SMS Share Hub

R13944038 Lin Tung-Yi
R13922029 Lin Fang-Chi
R13922165 Lee Meng-Hsueh

Abstract

This project addresses the under-utilization of free SMS quotas provided to users by proposing an application that benefits both businesses and individual users. The system leverages unused SMS quotas by securely matching users with businesses needing SMS services. Through a database, the app assigns users to send encrypted SMS messages on behalf of businesses, ensuring secure and efficient communication. Recipients can decrypt messages into plain text using the app, enhancing both functionality and resource utilization. The proposed system not only optimizes unused SMS quotas but also provides a practical and secure solution for businesses to engage in cost-effective communication.

1 Motivation

Many mobile users receive free SMS quotas as part of their plans, but a significant portion of these quotas often goes unused, leading to wasted resources. On the other hand, the demand for SMS persists despite the prevalence of modern messaging applications. SMS remains essential for scenarios such as service notifications, account verifications, and other time-critical communications.

This project aims to bridge these two needs by proposing a system that matches users with unused SMS quotas to businesses looking for cost-effective ways to send SMS messages. On one side, users can leverage their free SMS quotas to gain incentives such as discounts on registrations or restaurant coupons. On the other side, businesses benefit by accessing an affordable SMS distribution channel. This symbiotic relationship not only reduces wasted resources but also adds value for both parties.

2 System Overview

Our proposed system facilitates the efficient utilization of unused SMS quotas through the following steps:

1. The user books a restaurant via the booking app.

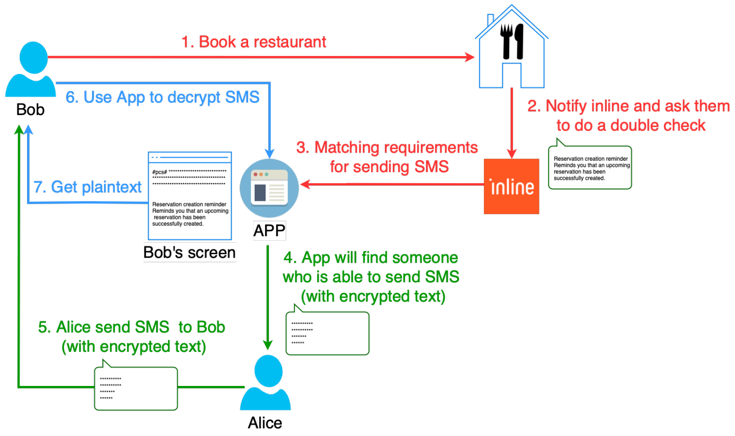

1. Book a restaurant
Bob
6. Use App to decrypt SMS
2. Notify inline and ask them to do a double check
Reservation creation reminder Reminds you that an upcoming reservation has been successfully created.
3. Matching requirements for sending SMS
inline
#pcs#
Reservation creation reminder Reminds you that an upcoming reservation has been successfully created.
7. Get plaintext
APP
Bob's screen
4. App will find someone who is able to send SMS (with encrypted text)
5. Alice send SMS to Bob (with encrypted text)
Alice

Figure 1: System Overview Diagram

2. The booking app notifies the restaurant inline and requests a double-check of the booking details.
3. Our app matches the requirements for sending SMS.
4. Our app identifies a participant with available free SMS quotas to send the message, encrypting the text for privacy.
5. The selected user (e.g., Alice) sends the SMS to the recipient (e.g., Bob) with the encrypted text.
6. The recipient uses our app to decrypt the SMS message.
7. The plaintext message is retrieved and displayed to the recipient on our app.

2.1 Development Environment

IDE: Android Studio Ladybug — 2024.2.1
Emulator Version: Small Phone API 31 (Android 12.0)
Database: SQLite
Programming Language: Java

3 Components

Break down the system into its main components. Describe each component in detail.

3.1 Registration and Login

The registration interface will include fields for username and password, and users will be asked to select their role, which can be either ”shop” or ”message provider.” For message providers, additional information will be required: an SMS quota and a phone number.

3.2 Short message Provider

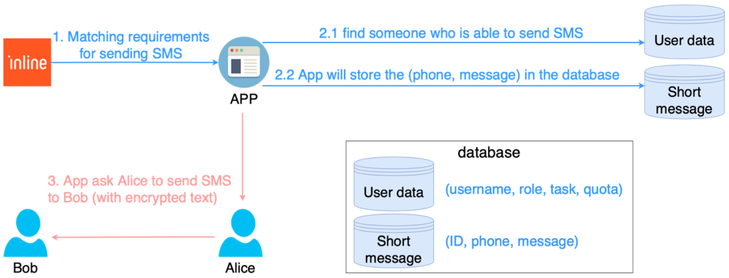

inline
1. Matching requirements for sending SMS
APP
2.1 find someone who is able to send SMS
User data
2.2 App will store the (phone, message) in the database
Short message
3. App ask Alice to send SMS to Bob (with encrypted text)
Bob
Alice
database
User data
(username, role, task, quota)
Short message
(ID, phone, message)

Figure 2: Overall Architecture for Short Message Provider

The overall architecture for the short message provider involves the following steps:

1. Matching the requirements for sending an SMS.
2. Identifying a user who is able to send SMS.
   (a) The app stores the (phone, message) pair in the database.
3. The app requests Alice to send an SMS to Bob with the encrypted text.

3.2.1 Finding the User to Send the Short Message

The process of finding the appropriate user involves the following steps:

1. A shop wants to send a short message to a specific number (e.g., 09000123).
2. The system identifies a user with the same provider as the recipient’s number.
3. The system checks if the user has available SMS quotas.
4. The short message is stored in the short message database.
5. The ”task” column in the database is updated with the short message ID.

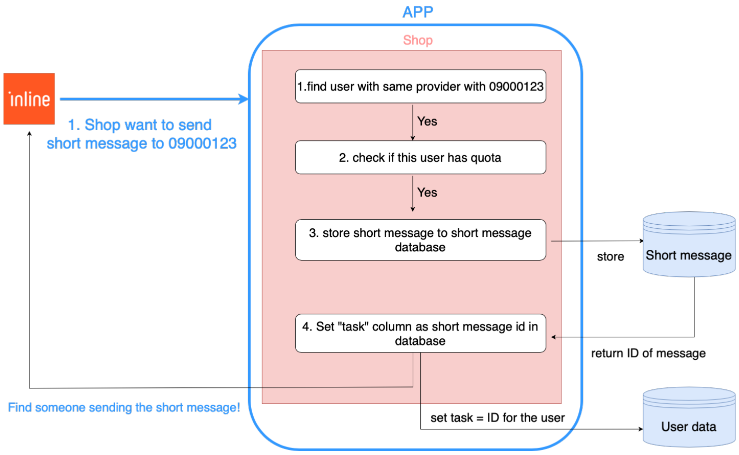

APP
Shop
inline
1. Shop want to send short message to 09000123
1.find user with same provider with 09000123
Yes
2. check if this user has quota
Yes
3. store short message to short message database
store
Short message
4. Set "task" column as short message id in database
return ID of message
Find someone sending the short message!
set task = ID for the user
User data

Figure 3: Finding a User to Send the Short Message

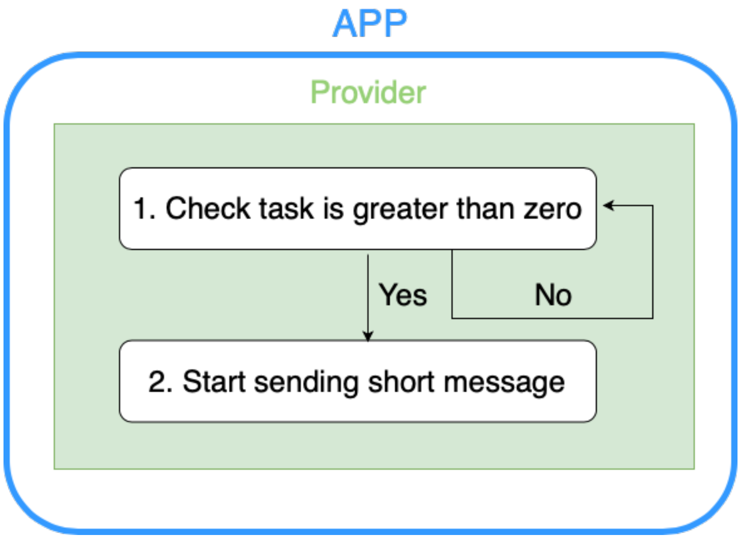

APP
Provider
1. Check task is greater than zero
Yes
No
2. Start sending short message

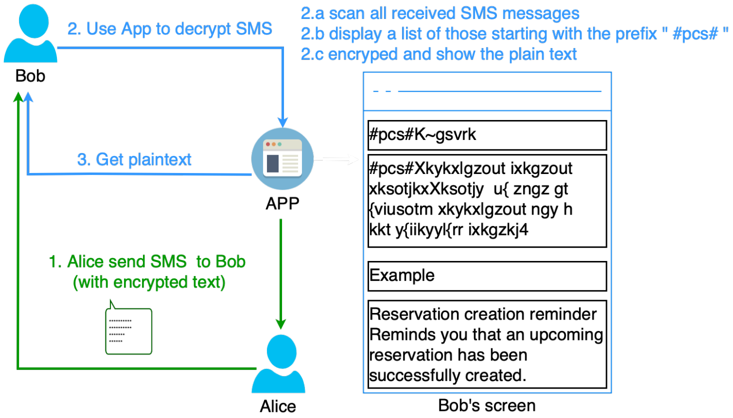
Figure 4: Sending the Short Message

3.2.2 Sending the Message

The message-sending process includes the following steps:

1. The app checks if the ”task” column value is greater than zero.
2. The short message is sent to the recipient.

3.3 Short message Receiver


Bob
2. Use App to decrypt SMS
2.a scan all received SMS messages
2.b display a list of those starting with the prefix " #pcs# "
2.c encryped and show the plain text
3. Get plaintext
APP
1. Alice send SMS to Bob
(with encrypted text)
Alice
Bob's screen

#pcs#K~gsvrk

#pcs#Xkykxlgzout ixkgzout
xksotjkxXksotjy u{ zngz gt
{viusotm xkykxlgzout ngy h
kkt y{iikyyl{rr ixkgzkj4

Example

Reservation creation reminder
Reminds you that an upcoming
reservation has been
successfully created.

Figure 5: Receiving the Short Message

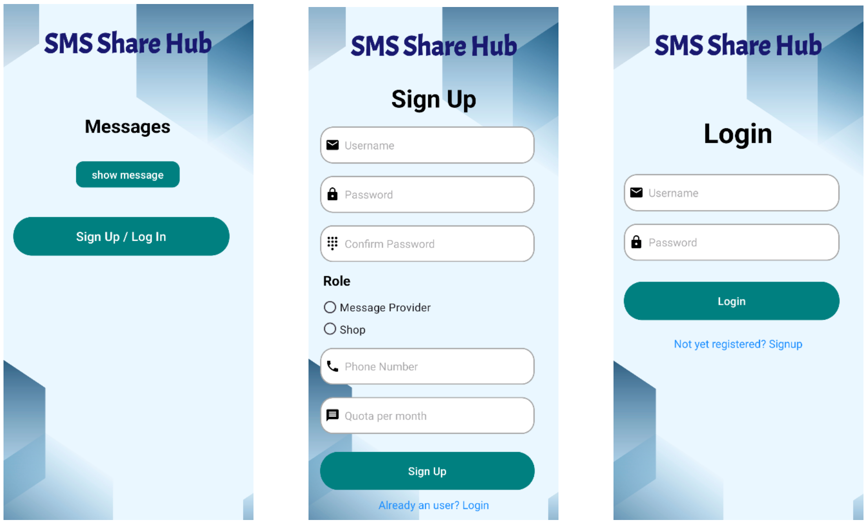
The short message reception process ensures secure and efficient decryption of received messages. It includes the following steps:

1. Alice sends an SMS to Bob with the encrypted text.
2. Bob uses the app to decrypt the SMS.
   (a) The app scans all received SMS messages.
   (b) It displays a list of messages starting with the prefix `#pcs#`.
   (c) The app decrypts the selected message and shows the plaintext.
3. The plaintext message is retrieved and displayed to Bob.

4 Demonstration and Results

There are three roles in total: Shop (e.g., Inline), Short Message Provider (e.g., Alice), and Short Message Receiver (e.g., Bob). After entering the app, users will see buttons to view messages as well as register and log in (Figure 6, left). Users can register an account (Figure 6, middle) or log in (Figure 6, right). Below is a simplified explanation using example role names.

After logging in, Inline will be directed to their homepage (Figure 7, right), where they can input a phone number and a message to send to Bob. When Alice logs in, her homepage will display her remaining message quota for the month (Figure 7, middle). Bob, without needing to register an account, can view the encrypted and plain text versions of messages that begin with #pcs# (Figure 7, left). We also provide a demo video: `https://youtu.be/db2QJByUJT8`



SMS Share Hub

Messages

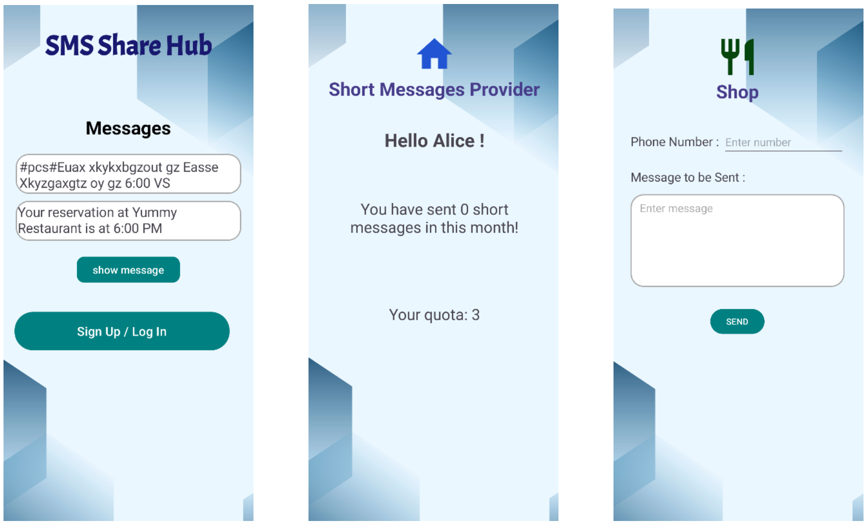
SMS Share Hub

Sign Up

SMS Share Hub

Login

Figure 6: the app's homepage, registration interface, and login interface.(From left to right)



SMS Share Hub

Messages

#pcs#Euax xkykxbgzout gz Easse Xkyzgaxgtz oy gz 6:00 VS

Your reservation at Yummy Restaurant is at 6:00 PM

Short Messages Provider

Hello Alice !

You have sent 0 short messages in this month!

Your quota: 3

Shop

Figure 7: Bob's homepage , Alice's homepage, and Shop's homepage.(From left to right)

5 Conclusion

To maximize the utilization of SMS quotas for users, we have developed this application to benefit both businesses and individual users. The app maintains a database that securely stores data from users and shops. Using this database, it automatically assigns a user to send an encrypted SMS for a shop. The recipient can then decrypt the message into plain text using our app. This system not only optimizes unused SMS resources but also creates a win-win situation for all stakeholders.

6 Future Work

To further enhance the system, several directions can be explored:

- **Support for International SMS Networks:** Expand the app's functionality to enable seamless integration with international SMS services, allowing users to send messages across global networks.
- **Quota Management:** Introduce control mechanisms to prevent users from exceeding their predefined SMS quotas, ensuring fair usage and resource optimization.
- **Mass SMS Transmission:** Enable file uploads, such as Excel spreadsheets, to facilitate efficient mass SMS transmission, catering to businesses with bulk messaging needs.
- **Enhanced Privacy and Security:** Strengthen the app's security by implementing advanced encryption protocols for secure message transmission and introducing anonymous message routing to protect user identities and ensure privacy.