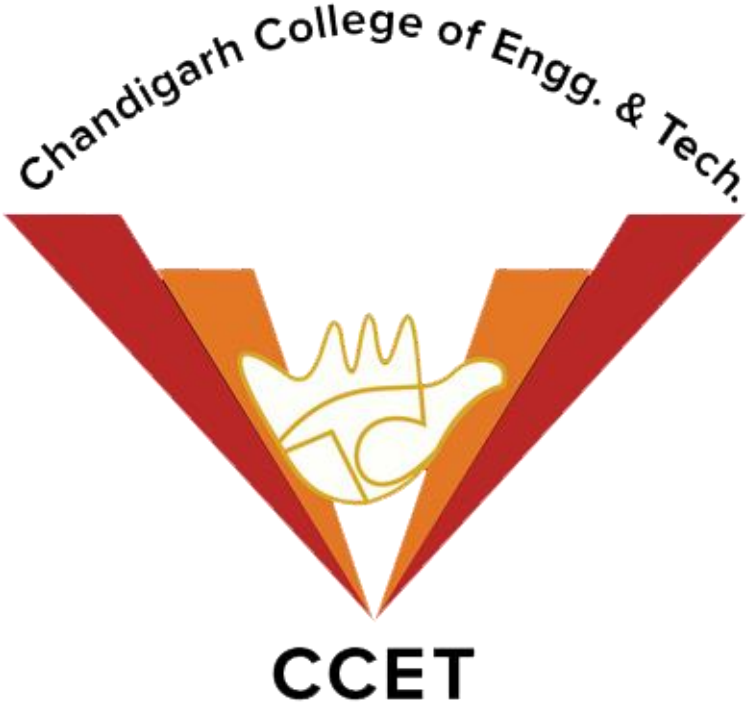
MAJOR PROJECT SRS(Software Requirements Specification)



Submitted To:-

Dr. Santosh Kumar Yadav Lecturer

CSE DEPARTMENT

Major Project SRS

Submitted By:-

Rohit Singh

6TH SEMESTER

9533/23(leet)

Table of Contents

6. **Performance Metrics**
   - 6.1. Response Time
   - 6.2. Memory Usage
   - 6.3. CPU Utilization
   - 6.4. Battery Consumption
   - 6.5. Storage Requirements
   - 6.6. Network Performance (for Online Features)
7. **Testing & Validation**
   - 7.1. Unit Testing
   - 7.2. Integration Testing
   - 7.3. Performance Testing
   - 7.4. Usability Testing
   - 7.5. Security Testing
   - 7.6. Validation Testing
8. **Future Enhancements**
   - 8.1. Advanced Graphical Calculator
   - 8.2. Voice Command Integration
   - 8.3. Cloud Storage for History
   - 8.4. AI-Based Smart Suggestions
   - 8.5. Multi-Language Support
   - 8.6. Dark Mode & Custom Themes
9. **User Interface Design**
   - 9.1. Home Screen
   - 9.2. Standard Calculator Interface
   - 9.3. EMI Calculator Interface
   - 9.4. Currency Converter UI
   - 9.5. Stock Profit Calculator UI
   - 9.6. Calculation History UI
   - 9.7. Unit Converter
10. **Risks and Mitigation**
    - 10.1. Potential Risks
    - 10.2. Mitigation Strategies
11. **Conclusion**

1. Introduction:-

1.1. Purpose

The purpose of this document is to define the functional and non-functional requirements for the Calculator App developed using Android Studio with Java. The application provides various calculation functionalities such as basic arithmetic operations, an EMI calculator, currency conversion, unit converter, factorial, square, cube calculations, a Stock Profit Calculator with support for multiple brokers, and a History Feature to store past calculations.

This document serves as a guide for developers, testers, and stakeholders to understand the app's purpose, design, and requirements.

1.2. Scope

The Calculator App is designed to be a comprehensive mathematical and financial tool with a simple and user-friendly interface. The key features of the app include:

- **Basic Arithmetic Calculations**: Addition, Subtraction, Multiplication, and Division.
- **EMI Calculator**: Helps users calculate monthly EMI payments based on loan amount, interest rate, and tenure.
- **Currency Converter**: Converts values between different currencies using real-time exchange rates.
- **Unit converter:** converting units (Length, weight, Temperature, Area, Volume, Speed, and Time)
- **Stock Profit Calculator**: Allows stock traders to compute profit/loss, considering brokerage fees and taxes.
- **History Feature**: Stores previous calculations for easy retrieval and reference.
- **Detailed Result Display**: Provides a breakdown of key financial figures such as buy price, sell price, quantity, brokerage, and net profit.
- Factorial, Square, Cube Calculations: Useful for academic and professional purposes.
- User-Friendly Interface: Simple navigation with a clean UI for an enhanced user experience.

The application is designed to be fast, lightweight, and efficient, ensuring smooth performance even on devices with limited resources. It will support both offline and online functionality, with real-time data fetching only for currency conversion.

1.3. Definitions

- EMI: Equated Monthly Installment
- F&O: Futures and Options
- GST: Goods and Services Tax
- UI: User Interface
- API: Application Programming Interface
- SQLite: Lightweight database for storing application data locally
- Stock Trading: Buying and selling of financial assets such as shares.
- Brokerage: A fee charged by brokers for executing a stock market trade.

1.4. References

- Android Developer Documentation (developer.android.com)
- Java Documentation (docs.oracle.com)
- Google Play Store Guidelines (play.google.com)
- Financial Calculations Reference (Investopedia, NSE India)
- SQLite Documentation (sqlite.org)

2. Overall Description:-

2.1. Product Perspective

The Calculator App is a standalone mobile application that does not require an internet connection for most calculations. However, certain features, such as the currency converter, will rely on real-time exchange rate data fetched via an API. The history feature ensures that users can retrieve their past calculations at any time, making the app useful for long-term financial planning.

2.2. Product Functions

The major functionalities include:

- Basic Calculator: Simple mathematical operations.
- Financial Calculations: EMI and stock profit calculations.

- Unit converter: converting units (Length, weight, Temperature, Area, Volume, Speed, and time)
- History Feature: Storing and retrieving past calculations.
- Detailed Result Display: Breakdown of calculations, including brokerage and taxes.
- User-friendly Interface: Simple, intuitive navigation.

2.3. User Characteristics

The primary users of this app include:

- General users: People who need a quick and easy-to-use calculator.
- Students: Users who require advanced calculations such as factorial, square, and cube.
- Financial professionals: Individuals managing finances and loans.
- Stock traders: Investors analyzing profit/loss scenarios before making decisions.
- Business Owners: Small business owners who need basic financial computations.

2.4. Constraints

- Internet Requirement: Currency conversion requires an internet connection.
- Storage Limitation: Limited storage for history due to device constraints.
- Accuracy Dependency: Stock profit calculations depend on accurate brokerage and GST data.
- Device Compatibility: The application should function efficiently on Android devices with minimal RAM.

3. Specific Requirements:-

3.1. Functional Requirements

3.1.1. Basic Calculator

- Supports standard arithmetic operations: addition, subtraction, multiplication, and division.
- Provides instant and accurate results.
- Allows decimal values and large numbers.
- Displays an error message for invalid inputs (e.g., division by zero).

- Supports both portrait and landscape orientations.

3.1.2. EMI Calculator

- Accepts user input for loan amount, interest rate, and loan tenure.
- Calculates monthly EMI, total payable interest, and total amount to be paid.
- Provides graphical representation of EMI breakdown.
- Allows users to compare multiple EMI plans.
- Displays amortization schedule for detailed analysis.

3.1.3. Currency Converter

- Fetches real-time exchange rates from an online API.
- Allows conversion between multiple international currencies.
- Displays currency conversion history.
- Enables offline conversion using the last fetched rates.
- Provides user-friendly interface for selecting currencies.

3.1.4. Stock Profit Calculator

- Users input buy price, sell price, quantity, and brokerage fees.
- Automatically calculates GST and other government charges.
- Computes net profit/loss with a detailed breakdown.
- Supports different stock brokers with predefined brokerage structures.
- Allows users to choose between Intraday, Delivery Equity, F&O Futures, and F&O Options.

3.1.5. History Feature

- Automatically saves all performed calculations.
- Allows users to search, filter, and delete previous calculations.
- Stores calculations persistently using SQLite database.
- Provides an option to export history as a text or CSV file.
- Enables users to mark important calculations as favorites.

3.1.6. Unit Converter

- Supports conversion between different units of measurement (e.g., length, weight, temperature, volume, speed, etc.).
- Provides accurate and real-time unit conversions.
- Allows users to select units from an intuitive drop-down menu.
- Stores frequently used conversions for quick access.
- Supports both metric and imperial systems.

3.2. Non-Functional Requirements

- Performance: All calculations should be performed within milliseconds.
- Security: No sensitive user data should be stored or shared.
- Usability: The app should have an intuitive interface with a smooth user experience.
- Scalability: The architecture should support future enhancements and additional features.
- Reliability: The app should function correctly under all standard usage scenarios.
- Portability: The app should be compatible with Android versions 6.0 and above.
- Accessibility: The app should support large text mode and high contrast mode for better readability.

4. System Architecture:-

4.1. Overview

The Calculator App follows a three-layered architecture, ensuring modularity and maintainability. The three primary layers include:

- Presentation Layer (UI Layer): Manages user interactions and displays results.
- Application Logic Layer (Business Logic): Processes calculations and logic for different modules.
- Data Storage Layer: Stores history, preferences, and cached data.

4.2. Components

4.2.1. User Interface (UI Layer)

- Activity & Fragment-based UI using XML layouts.
- Material Design Principles for an intuitive user experience.
- Toolbar with Navigation Drawer for feature access.

4.2.2. Business Logic Layer

- Java-based classes for handling calculations.
- Separate modules for arithmetic, EMI, currency conversion, and stock profit calculations.
- Real-time API integration for currency exchange rates.

4.2.3. Data Storage Layer

- SQLite Database for storing history and user preferences.
- SharedPreferences for lightweight data storage.
- Local Caching for improved performance.

4.3. Data Flow

1. User inputs values via UI.
2. Processing in Business Logic Layer based on selected operation.
3. Results stored in SQLite (if applicable).
4. Results displayed to the user.

5. Use Case Scenarios:-

5.1. Use Case 1: Performing Basic Arithmetic Calculation

Actors: User

Description: The user opens the calculator app and performs an arithmetic operation (addition, subtraction, multiplication, or division).

Flow of Events:

1. User launches the app.

2. User enters two numbers.
3. User selects an arithmetic operation.
4. The result is displayed on the screen.

5.2. Use Case 2: Calculating EMI

Actors: User

Description: The user calculates EMI based on loan amount, interest rate, and tenure.

Flow of Events:

1. User selects EMI calculator.
2. User enters loan amount, interest rate, and tenure.
3. User taps the “Calculate” button.
4. The monthly EMI amount is displayed.

5.3. Use Case 3: Storing and Retrieving Calculation History

Actors: User

Description: The user saves previous calculations and retrieves them later.

Flow of Events:

1. User performs a calculation.
2. The app automatically saves the result.
3. User navigates to the history section.
4. User selects a previous calculation to view its details.

5.4. Use Case 4: Using the Currency Converter

Actors: User

Description: The user converts a value from one currency to another using real-time exchange rates.

Flow of Events:

1. User selects the currency converter.
2. User enters the amount and selects the source and target currencies.
3. User taps the “Convert” button.
4. The converted amount is displayed along with the exchange rate used.

5.5. Use Case 5: Calculating Stock Profit/Loss

Actors: Stock trader, investor

Description: The user calculates stock profit/loss, considering brokerage fees and taxes.

Flow of Events:

1. User selects the stock profit calculator.
2. User enters buy price, sell price, quantity, and broker selection.
3. User selects the type of transaction (Intraday, Delivery, F&O Futures, F&O Options).
4. User taps “Calculate.”
5. The app displays a detailed breakdown of profit/loss, including brokerage, GST, and net gain.

5.6. Use Case 6: Using the Unit Converter

Actors: User

Description: The user converts values between different units of measurement.

Flow of Events:

1. User selects the unit converter.
2. User chooses the measurement category (e.g., length, weight, temperature, volume, speed).
3. User enters a value and selects the source and target units.
4. User taps “Convert.”
5. The c
6. onverted value is displayed.

6. Performance Metrics:-

To ensure the Calculator App meets the required performance standards, the following metrics will be measured:

6.1. Response Time

- The app should process basic arithmetic calculations (addition, subtraction, multiplication, and division) in less than 100 milliseconds.
- Complex calculations like EMI, stock profit, and currency conversion should be completed in under 500 milliseconds.

6.2. Memory Usage

- The application should use less than 50MB of RAM during standard operations.
- The history storage feature should optimize memory usage to prevent excessive storage consumption.

6.3. CPU Utilization

- The app should not exceed 15% CPU usage during complex calculations on mid-range devices.
- Background tasks like auto-saving history should run efficiently without affecting the main operations.

6.4. Battery Consumption

- The app should not consume more than 3% battery per hour during continuous usage.
- Energy-efficient processing techniques should be implemented to reduce unnecessary power consumption.

6.5. Storage Requirements

- The app size should remain under 20MB for installation.
- Cached data, including history and preferences, should not exceed 10MB unless explicitly allowed by the user.

6.6. Network Performance (for Online Features)

- Real-time currency conversion API calls should complete within 2 seconds on a 4G connection.
- The app should provide a fallback mechanism if the internet is unavailable.

7. Testing & Validation:-

7.1. Unit Testing

- Each function, including arithmetic operations, EMI calculations, and stock profit calculations, will be tested individually.
- Automated unit tests will validate input handling and expected output accuracy.

7.2. Integration Testing

- Ensures that all modules (basic calculations, history storage, currency conversion, etc.) work together without conflicts.
- Validates data flow between components and verifies seamless transitions.

7.3. Performance Testing

- Load testing to simulate multiple calculations at once and ensure stable performance under heavy usage.
- Response time testing to ensure calculations execute within the defined performance metrics.

7.4. Usability Testing

- Conducted with real users to evaluate ease of use, accessibility, and overall user experience.
- Ensures intuitive navigation and user-friendly interface.

7.5. Security Testing

- Tests data encryption for stored history and calculations.
- Ensures API security for currency conversion and online features.

7.6. Validation Testing

- Confirms that the app meets all specified requirements in the SRS document.
- Compares actual results with expected outcomes to ensure correctness and compliance.

8. Future Enhancements:-

8.1. Advanced Graphical Calculator

- Introduce a graphical calculator feature to plot equations and visualize mathematical functions.
- Allow users to analyze complex mathematical problems with graph representation.

8.2. Voice Command Integration

- Implement voice recognition to allow users to perform calculations hands-free.
- Improve accessibility for users with disabilities.

8.3. Cloud Storage for History

- Enable users to back up and restore their calculation history using cloud storage.
- Provide synchronization across multiple devices.

8.4. AI-Based Smart Suggestions

- Introduce AI-based recommendations for financial calculations, such as loan suggestions based on income and expenses.
- Provide automated insights for stock trading calculations.

8.5. Multi-Language Support

- Expand language support to cater to a global audience.
- Allow users to switch between languages seamlessly.

8.6. Dark Mode & Custom Themes

- Add a dark mode for reduced eye strain during nighttime use.
- Allow users to customize the app theme to their preference.

9.User Interface Design:-

The user interface (UI) of the Calculator App is designed to be intuitive, visually appealing, and easy to use for both beginner and advanced users. The UI follows material design principles to ensure consistency and usability.

9.1. Home Screen

- Displays a simple calculator interface with buttons for basic arithmetic operations.
- Includes a navigation menu to switch between different modes (Standard Calculator, EMI Calculator, Currency Converter, Stock Profit Calculator, Unit Converter, etc.).
- Provides a settings icon to allow users to customize preferences such as theme, language, and default calculation mode.

9.2. Standard Calculator Interface

- A numeric keypad with buttons for numbers 0-9.
- Operators (+, -, *, /) placed for easy accessibility.
- Additional functions such as clear (C), backspace, and equals (=) buttons.
- A display area at the top showing the entered numbers and results.

9.3. EMI Calculator Interface

- Three input fields: Loan Amount, Interest Rate, and Loan Tenure.
- A "Calculate" button that processes the EMI result instantly.
- Displays monthly EMI, total interest payable, and total payment.
- Option to save the calculation in history for future reference.

9.4. Currency Converter UI

- Two dropdown menus for selecting source and target currencies.
- An input field to enter the amount for conversion.

- A "Convert" button that retrieves real-time exchange rates and displays the converted amount.
- A refresh button to update exchange rates.

9.5. Stock Profit Calculator UI

- Fields for Buy Price, Sell Price, Quantity, and Brokerage selection.
- A dropdown for selecting trade type (Intraday, Delivery, F&O Futures, F&O Options).
- A "Calculate" button that provides a detailed breakdown of profit/loss, brokerage, taxes, and net earnings.
- A history-saving feature for tracking previous calculations.

9.6. Calculation History UI

- A dedicated screen that lists previous calculations with timestamps.
- Users can click on any entry to view details or re-use past calculations.
- A "Clear History" option to delete stored records.

9.7. Unit Converter

- Allows users to convert between different units such as length, weight, volume, and temperature.
- Includes dropdown menus for selecting unit categories and specific units.
- An input field for entering the value to be converted.
- A "Convert" button that instantly displays the converted result.

The UI design ensures a seamless user experience, making calculations fast, efficient, and visually appealing.

10. Risksb and Mitigation:-

10.1. Potential Risks

- API Downtime: May impact real-time currency conversion.
- Calculation Errors: Ensuring accuracy in financial calculations.
- User Data Loss: Managing history feature efficiently.

10.2. Mitigation Strategies

- Implementing backup APIs for exchange rates.
- Conducting thorough testing for all calculations.
- Allowing users to export history as a file.

11. Conclusion:-

The Calculator App is a comprehensive solution for users who require quick and accurate calculations across various domains, including standard arithmetic, EMI calculations, currency conversion, stock profit analysis, and unit conversions. Designed with user experience in mind, the app ensures accessibility, efficiency, and security through well-structured features and optimizations. With built-in data storage, robust security mechanisms, and real-time updates, users can rely on the app for precise and up-to-date financial computations. By addressing potential risks with effective mitigation strategies, the app guarantees reliability, accuracy, and ease of use. Future enhancements will further improve its functionality, making it an indispensable tool for everyday calculations.