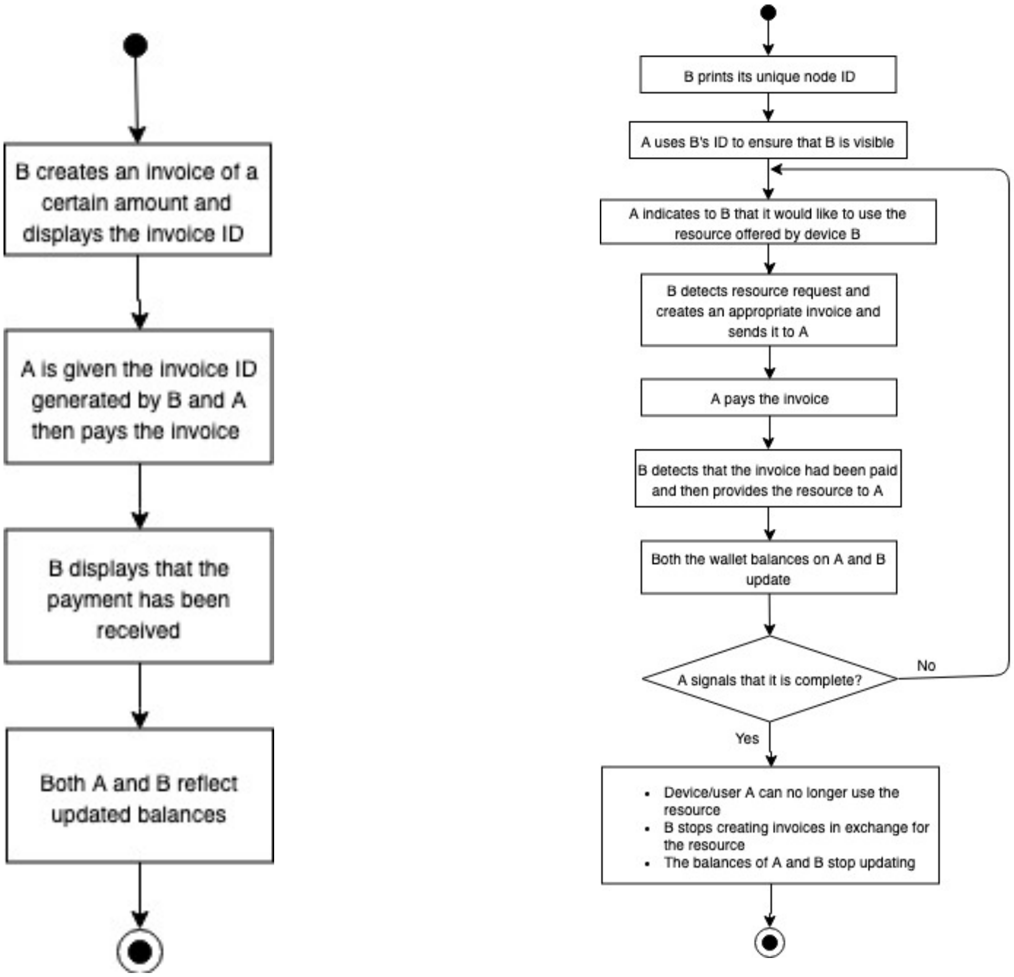
## Implementation development plan:

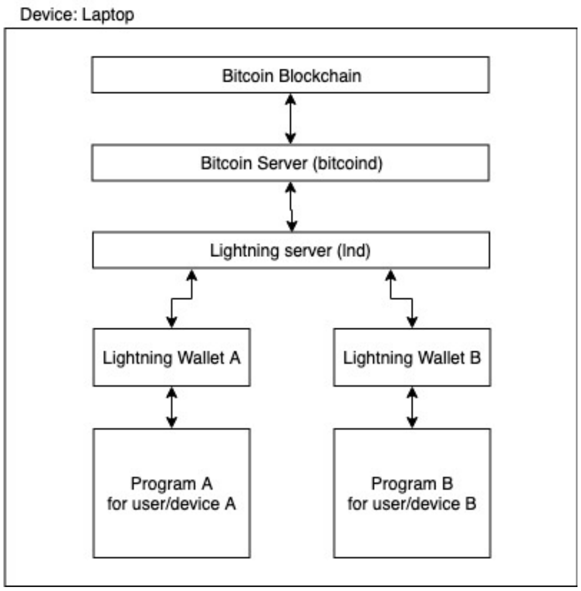
In all the steps of implementation, there will be 2 entities that will need to communicate with each other through the Lightning Network. The entities will be called A and B.

For all the steps the aim will be to ensure that the two entities, A and B, can communicate in the correct way. The two activity diagrams shown below in figure x outline the scenarios that should be tested at each step in the development. The activity diagram in figure x.1 shows the most basic form of transaction that should happen between entity A and B in which B will produce an invoice that A will then pay. The activity diagram shown in figure x.2 shows a more complicated scenario where after having established an initial connection, entity A can continue to use a resource provided by B and be charged for the resource in micropayments.



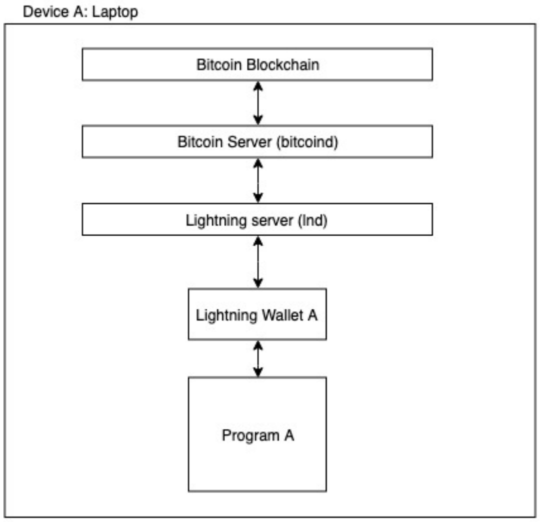
B creates an invoice of a certain amount and displays the invoice ID
A is given the invoice ID generated by B and A then pays the invoice
B displays that the payment has been received
Both A and B reflect updated balances
B prints its unique node ID
A uses B's ID to ensure that B is visible
A indicates to B that it would like to use the resource offered by device B
B detects resource request and creates an appropriate invoice and sends it to A
A pays the invoice
B detects that the invoice had been paid and then provides the resource to A
Both the wallet balances on A and B update
A signals that it is complete?
No
Yes
Device/user A can no longer use the resource
B stops creating invoices in exchange for the resource
The balances of A and B stop updating

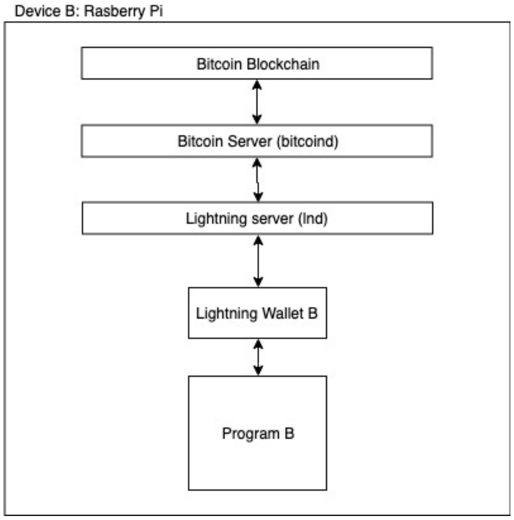
Step 1:



1.1 Download the Bitcoin blockchain onto a laptop computer
1.2 Install and set up a Bitcoin server (bitcoind) and a Lightning Server (lnd)
1.3 Create two separate Lightning wallets and instances so that it is as if two different devices on the Lightning Network exist.
1.4  Implement and test two programs that recreate the scenarios shown in the activity diagrams in figure x.

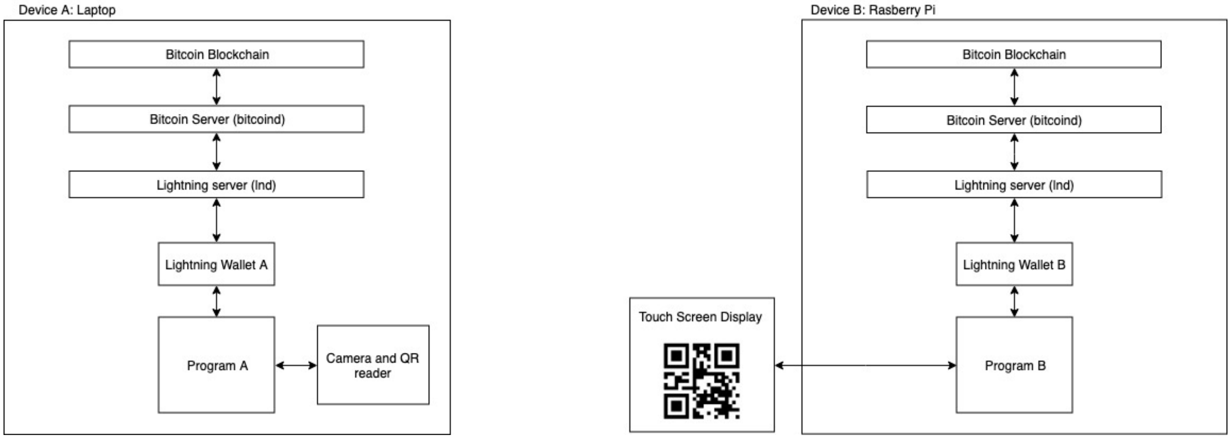
Step 2:





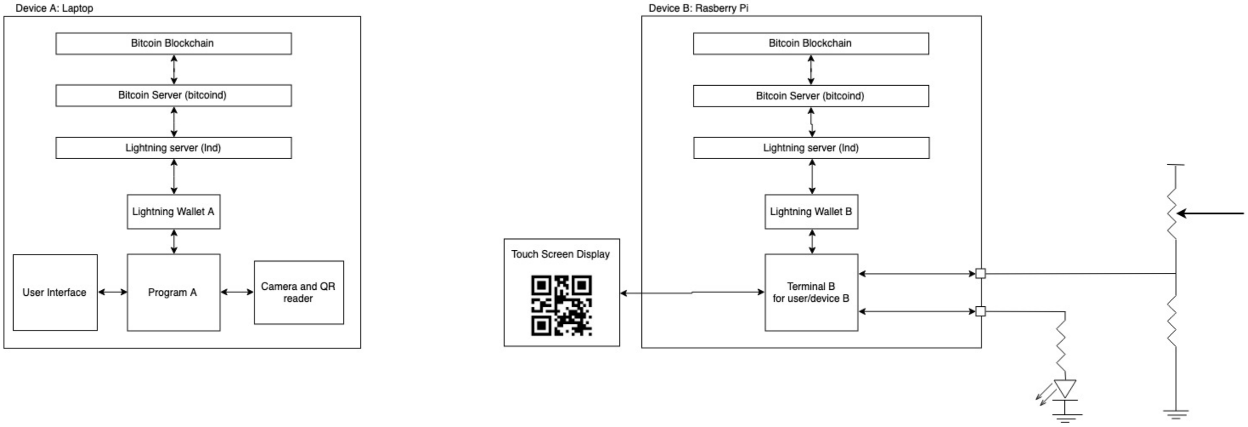
2.1 Set up Bitcoin and Lightning on a Raspberry Pi device
2.2 Implement and test two programs that recreate the scenarios shown in the activity diagrams in figure x. (This will require manually informing device A of the Lightning node ID of device B.

Step 3:



3.1 Integrate a camera and QR reader into system A
3.2 Integrate a display and QR presenter into system B
3.3 Implement and test two programs that recreate the scenarios shown in the activity diagrams in figure x but with the following developments:
- B displays its invoice/ID info on the QR display
- A gets B's invoice/ID data from its camera and QR decoder

Step 4:



4.1 Add details such as a user interface for both A and B
4.2 Add a potentiometer and configure its input to represent the amount of the resource that A is using and ensure that B creates invoices with amounts based on the potentiometer reading.
4.3 Add an LED as a way for B to show that its resource is or is not currently available to A.