

The system: A public phone booth on a street
Requirements:

Functional Requirements:

The system must:

- Perform calls via a phone in the booth.
- Accept money from users via a coin inserter to make calls.
- Show the amount of time to users they are allowed to speak on the number display.
- Provide a short series of short beeps when the called number is busy.
- Provide a long series of long beeps when the called number does not respond.
- Stop being available for use when the paid time expires.
- Return exchange if users pay for more time than they actually use.
- Refund the paid amount if a call failure occurs.

Non-Functional Requirements:

Usability Requirements:

The system should:

- Include a door that can be closed if the user prefers privacy.
- Be presented as a red booth with a blue “tube-type” phone inside, equipped with a coin inserter, coin exchanger and a small number display.
- Include concise instructions written on the wall inside the booth, explaining how to use it, provided in English.

Reliability Requirements:

The system should:

- Be available 24/7.
- Have a call failure rate of no more than 1 in 10 attempts.
- Operate without failures during ongoing calls.
- Should be weatherproof

Performance Requirements:

The system should:

- Use 5–6 watts of electrical power per hour in active mode.
- Use 1-2 watts of electrical power per hour in passive mode.
- Maintain a ping time of no more than 100 milliseconds.

Supportability Requirements:

The system should:

- Be presented "as is," with no variations or customization options.
-
-

Use Cases:

Title: Make a Call

Primary Actor: User

Success Scenario:

1. The user enters the booth.
 2. The user inserts coins into the coin inserter.
 3. The user picks up the handset.
 4. The user dials a phone number.
 5. The user waits for the call to connect.
 6. The user speaks until the paid time expires or ends the call when satisfied.
-

Title: Pay Money

Primary Actor: User

Success Scenario:

1. The user enters the booth.
 2. The user inserts coins into the coin inserter.
 3. The system displays the allowed speaking time in minutes on the number display.
-

Title: Check Allowed Speaking Time on Display

Primary Actor: User

Success Scenario:

1. The user enters the booth.
 2. The user inserts coins into the coin inserter.
 3. The user sees the allowed speaking time in minutes on the number display.
 4. The user picks up the handset.
 5. The display updates to show the remaining speaking time, decrementing each minute while the phone is in use.
-

Title: Attempt to Connect to a Number

Primary Actor: User

Success Scenario:

1. The user enters the booth.
 2. The user inserts coins into the coin inserter.
 3. The user picks up the handset.
 4. The user dials a phone number.
 5. The user hears:
 - A short series of beeps if the number is busy.
 - A long series of beeps if the number does not respond.
 6. If the call connects, the user can speak.
-

Title: Disconnect Call When Paid Time Expires

Primary Actor: User

Success Scenario:

1. The user enters the booth.
 2. The user inserts coins into the coin inserter.
 3. The user picks up the handset.
 4. The user dials a phone number.
 5. The call connects, and the user speaks.
 6. The system disconnects the call when the paid time expires.
-

Title: Return Change

Primary Actor: Phone

Success Scenario:

1. The user enters the booth.
 2. The user inserts coins into the coin inserter.
 3. The user picks up the handset.
 4. The user dials a phone number.
 5. The call connects, and the user speaks.
 6. The user ends the call before using all the paid time.
 7. The system returns the remaining balance in coins through the coin return slot.
-

Title: Refund in Case of Failure

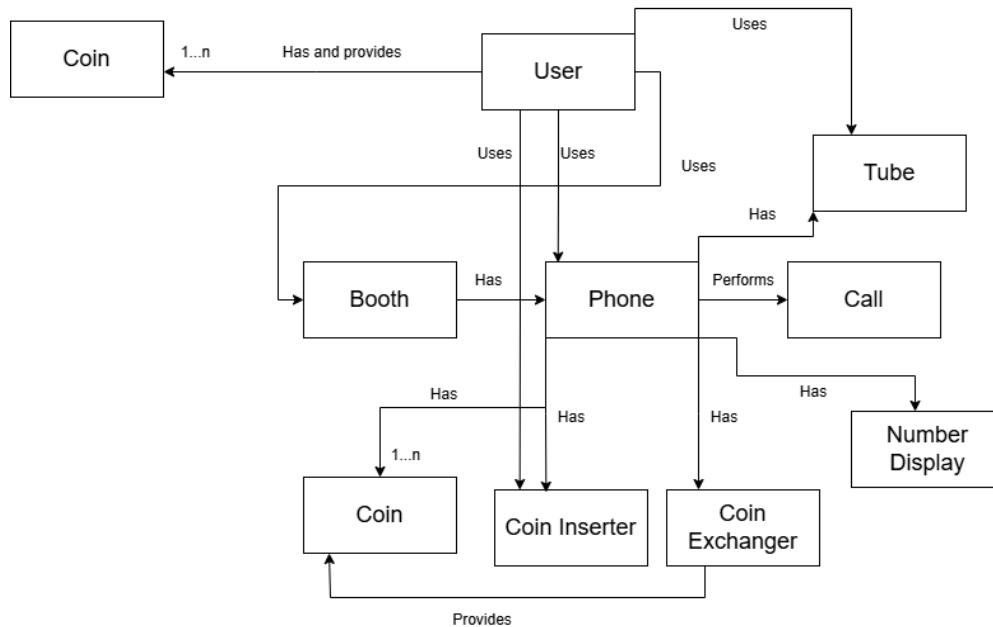
Primary Actor: Phone

Success Scenario:

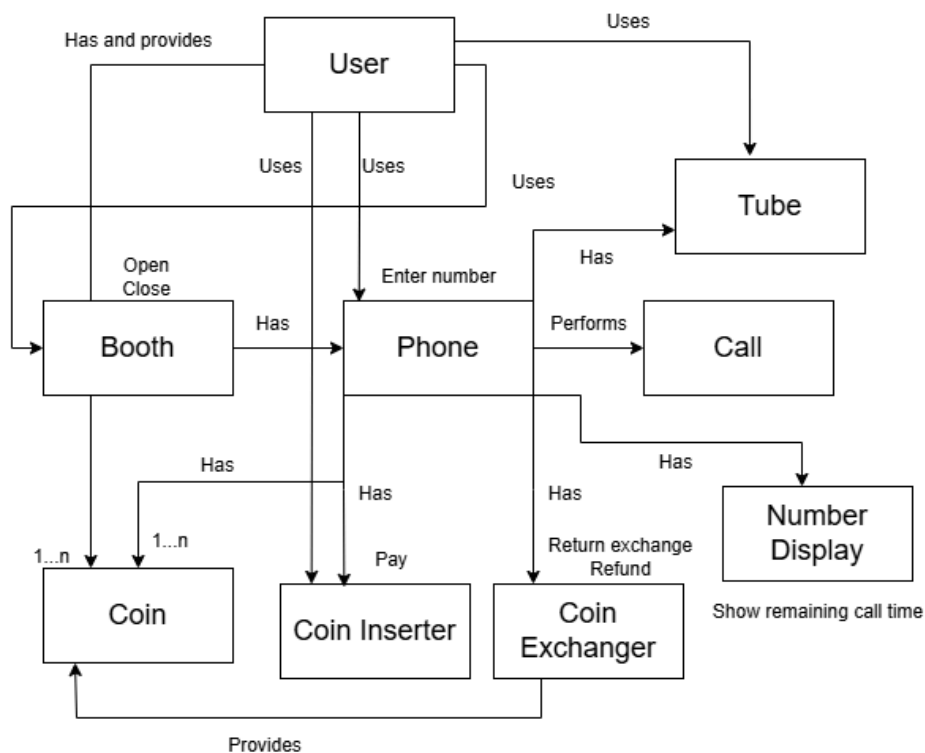
1. The user enters the booth.
2. The user inserts coins into the coin inserter.
3. The user picks up the handset.
4. The user dials a phone number.
5. The call fails to connect due to an error.

6. The system refunds the paid amount in coins through the coin return slot.

Objects and their relationships:



Classes and their relationships:



The key difference lies in how the "Coins" entity is represented. When discussing **classes**, "Coin" is treated as an abstract entity referenced by the Phone, Coin Exchanger, and User. However, when discussing **objects**, we differentiate between the User's coins and the Phone's coins, as these are distinct instances of the abstract "Coin" entity.

For all other classes, they are designed to be instantiated as single entities, meaning the remaining "objects" directly correspond to their respective "classes."

Class diagrams picturing classes, their attributes, and relations in the system:

