The e-Cheque System

System Specification

Sherif Saad Mohammed

TABLE OF CONTENTS

OVERVIEW OF THE ELECTRONIC CHEQUE SYSTEM

SYSTEM OVERVIEW

Recent years have seen a tremendous increase in e-commerce transactions. The success of e-commerce relies on developing adequate payment technologies. One such technology is e-Cheque.

An e-Cheque is an electronic document which substitutes the paper check for online transactions. Digital signatures (based on public key cryptography) replace handwritten signatures.

The e-Cheque system is designed with message integrity, authentication and non-repudiation features, strong enough to prevent fraud against the banks and their customers. The minimum security requirements supported by the e-Cheque system are as follows:

- Confidentiality: keeping information (e.g. e-mail message, payment order, etc) secret.
- Authentication: knowing and verifying the origin and/or destination of information.
- Integrity: verifying that the data hasn't been tampered with.
- Non-repudiation: knowing that the data, once sent cannot be retracted or denied.

The e-Cheque is compatible with interactive web transactions or with email and does not depend on real-time interactions or on third party authorizations. It is designed to work with paper cheque practices and systems, with minimum impact on payers, payees, banks and the financial system.

Payers and payees can be individuals, businesses, or financial institutions such as banks. E-Cheques are transferred directly from the payer to the payee, so that the timing and the purpose of the payment are clear to the payee.

The payer writes an e-Cheque by structuring an electronic document with the information legally required to be in a cheque and digitally signs it. The payee receives the e-Cheque over email or web, verifies the payer's digital signature, writes out a deposit and digitally signs it.

The payee's bank verifies the payer's and payee's digital signatures, and then forwards the cheque for clearing and settlement. The payer's bank verifies the payer's digital signature and debits the payer's account. Like paper cheques, e-Cheques can bounce or be returned, for stop payment instructions, insufficient funds or accounts being closed.

SYSTEM OPERATION

The e-Cheque system manages the transfer of funds (represented by electronic cheques) between different clients, different banks, and clients and their banks. The e-Cheque system supports the transferability of funds between different clients. This feature does not exist in traditional e-payment systems such as credit card and debit cards systems (e.g. Visa, Master card, Gift Card, etc).

In order to send a cheque, the client simply fills out a standard e-cheque. The system allows clients to define common payees in order to speed the e-cheque creation process. When the cheque has been written it can be easily transferred from the payer to the payee over a secure e-cheque channel. This secure channel will be established between the payer and the payee before the transaction begins.

The e-cheque is automatically signed by the user using his private key based on RSA algorithm and SHA-128; this ensures the authenticity and the integrity of the e-cheque. The signed cheque is then encrypted using a secret key of length 128 bit based on the AES (Advanced Encryption Standard) algorithm to prevent tampering by a third party.

When the payee receives the e-cheque he can open and view it using the e-cheque system. The e-cheque system automatically detects any tampering in the cheque that may have

occurred since its creation. In order to deposit the cheque, the payee simply connects to the bank (which is expected to provide e-cheque services) and uploads the e-cheque to his bank account.

Once the bank receives the e-cheque, it will decrypt it using the e-cheque system. After clearing (i.e. verifying both the cheque signature and account balance) with the payer's bank, the payee's account will be credited accordingly.

THE E-CHEQUE PROTOCOLS

The majority of e-payment systems on the Internet today are based on three-party communication protocols involving a trusted third-party besides the payer and payee. The trusted third-party is needed to authenticate and verify the payment or the transaction process between the clients. Unfortunately this decreases the system performance because of the increased number of messages that must be exchanged. The e-Cheque system has been optimized so as to work with only two parties, namely the payee and payer. The e-cheque operational scenarios are structured around five main protocols as follows:

1. System Setup.
2. Client Registration.
3. Cheque Withdrawal.
4. Cheque Payment.
5. Cheque Deposit.

SYSTEM SETUP

Each client that wishes to use the e-cheque system must obtain a digital identity. A digital identity is achieved by obtaining a valid digital certificate from a recognized certificate authority (CA) (e.g. Verisign). The working scenario for obtaining a digital identity is as follows:

1. An asymmetric key pair (private/public key) is generated on the client side.
2. A certificate request (CRQ) is compiled including both information about the requesting subject (purpose of the requested certificate) and the public key.

3. The CRQ is signed using the newly generated private key.
4. The CRQ is submitted to a CA.
5. A CA validation agent validates the content of the CRQ, including the subject information. If the certificate is intended to identify the subject, the agent or other authorized representative must manually validate the identity of the subject.
6. The CA signs the CRQ with the CA private key to produce a certificate.
7. The certificate is installed at the client and associated with the private key.

Note: by the end of the system setup phase each e-cheque's client holds a valid digital certificate. Each client holds a pair of asymmetric key that represents his digital identity in the e-cheque system.

CLIENT REGISTRATION

A client (payer/payee) must register his digital identity at an e-cheque bank provider. After the client has registered his identity he becomes an owner of an e-cheque account. The scenario for obtaining an e-cheque account is as follows:

1. The client connects to an e-cheque bank provider.
2. The bank accepts the connection request.
3. The bank and the client exchange their digital certificates.
4. The bank and the client authenticate each other using challenge response messages through the exchanged digital certificates.
5. The bank generates its own symmetric key as secure session key and sends it encrypted to the client using the client's public key.
6. The client generates his own symmetric key as secure session key and sends it encrypted to the bank using the bank's public key.
7. The client sends a "create e-cheque account" request to the bank encrypted by the bank's session key.
8. The bank decrypts and validates the e-cheque account request according to the bank e-business rules.
9. The bank creates an e-cheque account for the client and stores the client's e-cheque account information on the bank's server.

10. The bank sends a “created e-cheque account” acknowledgment (message) to the client encrypted by the client session key.
11. The bank and the client close the connection channel.

Note: all the exchanged messages are recorded, encrypted using the destination session key, hashed with a hash function and signed using the source private key to comply with the security requirements. By the end of the client registration phase, the registered client will be the owner of an e-cheque account at an e-cheque bank provider. Now the client can withdraw or deposit e-cheques.

E-CHEQUE WITHDRAWALS

A client can withdraw an e-cheque from his e-cheque account. The e-cheque system supports two main types of e-cheque, namely, prepaid e-cheque and postpaid e-cheque. The scenario for withdrawing a prepaid e-cheque is as follows:

1. The registered client connects to his e-cheque bank provider site.
2. The bank accepts the connection request.
3. The bank and the client exchange their digital certificates.
4. The bank and the client authenticate each other using challenge response messages based on the exchanged digital certificates.
5. The bank generates its own symmetric key as secure session key and sends it encrypted to the client using the client’s public key.
6. The client generates his own symmetric key as secure session key and sends it encrypted to the bank using the bank’s public key.
7. The client creates a prepaid e-Cheque withdrawal request and sends it encrypted using the bank’s session key.
8. The bank decrypts and validates the prepaid e-Cheque withdrawal request, and accepts it according to the bank e-business rules.
9. The bank withdraws the amount of money indicated in the withdrawal request and updates the client’s account balance accordingly.

10. The bank creates an e-Cheque document with a balance equal to the requested amount.
11. The bank calculates the hashing code of e-cheque document and signs this hash with the bank's private key.
12. The bank sends the e-cheque data (e-cheque object and sign object) to the client encrypted using the client session key.
13. The client receives the encrypted e-cheque data.
14. The client decrypts and validates the received e-cheque data.
15. The client stores the encrypted e-cheque data.

The scenario for withdrawing a post payment e-cheque is as follows:

1. The registered client connects to his e-cheque bank provider.
2. The bank accepts the connection request.
3. The bank and the client exchange their digital certificates.
4. The bank and the client authenticate each other using challenge response messages based on the exchanged digital certificates.
5. The bank generates its own symmetric key as secure session key and sends it encrypted to the client using the client's public key.
6. The client generates his own symmetric key as secure session key and sends it encrypted to the bank using the bank's public key.
7. The client creates a postpaid e-cheque withdrawal request and sends it encrypted using the bank's session key.
8. The bank decrypts and validates the postpaid e-cheque withdrawal request, and accepts it according to the bank e-business rules.
9. The bank creates an e-cheque document.
10. The bank calculates the hashing code of the generated e-cheque object and signs the hashing code with the bank's private key.
11. The bank sends the e-cheque data (e-cheque object and sign object) to the client encrypted using the client's session key.
12. The client receives the encrypted e-cheque data.

13. The client decrypts and validates the received e-cheque data.
14. The client stores the encrypted e-cheque data.

Note: all the exchanged messages are recorded, encrypted using the destination session key, hashed with a hash function and signed by the source private key to comply with the security requirements. By the end of the e-cheque withdrawal phase the client now holds one or more e-cheque objects which he could use for internet payment.

E-CHEQUE PAYMENT

Any two clients can exchange any number of e-cheques as follows:

1. The sending client obtains the IP address of the receiving client by searching through the e-cheque network.
2. The sending client sends a connection request to the receiving client.
3. The sending and receiving clients exchange their digital certificates.
4. The sending and receiving clients authenticate each other using challenge response messages from the exchanged digital certificates.
5. The sending client generates a symmetric key as a secure session key and sends it to the receiving client encrypted using the receiving client's public key.
6. The receiving client generates a symmetric key as a secure session key and sends it to the sending client encrypted using the receiving client's public key.
7. The sending and receiving clients exchange e-business data encrypted using each other's session key.
8. The sending and receiving client agree on a final e-business data.
9. The sending client sends an e-cheque payment request to the receiving client encrypted using the receiving client's session key.
10. The receiving client decrypts and validates the e-cheque payment request, and accepts it according to the agreed e-business data.
11. The receiving client sends a message indicating acceptance of the e-cheque payment request, encrypted using the sending client's session key.
12. The sending client encrypts one or more of his e-cheque object(s) using the receiving client's session key, and sends it to the receiving client.

13. The sending client decrypts and validates the received e-cheque object(s).
14. The sending client generates an acknowledgment for the received e-cheque(s) and e-business data and sends it encrypted using the sending client's session key.
15. The sending client decrypts and validates the received acknowledgment and stores it.

Note: all the exchanged messages are recorded, encrypted using the receiving client's session key, hashed with a hash function and signed by the sending client's private key to comply with the security requirements.

E-CHEQUE DEPOSIT

The client (customer/merchant) can deposit an e-cheque object into his e-cheque account at his e-cheque bank provider.

1. The client connects to an e-cheque bank provider.
2. The bank accepts the connection request.
3. The bank and the client exchange their digital certificates.
4. The bank and the client authenticate each other using challenge response messages from the exchanged digital certificates.
5. The bank generates its own symmetric key as secure session key and sends it encrypted to the client using the client's public key.
6. The client generates its own symmetric key as secure session key and sends it encrypted to the bank using the bank's public key.
7. The client creates an e-cheque deposit request (ECDR) and sends it to the bank encrypted using the bank's session key.
8. The bank decrypts and validates the deposit request, and accepts it according to the bank e-business rules.
9. The bank sends a message indicating acceptance for the deposit request to the client, encrypted using the client's session key.
10. The client decrypts and validates the received acceptance message.
11. The client selects one or more e-cheque object(s) and sends it to the bank encrypted using the bank's session key.

12. The bank decrypts and validates the received e-cheque object(s).
13. The bank starts the cashing process of the received e-cheque object(s) according to the characteristics of each e-cheque object.
14. The bank updates the client's account balance by withdrawing the check amount.
15. The bank generates an acknowledgment report about the cashing process and sends it to the client encrypted using the client's session key.
16. The client decrypts and validates the received acknowledgment report, and then stores it.

Note: all the exchanged messages are recorded, encrypted using the receiver session key, hashed with a hash function and signed by the sender's private key to comply with the security requirements.

ELECTRONIC CHEQUE ARCHITECTURE

The e-Cheque system consists of two main components, the e-cheque client and the e-cheque server. The client component is used by e-cheque clients (payer/payee) to perform different operations. These operations are registration, e-cheque withdrawal, deposit, writing cheque, sending and receiving cheque. The server component is used by financial institutions (e.g. Bank) to manage e-Cheque users' accounts.

E-CHEQUE CLIENT:

The e-Cheque client consists of 20 classes grouped into 4 java packages. These packages are cryptoCheque, eChequeIO, eChequeGUI and eChequeData.

1. Package cryptoCheque:

The cryptoCheque package contains 4 classes; these classes provide different security services to the e-Cheque system. The 4 classes are:

1.1 Class: AESCrypt

This class provides AES encryption, decryption wrap and unwrap services. The class has no member variables and it contains four public methods.

1.2 Class: DigitalSigneture

The DigitalSigneture class provides secure digital signature services to the e-cheque user. The system use this class to sign a message or an e-cheque object with the user private key or to verify the signature of a user on a given message or an e-cheque object. This class does not contain any member data; it contains 2 public member functions and one constructor.

1.3 Class: RSAGenerator

The RSAGenerator class generates a pair of RSA keys (public and private) that are used by the system to provide digital signature services. This class has no member data and it has one public member function.

1.4 Class: SymmetricKeyGenerator:

The SymmetricKeyGenerator class used to generate symmetric keys for AES encryption and decryption. The class contains two public methods, one to generate random key, and the second one to generate a key based on an input string.

2. Package eChequeIO

This package contains classes used to provide Input/Output services for the e-cheque system. This package contains 5 classes:

2.1 Class: DigitalCertificateIO

The DigitalCertificateIO class is responsible for reading and writing DigitialCertificate objects from/to hard drive. This class contains two public member functions and one constructor.

2.2 Class: EChequeDataManager

The EchequeDataManager class is responsible for secure storage of the e-Cheque user data such as personal information, bank information and other business data used inside the e-Cheque system. This class contains 7 private member data and 5 public member functions and one constructor.

2.3 Class: EChequeIO

The EChequeIO class is responsible for reading and writing e-cheque objects from/to hard drive. This class contains two public member functions and one constructor.

2.4 Class: EChequeClient

The EChequeClient class is responsible for reading and writing e-cheque data and objects from/to network sockets. In addition, this class is responsible for starting connection to the e-Cheque Server. This class contains 19 private member data, 3 constructors, 2 public member functions and 5 private member functions.

2.5 Class: EChequeServer

The EChequeClient class is responsible for reading and writing e-cheque data and objects from/to network sockets. In addition, this class is responsible for accepting connection

request sent by the e-Cheque Client. This class contains 12 private member data, one constructor, 2 public member functions and 5 private member functions.

3. Package eChequeGUI

This package contains the classes that implement the e-Cheque GUI components. This package contains 5 classes:

3.1 Class: LoginECheque:

The loginECheque class extends the JFrame class and it presents the e-Cheque login screen. It is the main class of the system, it contains the main method. This class contains 12 private member data and 4 public member functions.

3.2 Class: ElectronicChequeJFrame:

The ElectronicChequeJFrame class extends the JFrame class. It presents the main screen of the system. This class contains 28 private member data, one constructor and 10 public member functions.

3.3 Class: EbankingJFrame:

The EbankingJFrame class extends the JFrame class. This class provides the GUI of the e-banking service provided by the e-Cheque system. This class contains 14 private member data, one constructor and 5 private member functions.

3.4 Class: ChequeJFrame:

The ChequeJFrame class extends the JFrame class. This class provides the GUI of the e-Cheque object. The class contains 34 private member data, one constructor and 15 private member functions.

3.5 Class: RegistrationJFrame

The RegistrationJFrame class extends the JFrame class. This class provides the GUI of the user registration process. The class contains 32 private member data, one constructor and 3 private member functions.

3.6 Class: SendChequeJFrame

The SendChequeJFrame class extends the JFrame class. This class provides the GUI of the send e-Cheque process. The class contains 18 private member data, one constructor and 3 private member functions.

3.7 Class: ReceiveChequeJFrame

The ReceiveChequeJFrame class extends the JFrame class. This class provides the GUI of the send e-Cheque process. The class contains 18 private member data, one constructor and 3 private member functions.

4. Package eChequeData:

This package contains the classes that implement the e-Cheque data such as the e-Cheque object, digital certificate and registration data. This package contains three classes.

4.1 Class: DigitalCertificate

The DigitalCertificate class presents the digital certificate used by the e-cheque users. The class contains 8 private member data and 16 public member functions and one constructor.

4.2 Class: ECheque

The ECheque class is a template class that represents the e-Cheque object. The class contains 22 public member functions, one constructor and 11 private member data.

4.3 Class: EChequeRegistration

The EchequeRegistrationclass represents the user registration object. The class holds user registration data and business information. The class contains 23 public member functions, one constructor and 11 private member data.

E-CHEQUE SERVER:

The e-Cheque system on the bank side contains 10 classes. These classes are grouped into 4 java packages. These packages are cryptoCheque, eChequeIO, eChequeData and eBank. The classes in these packages are the same as for the e-Cheque client. The differences between the e-Cheque on the client and the e-Cheque on the bank side are:

1. Class: EChequeBankSideJFrame:

This class extends the JFrame class and presents the GUI of the e-Cheque system on the bank side. This is the main class of the system on the bank side. This class contains 8 private member data, one constructor and 4 private member functions.

2. Class: BankSever:

The BankServer class is responsible for accepting e-Cheque client connection request to the bank side. This class contains one constructor, one private member data and one public member function.

3. Class: EChequeServer:

The EChequeServer is responsible for processing e-Cheque client connections. These include processing and replying client's requests. This class contains 6 private member data, one constructor, 7 private member functions and 2 public member functions.

4. Class: EchequeDB:

The EchequeDB class is responsible for managing and modifying the e-Cheque database on the bank side. This includes managing the clients account and the e-Cheque data. This class contains 8 private member data, one constructor, 4 private member functions and 3 public member functions.

PACKAGE DIAGRAM

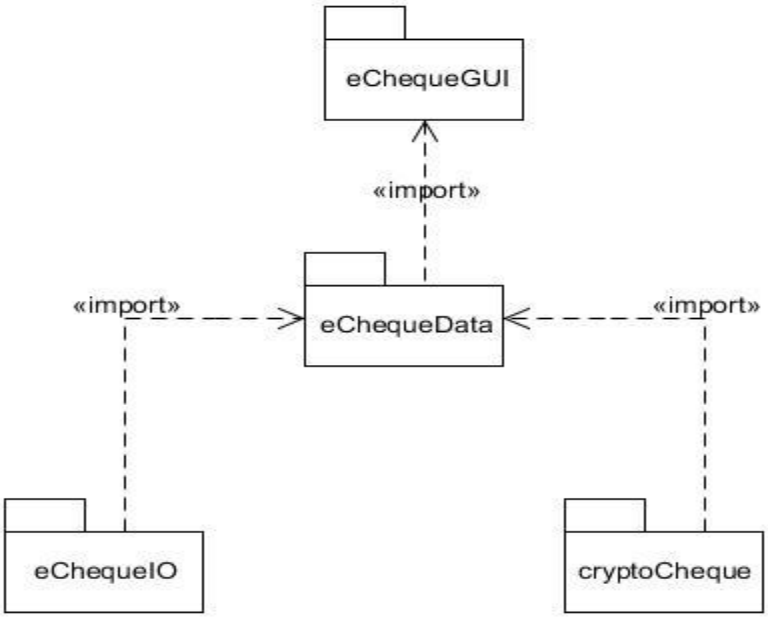

FIGURE 1: THE E-CHEQUE PACKAGE DIGARAM

THE CLASS DIAGRAM

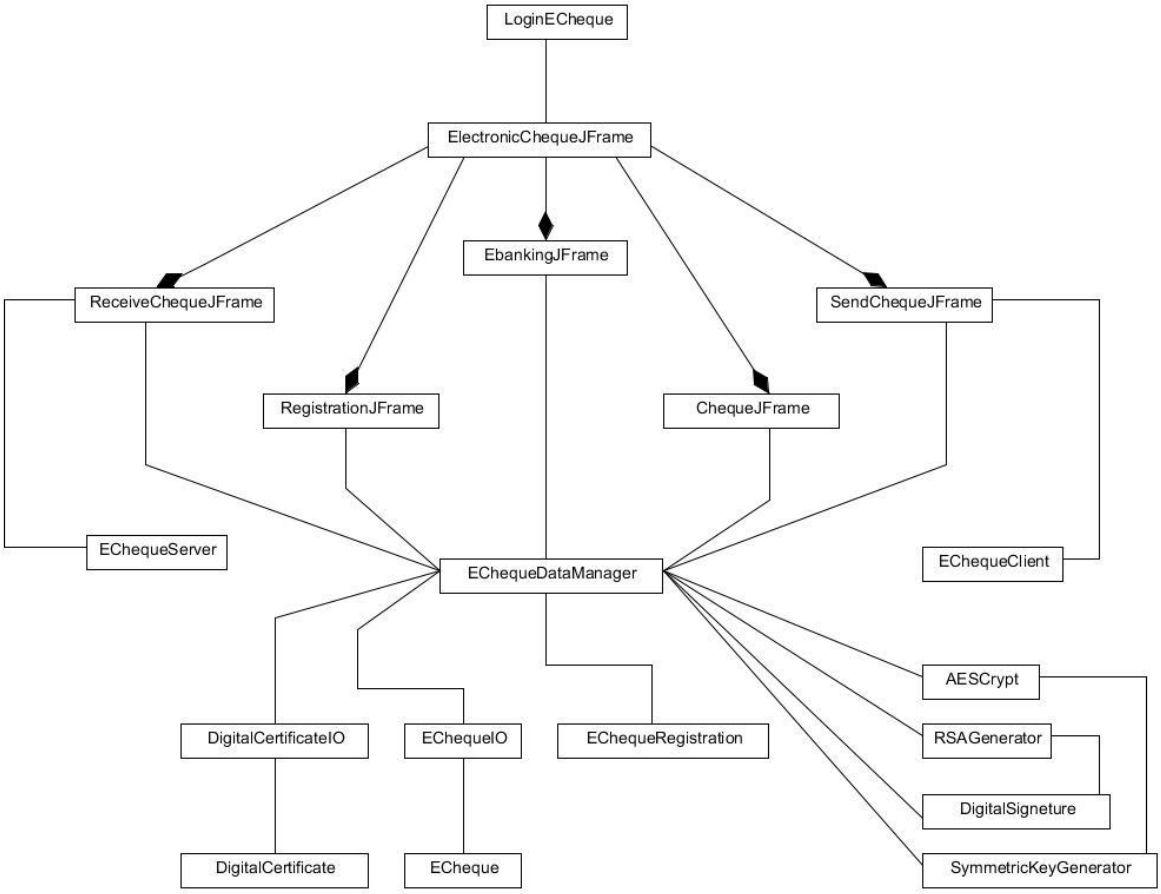

FIGURE 2: E-CHEQUE CLASS DIAGRAM

SEQUENCE DIAGRAMS

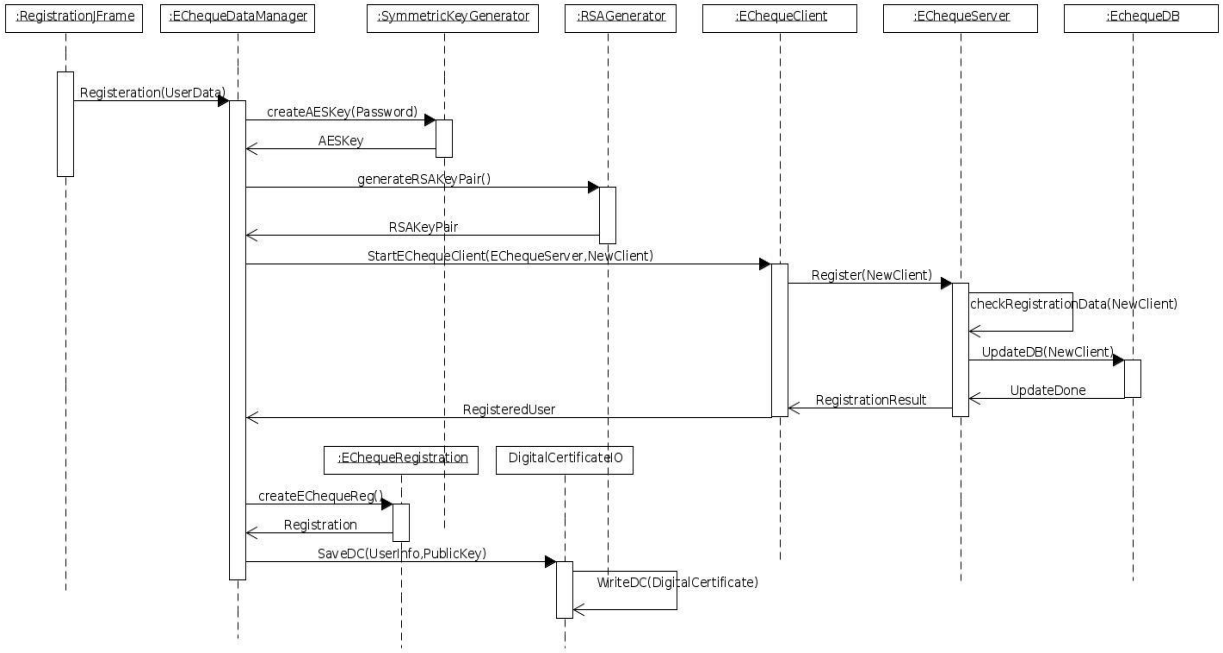

FIGURE 3: E-CHEQUE REGISTRATION SEQUENCE DIAGRAM

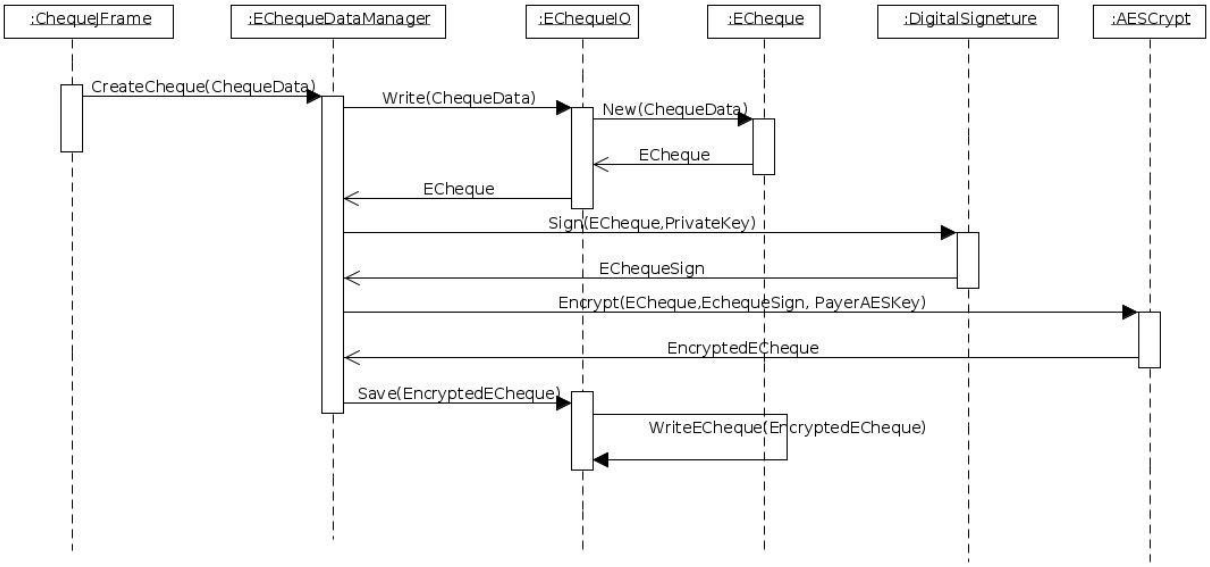

FIGURE 4: CREATE E-CHEQUE SEQUENCE DIAGRAM

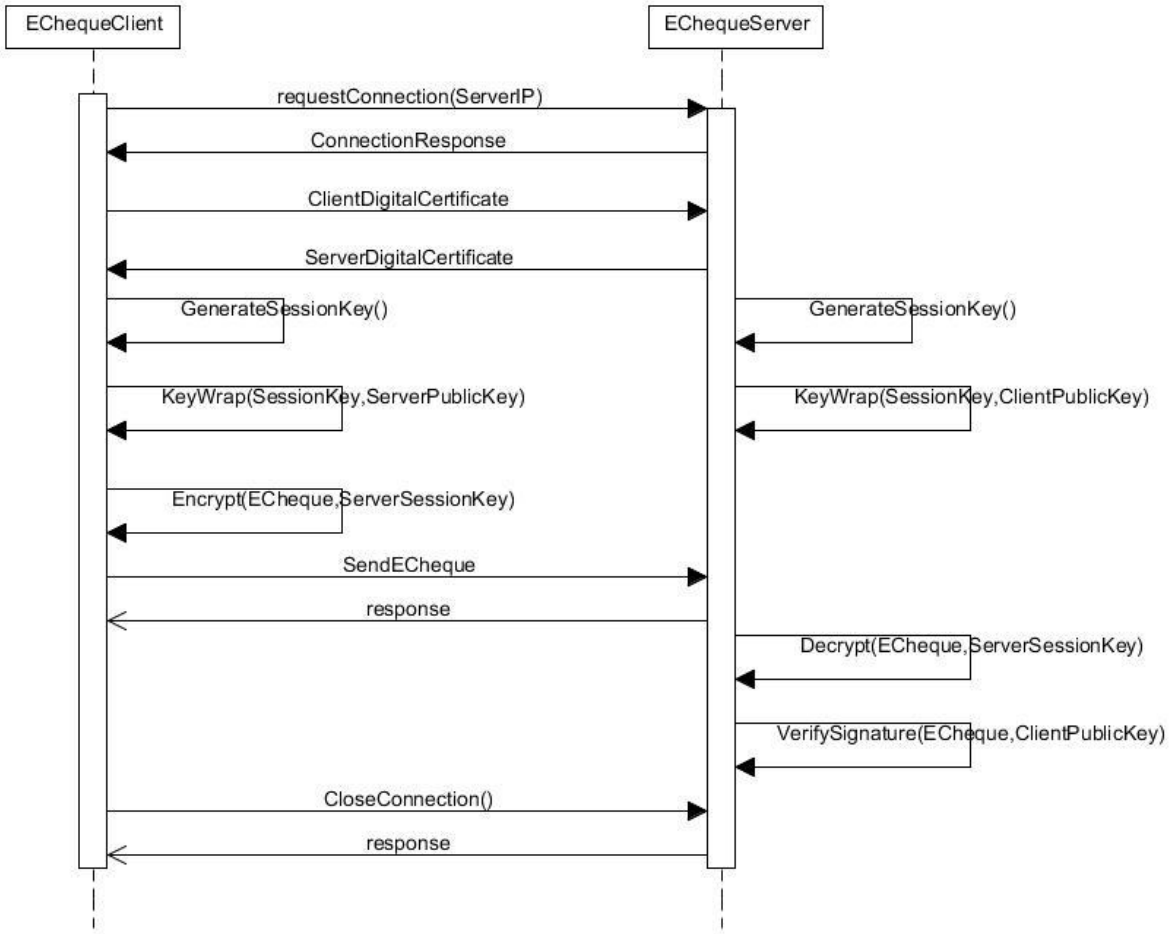

FIGURE 5: SEND ECHEQUE SEQUENCE DIAGRAM

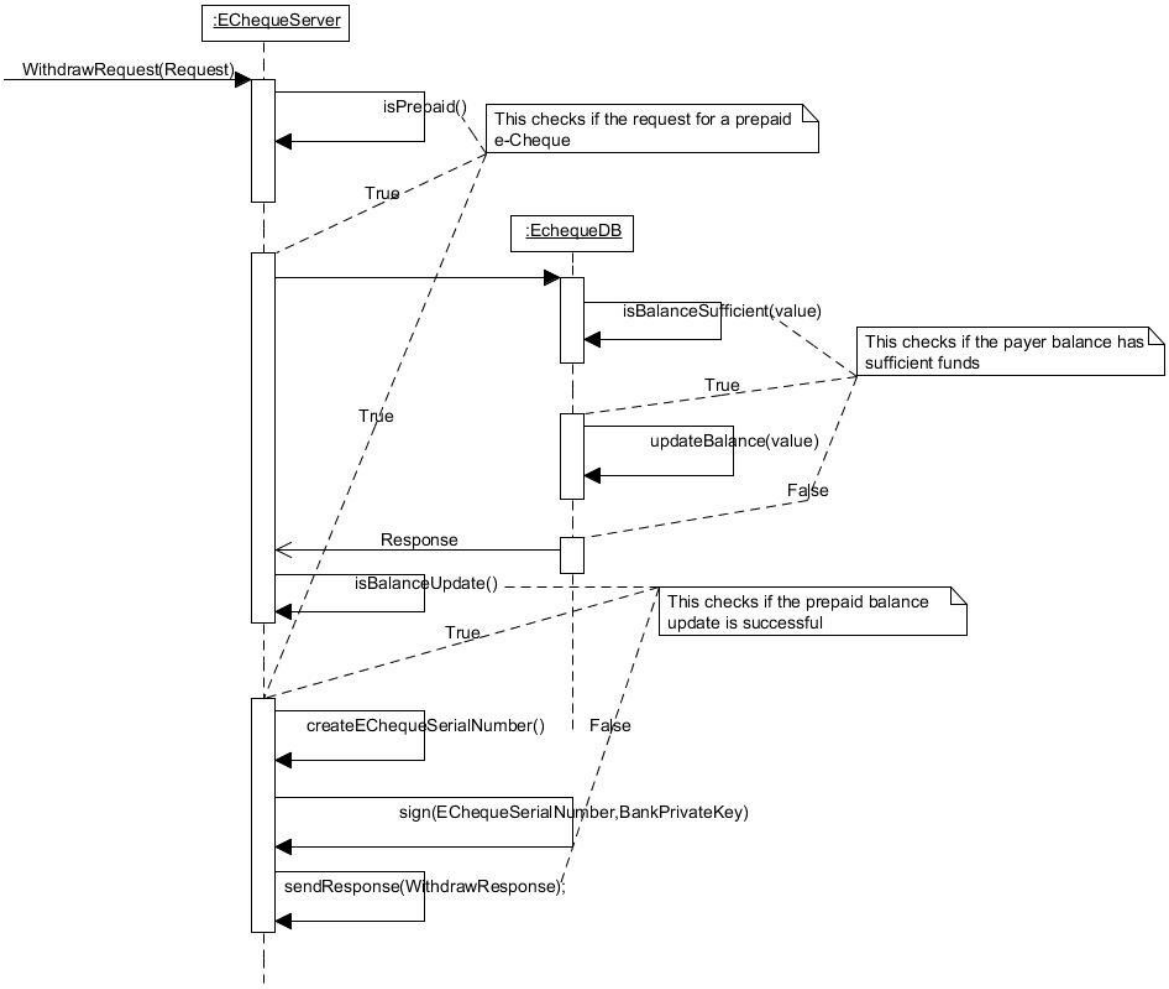

FIGURE 6: WITHDRAW ECHEQUE SEQUENCE DIAGRAM

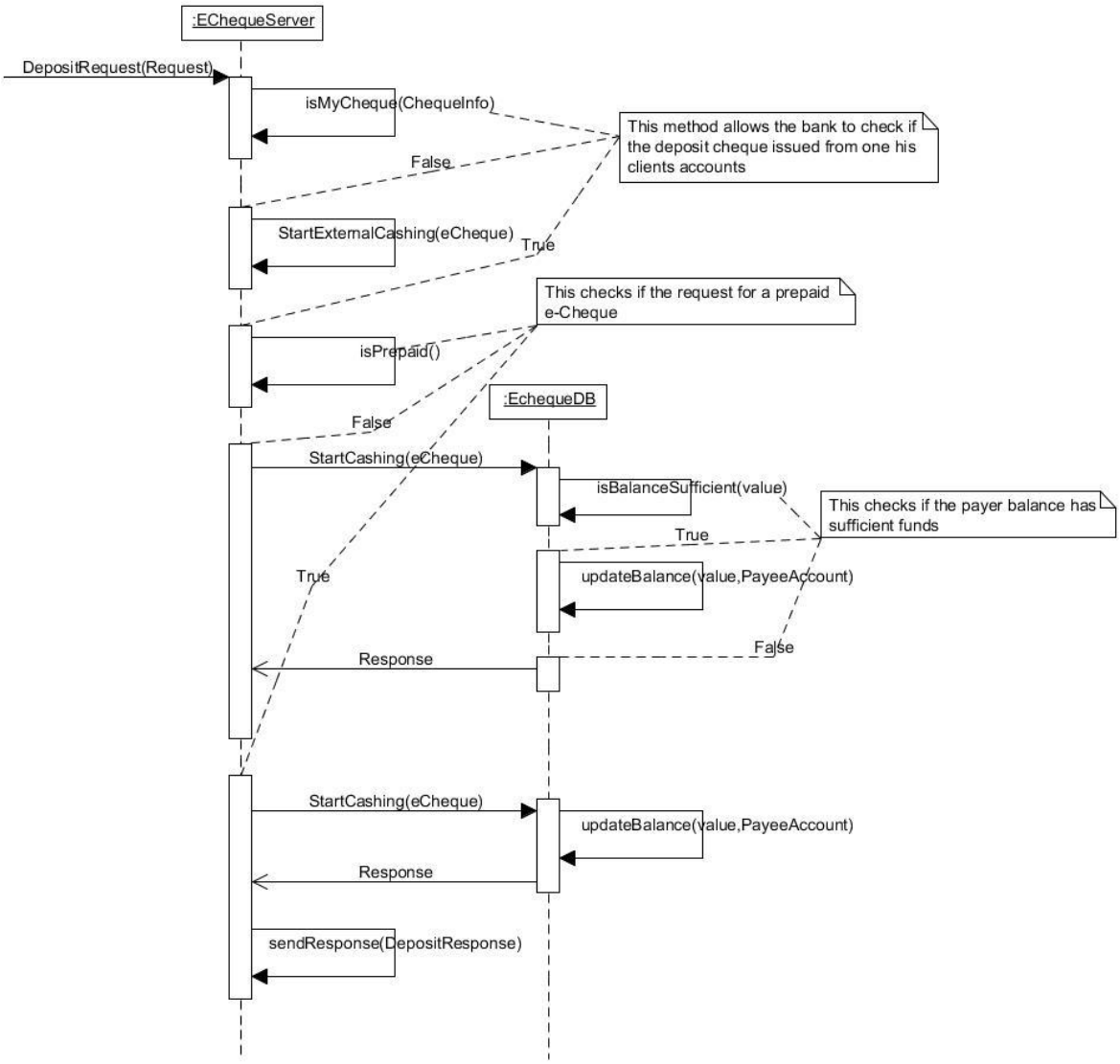

FIGURE 7: DEPOSIT ECHEQUE SEQUENCE DIAGRAM

USE CASE DIAGRAMS

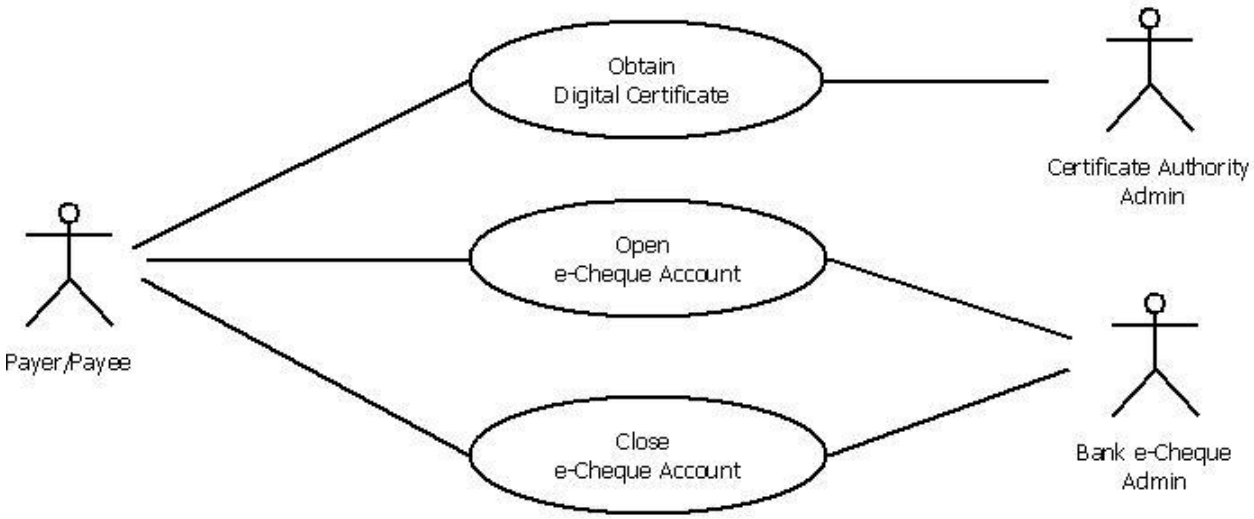

FIGURE 8: COMMON-USE CASES

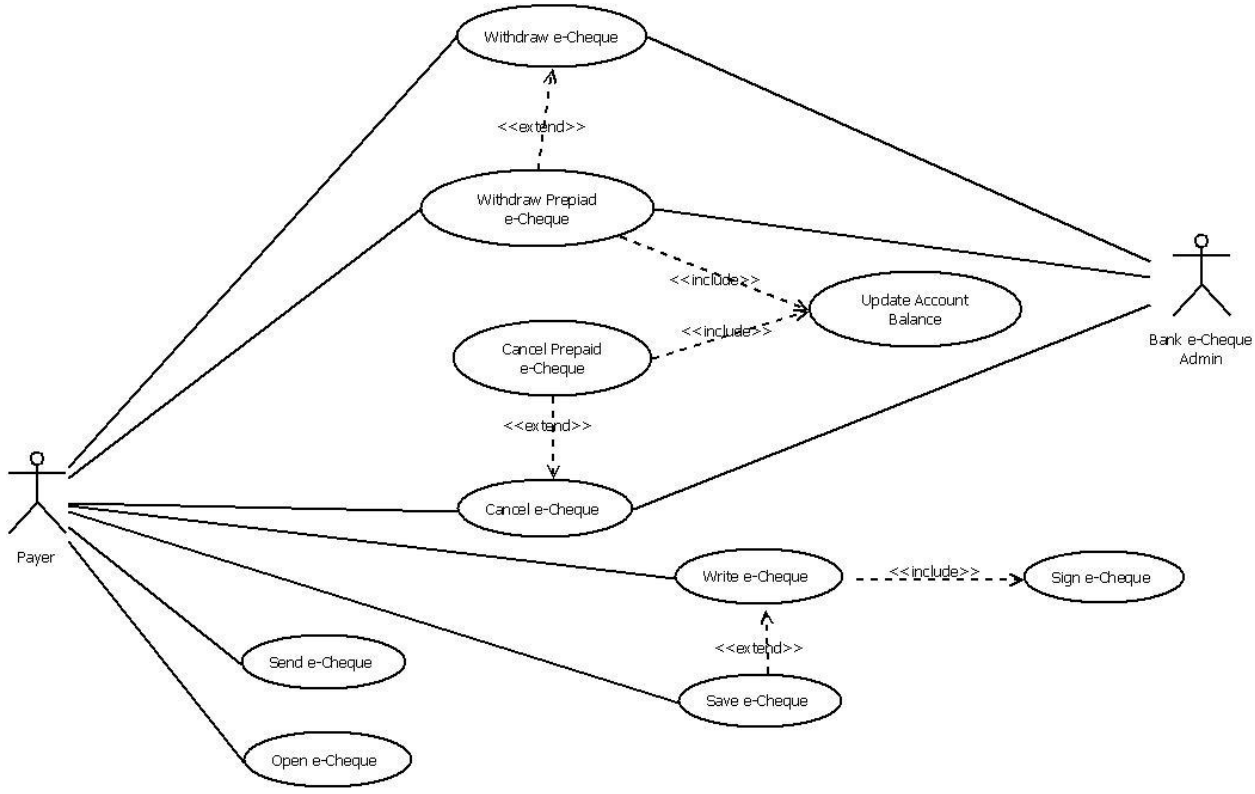

FIGURE 9: PAYER-USE CASES

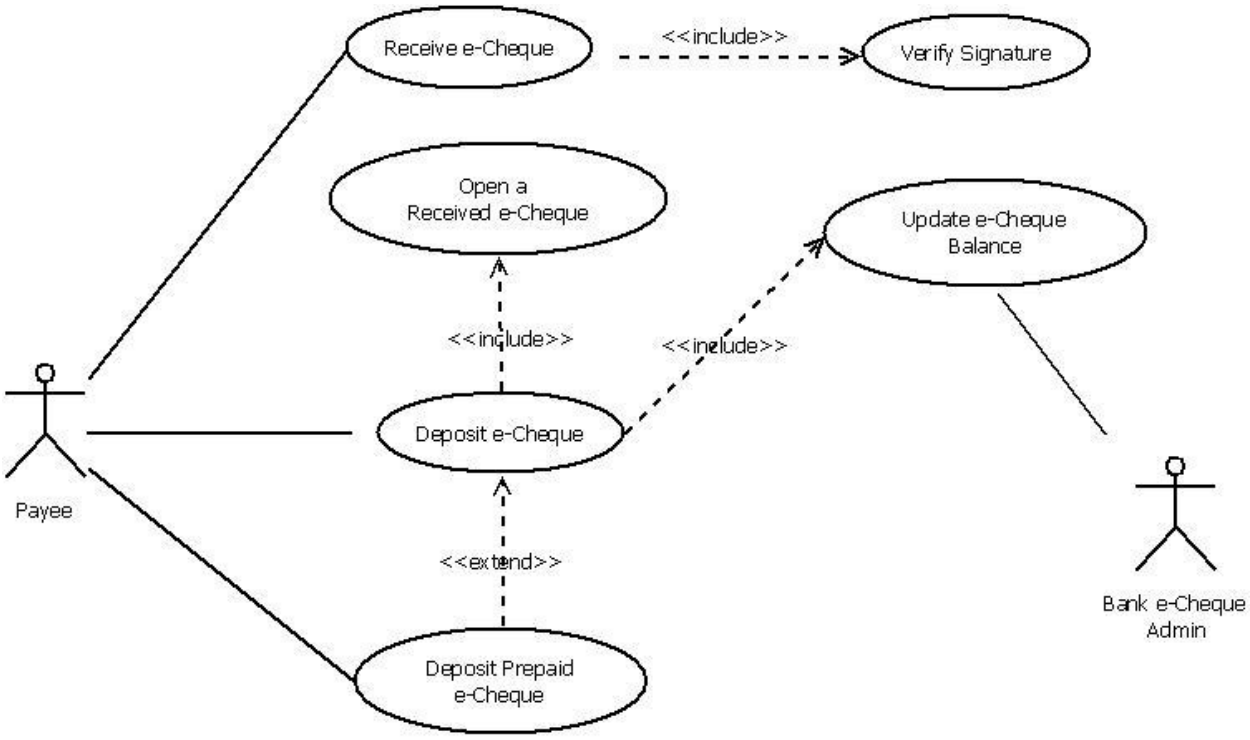

FIGURE 10: PAYEE-USE CASES