**Exercise 3 – SRS and Use-case diagrams**

**Topic: *Restaurant Employees Tips Giver and Calculator***

# 1. Scope

The scope of this document is to define the comprehensive requirements for the development of a mobile application that assists restaurant employees in calculating tips and provides tips on improving customer service.

# 2. General Description

### a. Target Audience

The target audience for this application includes:

* Restaurant owners and managers
* Restaurant employees, including waitstaff, cooks, and bartenders
* Customers who wish to provide tips electronically

### b. Objectives

The primary objectives of the "Restaurant Employees Tips Giver and Calculator" application are as follows:

* Provide a user-friendly and intuitive interface for restaurant employees to enter bill amounts and calculate tips effortlessly.
* Automate the tip calculation process based on user inputs, ensuring accuracy and efficiency.
* Offer valuable tips and suggestions on improving customer service, customized to each user's needs and performance.
* Store historical tip data for employees, allowing them to track their earnings and identify areas for improvement.

### c. Constraints

**Platform Compatibility :**

Developing the app for both iOS and Android platforms is essential to maximize the user base and ensure a wider reach.

It's crucial to consider the differences between the two platforms, such as design guidelines, screen sizes, and performance optimization, to provide a seamless user experience on both.

Ongoing platform updates and compatibility testing will be necessary to ensure the app remains functional as new versions of iOS and Android are released.

**Budget and Timeline:**

Establishing a clear budget is fundamental to the success of the project. It should cover development costs, marketing, maintenance, and any unforeseen expenses.

Creating a detailed project timeline with milestones and deadlines will help in managing resources efficiently and meeting project goals.

Regular budget and timeline reviews are necessary to identify any deviations from the plan and to make adjustments as needed to keep the project on track.

**Data Privacy:**

Protecting users' data and privacy. The application will strictly adhere to both local and international data privacy regulations. User data protection and obtaining user consent are the top priorities.

# 3. Functional Requirements

### User Registration and Authentication:

**Account Creation:** Employees can create accounts using their email address or phone number, providing essential registration details.

**Secure Authentication:** Secure authentication mechanisms will be implemented, ensuring user data security during access to the application.

### Bill Amount Entry:

**Bill Amount Input:** Users can input the bill amount for a customer's meal with ease.

### Tip Calculation:

**Automatic Tip Calculation:** The application calculates the tip amount automatically based on the entered bill amount and the user-specified tip percentage.

### Performance Analytics:

**Tip Tracking:** The application will track and store tips received by employees, allowing them to access their historical earnings data.

### Customer Service Tips:

**Personalized Tips:** The application provides personalized tips and suggestions for improving customer service based on user performance and feedback.

### Notifications:

**Notification System:** Users will receive notifications for various events, including account activity, earnings updates, and personalized customer service tips.

# Non-Functional Requirements

### Performance:

**Responsiveness:** The application should load quickly and respond promptly to user actions, providing a seamless user experience.

**Tip Calculation Speed:** Response times for tip calculations and analytics should meet acceptable standards, ensuring timely information for users

### Security:

* User data, including earnings and account information, should be securely stored and transmitted.
* Implementing encryption and secure authentication methods.

### Usability:

**User-Friendly Interface:** The application will feature an intuitive and user-friendly interface for easy navigation and data entry.

**User Feedback:** A feedback mechanism will be integrated to collect user input and suggestions for continuous improvement.

### Scalability:

**Scalability Planning:** Scalability is a critical consideration, especially for a mobile application that aims to grow its user base over time.

The application's architecture should be designed to handle increased traffic, data, and user interactions without experiencing performance bottlenecks or downtime.

Plan for data storage and database scalability to ensure that the system can handle larger datasets as the user base expands.

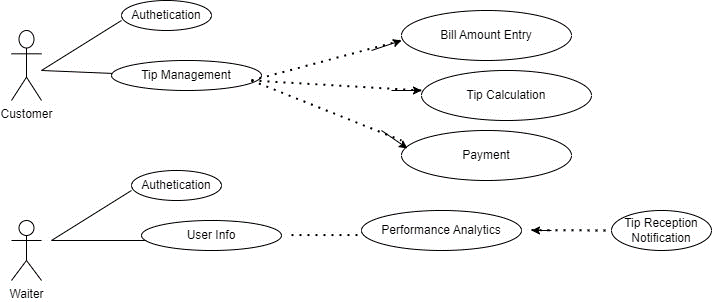
Scalability planning is essential to ensure that the application can handle increased usage without compromising performance, user experience, or reliability. It's a key aspect of long-term success for any software project.

### 

### Privacy:

**Data Privacy Compliance:** Ensure full compliance with data privacy regulations, including obtaining user consent and protecting user data.

# 5. Use Case Models (Diagrams Using UML)



# 6. Appendices

### a. Definitions, Acronyms, Abbreviations

SRS: Software Requirements Specification

UML: Unified Modeling Language

UI: User Interface

UX: User Experience

**API:** Application Programming Interface

SQL: Structured Query Language

**iOS:** Apple's mobile operating system (formerly iPhone OS)

**User Interface (UI):** The visual and interactive elements of an application that users interact with, including screens, buttons, menus, and forms.

**User Experience (UX):** The overall experience and satisfaction of users when interacting with an application, encompassing usability, design, and ease of use.

**Operating System (OS):** System software that manages computer hardware and provides services for computer programs.

**Encryption:** The process of converting data into a coded format to prevent unauthorized access and ensure data security.

**API (Application Programming Interface):** A set of rules and protocols that allow different software applications to communicate and interact with each other.

**SDK (Software Development Kit):** A collection of software tools, libraries, and documentation that aids in the development of software applications for a specific platform or framework.

**GPS (Global Positioning System):** A satellite-based navigation system that provides location and time information anywhere on Earth.

**HTTP (Hypertext Transfer Protocol):** A protocol used for transmitting data and information over the internet, commonly used for web browsing.

**SSL (Secure Sockets Layer):** A cryptographic protocol that ensures secure communication and data exchange over computer networks, commonly used for secure web browsing (HTTPS).

**Data Encryption:** The process of converting data into a code to prevent unauthorized access.

### b. References

**Android Developer Documentation (developer.android.com).**

Reference documentation for Android app development, including guidelines, best practices, and APIs for Android platform development.

**iOS Developer Documentation (developer.apple.com).**

Reference documentation for iOS app development, including guidelines, best practices, and APIs for Apple's iOS platform development.

**IEEE Std 830-1998:** [**IEEE SRS Guidelines**](https://chat.openai.com/c/link)

Guidelines for creating Software Requirements Specifications (SRS) documents.

**OWASP (Open Web Application Security Project) Top Ten.**

A widely recognized reference for web application security issues, helping ensure secure software development practices.