SOFTWARE ARCHITECTURE

Final Project

2018-2019

ISTANBUL AYDIN UNIVERSITY
Software Engineering Department

Money Transfer System

Version 1.0 approved

Architecture Documentation

Prepared by:

Hasan Sidawi	**B1705.090059**
Samiullah Niazi	**B1605.090057**
Majed Bawarshi	**B1605.090072**
MHD Omar Bahra	**B1605.090065**

Supervisor: **Assoc. Prof. (Ph.D.) ILHAM HUSEYINOV**

Contents

Short Description of Scenario:

The customer goes to the employee at Money Transfer office for money transfer operations. System takes the information of the customer and does the operation the customer requires.

USE CASE:

Scope:

Money Transfer System.

Primary Actor:

Employee.

Stakeholders and Interests:

- Sender: Person who sends the money.
- Receiver: Person who receives the money.
- Employee: The person who's responsible for the operation between sender and receiver.
- Company: The intermediary between sender and receiver, so it is own interest is to satisfy customers.

Preconditions:

- Employee is identified and authenticated.
- All locations around the world are stored in the system.

Success Guarantee (Post-Conditions):

- The money is transferred.
- Receiver received the money.
- Receipt is printed for both sender and receiver.

Main success scenario:

Sending:

1. Sender arrives at money transfer office to transfer money.
2. Sender fills the form (Information of himself/herself and receiver).
3. Employee starts a new transaction operation.
4. Employee enters the information of sender and receiver and the amount of money to be transfered.
5. System records one transaction.
6. Sender pays with card or cash.
7. System prints receipt (10 digits code) and provide it to the sender.

Receiving:

1. Receiver goes to the office to receive money.
2. Receiver fills a form (Information of himself/herself, sender and 10 digits code).
3. Employee enters information in the system.
4. The system presents the transaction.
5. Receiver receives payment and receipt.
6. System updates and sends an approving message to the sender.

Extensions (or Alternative flows):

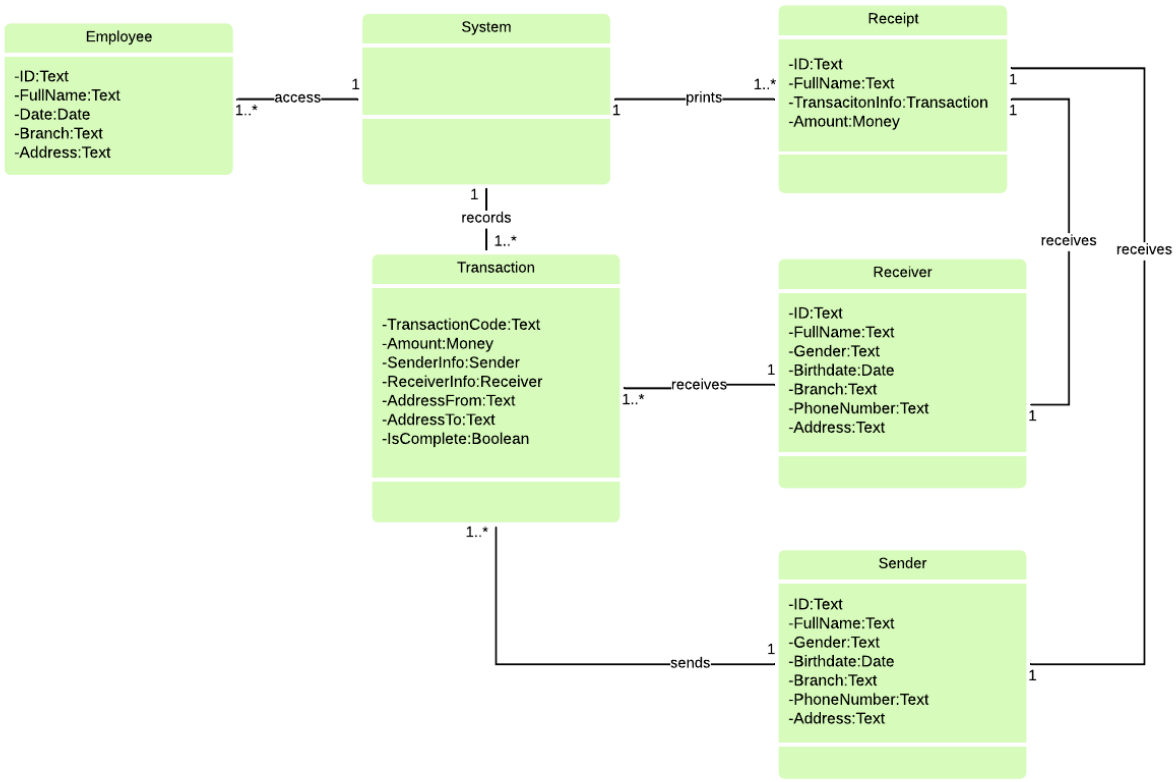
1. The system failed to authenticate the operation:
    a. The employee/sender has entered wrong information about the receiver or the sender.
    b. The employee/receiver has entered wrong information about the sender/receiver.
2. The system is not functioning (crashed)
3. The sender can cancel the transaction:
    a. before the receiver receives it.
    b. If receiver could not receive the money, then sender can take the money back.

Special Requirements:

1. Monitor.
2. Credit Card reader.
3. Electronic Signature reader.

4. Keyboard.
5. Mouse.
6. Printer.

DOMAIN MODEL DIAGRAM (DEVELOPED BY GRAMMAR METHOD):



candidates of verbs or associations:

- arrive
- Provide
- Sends
- Transfer
- Fills
- Starts
- Enters
- Records

- Prints
- Receives
- Presents
- Update
- Pays

candidates of classes:

- Sender.
- Receiver.
- Employee.
- Receipt.
- Form.
- System .
- Transaction.
- Payment.

candidates of methods or operation contracts:

- startTransaction()
- endTransaction()
- isCodeValid(code:Text)
- printRecipt(amount:Money, transaction: Transaction, fullName:Text)
- insertSenderInfo(id:Text, fullName:Text)
- insertReceiverInfo(id:Text, fullName:Text)
- createTransactionCode()
- getPayment(amount:Money)

candidates of attributes:

- TransactionCode.
- Amount.
- SenderInfo.
- ReceiverInfo.
- AddressFrom.
- AddressTo.
- IsComplete.

- ID.
- FullName.
- TransacitonInfo.

Nouns:

- Sender.
- Receiver.
- Employee.
- Receipt.
- System.
- Transaction.

Attributes:

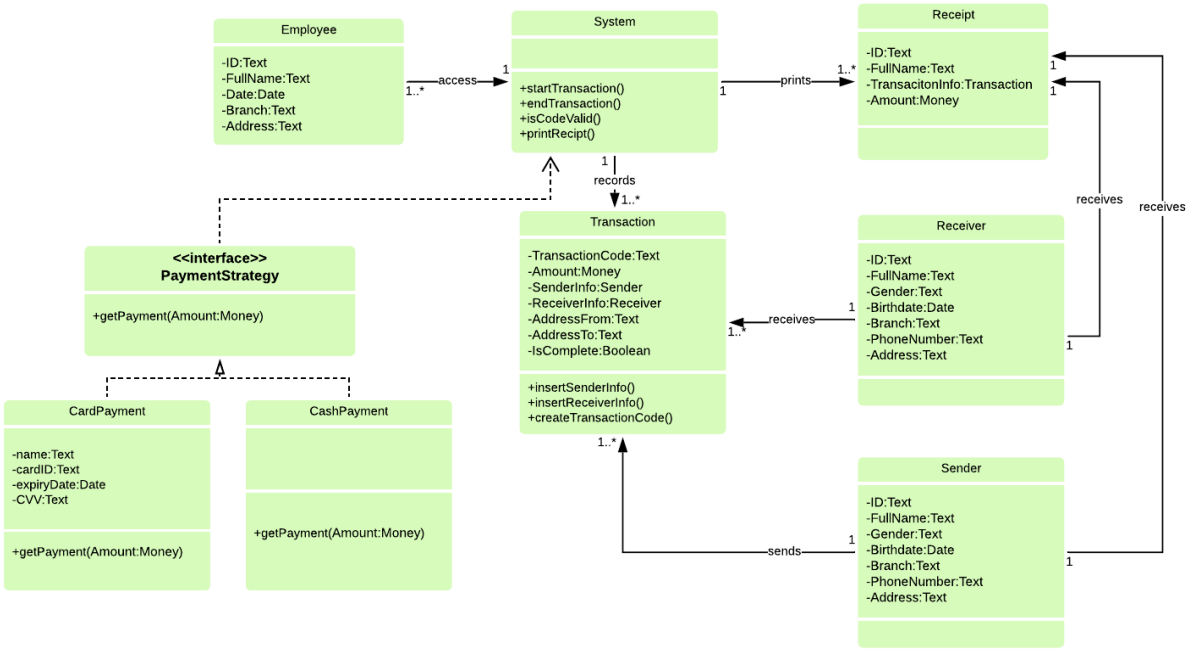
1. Sender:
    a. ID.
    b. fullName.
    c. Gender.
    d. birthDate.
    e. Branch.
    f. phoneNumber.
    g. Address.
2. Receiver:
    a. ID.
    b. fullName.
    c. Gender.
    d. BirthDate.
    e. Branch.
    f. PhoneNumber.
    g. Address.
3. Transaction:
    a. TransactionCode.
    b. Amount.
    c. SenderInfo.
    d. ReceiverInfo.
    e. AddressFrom.
    f. AddressTo.
    g. IsComplete.
4. Employee:
    a. ID.
    b. FullName.
    c. Date.
    d. Branch.
    e. Address.
5. Receipt:
    a. ID.
    b. FullName.
    c. TransactionInfo.
    d. Amount.

Verbs:

- Access.
- Prints.

- Sends.
- Receives.

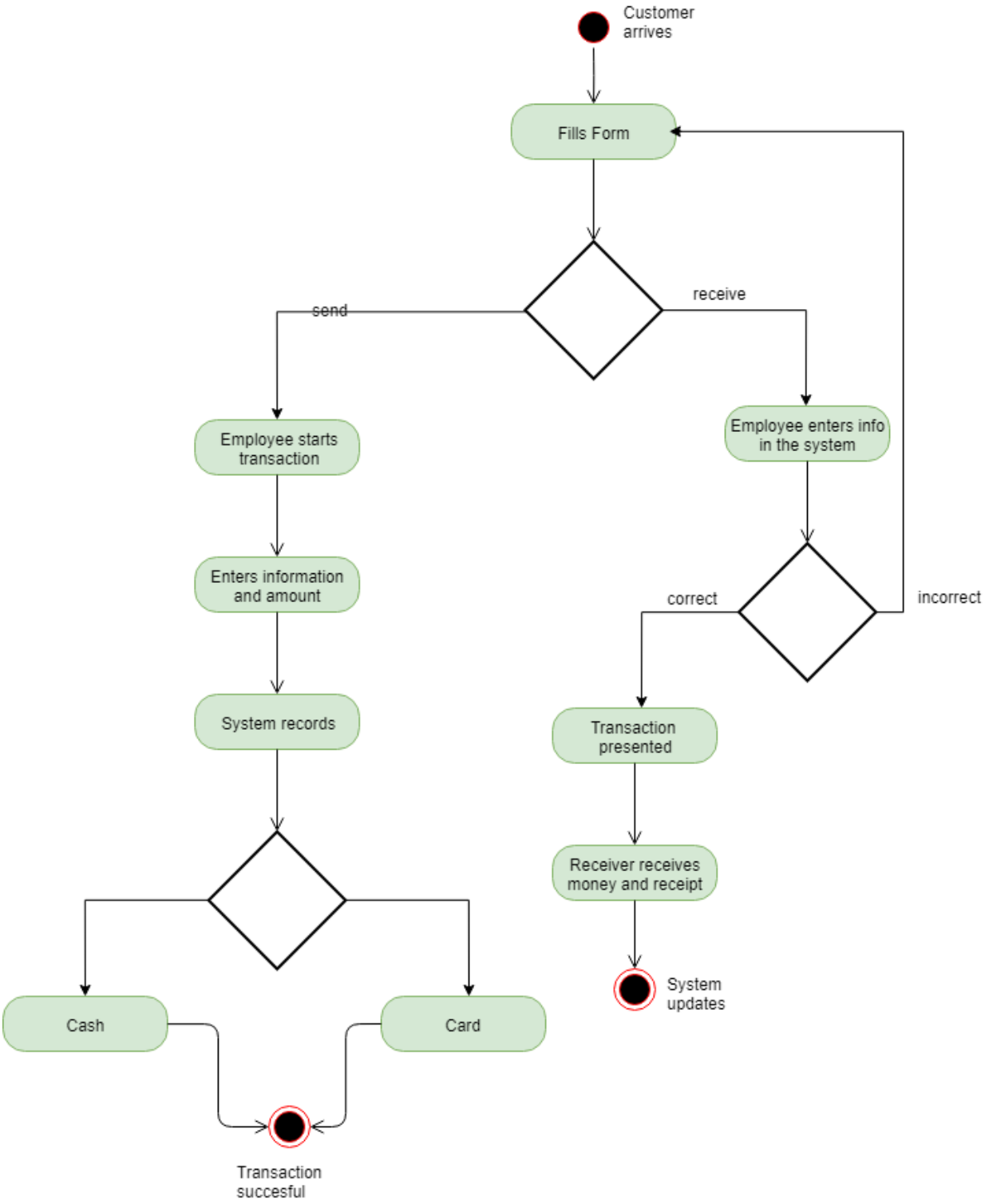
Class Diagram:



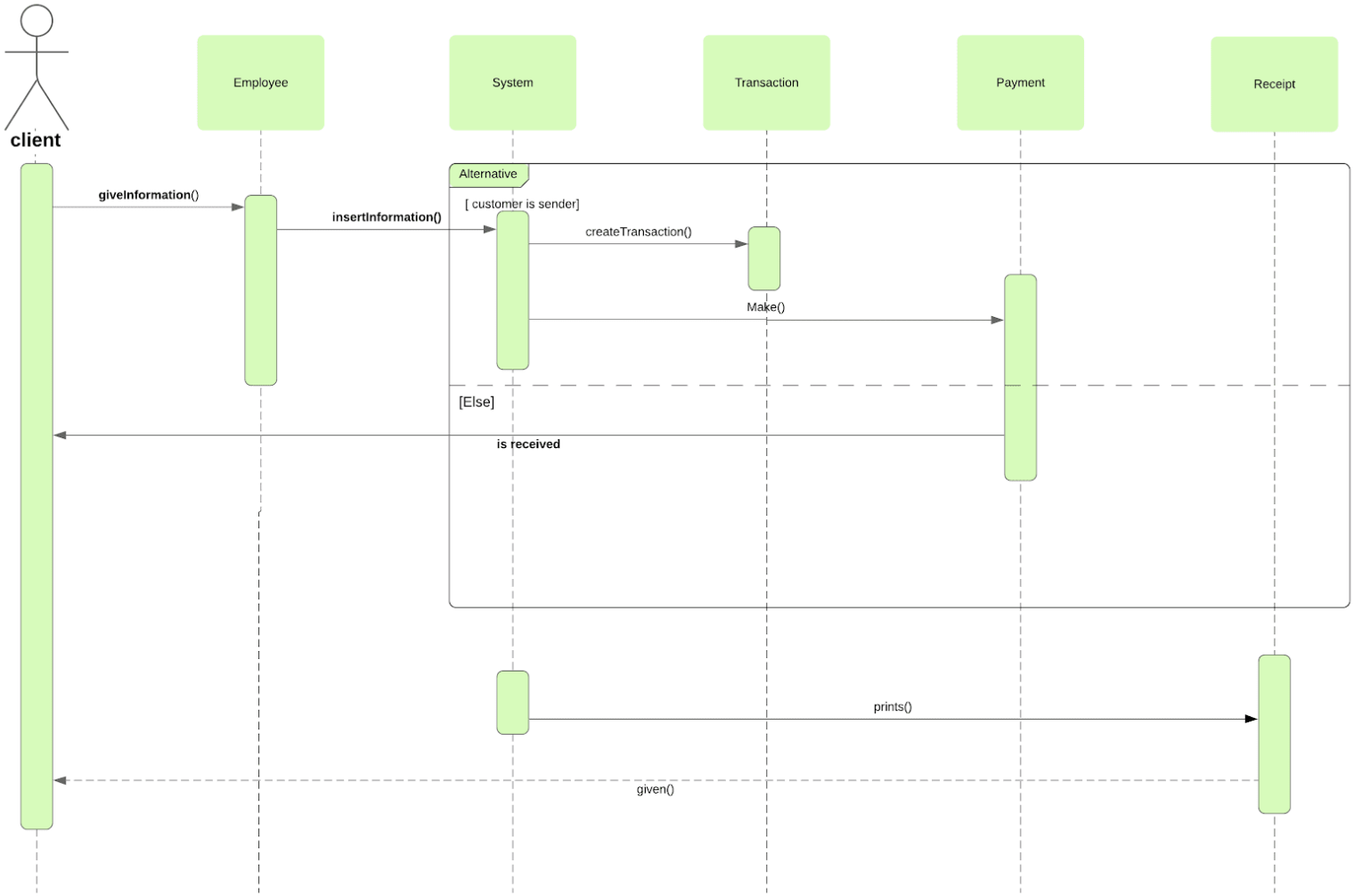
Pattern:

a) System is creator. It creates visitor object.
b) System is the main controller.
c) Transaction class is the information expert. It generates the transaction code.

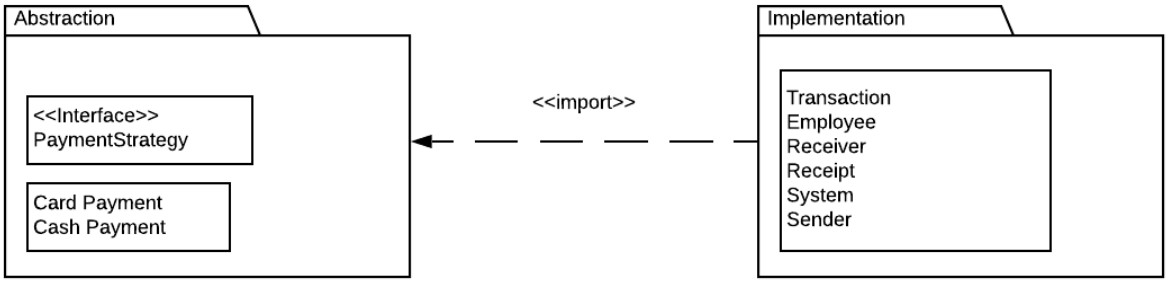
Activity Diagram:



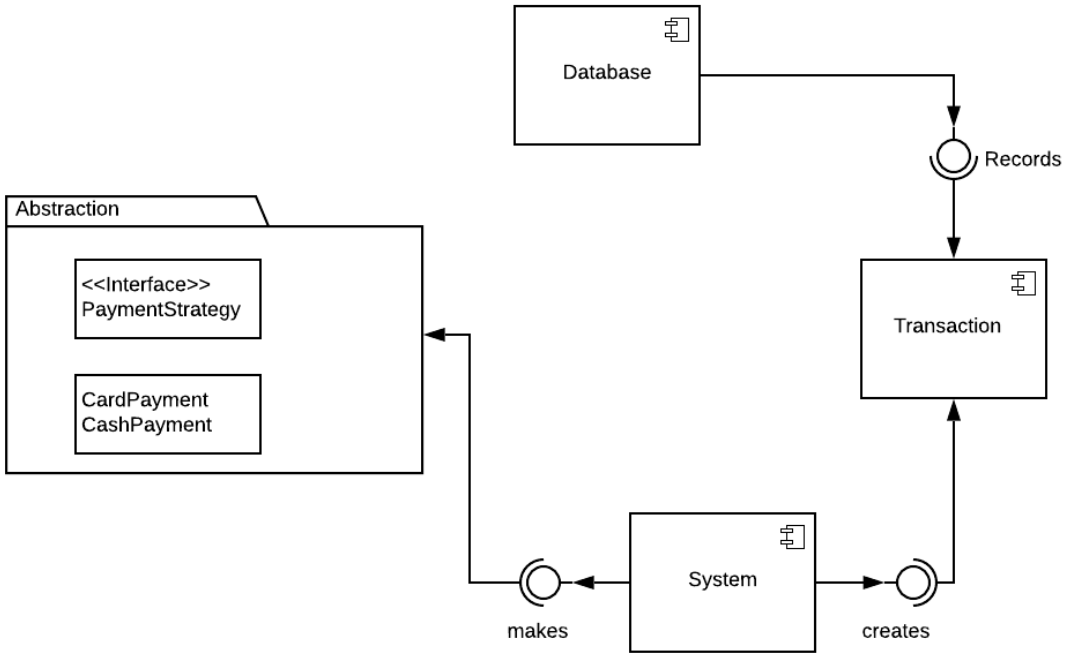
Sequence Diagram:



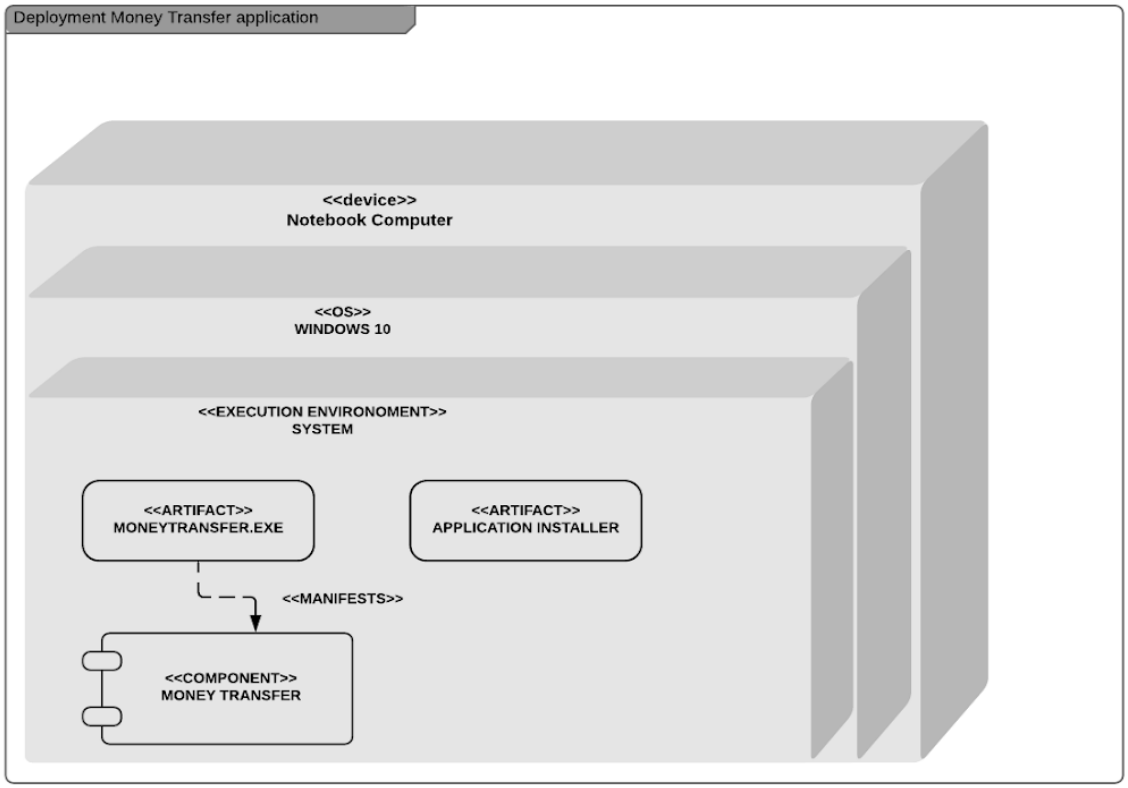
Package Diagram:



Component diagram:



Deployment Diagram:



Quality Attributes:

- Functionality.
- Reliability.
- Maintainability.
- Efficiency.
- Usability.
- Portability.

Architectural Pattern:

Layered architecture.

Five reasons:

1. Because security is very important in our system(to ensure customer money safety).
2. And performance does not have high priority.
3. As our system won't be used by clients only employees will use our software, so usability is not a very important thing.
4. Layered architecture increases maintainability, flexibility and scalability.
5. Maintainability is important in case of any error from the sender or employee it should be changable.

References:

1. https://www.lucidchart.com