input Fields:

* Car Price (with currency symbol)
* Down Payment (with percentage option)
* Loan Tenure (dropdown: 12, 24, 36, 48 months, etc.)
* Credit Score (slider or numeric input, range: 300-850)

Action Buttons:

“Calculate Loan”

“Clear Form”

Tooltips/Help: Small icons with explanations for each input field.

**Screen 3: Results Screen**

Purpose: Display loan details and breakdown.

Key Elements:

Loan Amount: Bold and prominent.

Monthly Payment: Bold and highlighted.

Total Payment: Below the monthly payment.

Breakdown Table:

* Columns: “Principal,” “Interest,” “Total.”
* Button: “Compare Loan Options.”

**Screen 4: Comparison Screen**

Purpose: Show interest rates and payment comparisons for multiple lenders.

Key Elements:

Table Layout:

* Columns: “Lender Name,” “Interest Rate,” “Monthly Payment,” “Total Payment.”
* Filters/Sort Options:
* Filter by loan tenure.
* Sort by lowest interest rate or total payment.

Button: “Back to Results.”

**Screen 5: Settings and Help Screen**

Purpose: Provide additional options and FAQs.

Key Elements:

Sections:

* Settings: Toggle for currency type (e.g., USD, EUR).
* FAQs: Common questions about loan calculation and tool usage.

Button: “Return to Calculator.”

**Navigation Between Screens**

Home Screen → Loan Calculator Screen:

* User clicks “Start Loan Calculation” on the home screen to navigate to the calculator.

Loan Calculator Screen → Results Screen:

* After entering inputs and clicking “Calculate Loan,” the results are displayed on the results screen.
* Results Screen → Comparison Screen:
* User clicks “Compare Loan Options” to navigate to the lender comparison table.

Comparison Screen → Results Screen:

* User clicks “Back to Results” to return to the calculated loan results.

Any Screen → Settings and Help Screen:

* User clicks a setting or help icon available in the navigation bar.
* Settings and Help Screen → Previous Screen:
* User clicks “Return to Calculator” to go back to the loan calculator or other screens.

When it comes to security, we refer to the protection of the system itself. This includes the data that surrounds it and the supervision of who has authorized access without causing any disruptions.

Security Problems

Unauthorized Access to sensitive data such as a customer's credit score and personal information, the solution or this could be using multifactor authentications like Authy.

Injection Attacks

Malicious inputs that are given to the system to execute certain task to exploit it or manipulate what the system holds, this can be solved by validating all the inputs to comfort all of the formats, also using parameterized queries or prepared statements in databases to prevent SQL injections.

Insider Threats

This is usually due to giving too much access to all of the users therefore exploiting the private data with malicious intent and misuse, the simple solution for this issue would be to restrict and have a role-base access control to limit user to only necessary data that they would need to do their job.

Phishing or Scam

There are times where attackers might send out fake emails or text messages to try and manipulate the users into sending money or revealing their personal information or credentials, the solution to this would be to implement an email filtering tool that will pick up on patterns and block these phishing attempts.

Reference

<https://www.nist.gov/cyberframework>

[https://owasp.org](https://owasp.org/)