# Tổng quan

# Các thành phần của mô hình

## Handler

* Vai trò:
* Chọn ngẫu nhiên 1 node làm proposer trong khoảng thời gian TIME\_SLOT (cụ thể trong chương trình là 60s) và broadcast cho tất cả các node.
* Phát sinh ngẫu nhiên các transaction (Không cần kiểm tra tính hợp lệ khi phát sinh các giao dịch) trong khoảng thời gian TIME\_GENERATE (cụ thể trong chương trình là 10s) và broadcast cho tất cả các node

## Node

* Vai trò:
* Lưu Blockchain: chuỗi các block (*BlockChainNode*)
* Đóng các transaction vào block (nếu là proposer)
* Nhận các transaction được broadcast từ handler, lưu giữ các transaction này và được dùng để đóng block (*poolTransaction*)
* Class **Transaction**:
* Thuộc tính:
* *Sender*: Tài khoản gửi
* *Receiver*: Tài khoản nhận
* *Amount*: Số tiền giao dịch
* *Id*: Chỉ số dùng để phân biệt giữa các giao dịch
* *TxHash*: Hash của transaction dùng để đảm bảo tính toàn vẹn thông tin dịch (Sử dụng MD5)
* Phương thức:
* getID(): Trả về id của Transaction
* verifyTransaction(): Kiểm tra tính toàn vẹn của giao dịch (Các thông tin về sender, receiver, amount, id dựa trên TxHash) và số tiền giao dịch hợp lệ (từ số dư của tài khoản)
* updateBalance(): Cập nhật số dư khi thực hiện Transaction
* Class **Block**:
* Thuộc tính:
* *prevHash*: Hash của block trước đó
* *txs*: Chứa danh sách các giao dịch của block
* *ID*: Node tạo block
* *Height*: Thứ tự (chiều cao) của block
* Phương thức:
* Owner(): Trả về NodeID tạo block
* getHeight(): Trả về height của block
* getPrevHash(): Trả về hash của block trước đó
* VerifyBlock(): Kiểm tra tính hợp lệ của Block (Transaction/prevHash và số transaction trong 1 block hợp lệ)
* Class **BlockChain**:
* Thuộc tính:
* Blk: Danh sách block
* Height: Chiều cao hiện tại của BlockChain
* lastHash: Hash block mới nhất của Blockchain
* Phương thức:
* update(): Gắn block vào blockchain
* getHeight(): trả về chiều cao hiện tại của BlockChain
* getLastHash(): trả về hash của block cuối cùng
* getLastBlock(): Trả về block mới nhất của BlockChain

## Account

* Dùng để lưu danh sách tài khoản và số dư của tài khoản (*accounts*: Kiểu từ điển)

# Cơ chế hoạt động của mô hình

## Chọn Proposer

* Handler:
* Chọn ngẫu nhiên 1 node làm proposer và broadcast cho tất cả các node
* Sau đó sẽ nghỉ trong khoảng thời gian TIMESLOT  
  A screenshot of a computer

  Description automatically generated with medium confidence
* Đối với node hiện thị các Block:
* Ghi lại sự kiện Block mới được tạo (nếu có)
* Đối với tất cả các node:
* Cập nhật ProposerID
* Nếu node còn đang mining thì dừng lại
* Nếu node là ProposerID: bắt đầu mining (đóng các transaction vào block, tối đa là 5 transaction)

## Phát sinh các giao dịch

* Handler:
* Sinh ngẫu nhiên giao dịch giữa và broadcast cho tất cả các node
* Mỗi giao dịch sẽ được tạo trong khoảng thời gian TIMEGENERATE  
  Text

  Description automatically generated
* Node:
* Nhận và thêm transaction này vào pool các transaction, và dùng các transaction này để xác thực khi có block được tạo hoặc dùng để đóng block

## Tạo Block

## Xác thực Block

## Cập nhật Block

## Gửi danh sách vote cho Node hiện thị block

* Dùng để cập nhật danh sách số lượng node vote nếu block được tạo thành công

## Update BlockChain của node

* Câp nhật Block mới vào BlockChain hiện tại của node
* Loại bỏ các transaction chứa trong Block
* Cập nhật lại số dư của các tài khoản sau khi giao dịch
* Lưu lại lịch sử trạng thái ra màn hình và đánh dấu Block mới được tạo vào NodeID=1 (Node này sẽ hiển thị các block được tạo)

# Đánh giá

# Tài liệu tham khảo