Date: 13-01-2022
Student: Rob Honig

Course: CP108 Plutus/Haskell I

Problem statement: Most African governments are trying different measures to eliminate *corruption*, *bad governance*, *mismanagement* and *lack of accountability* in their countries but these efforts are mostly unsuccessful. As a tool for change, *blockchain* can help in solving some of these governance issues plaguing Africa.

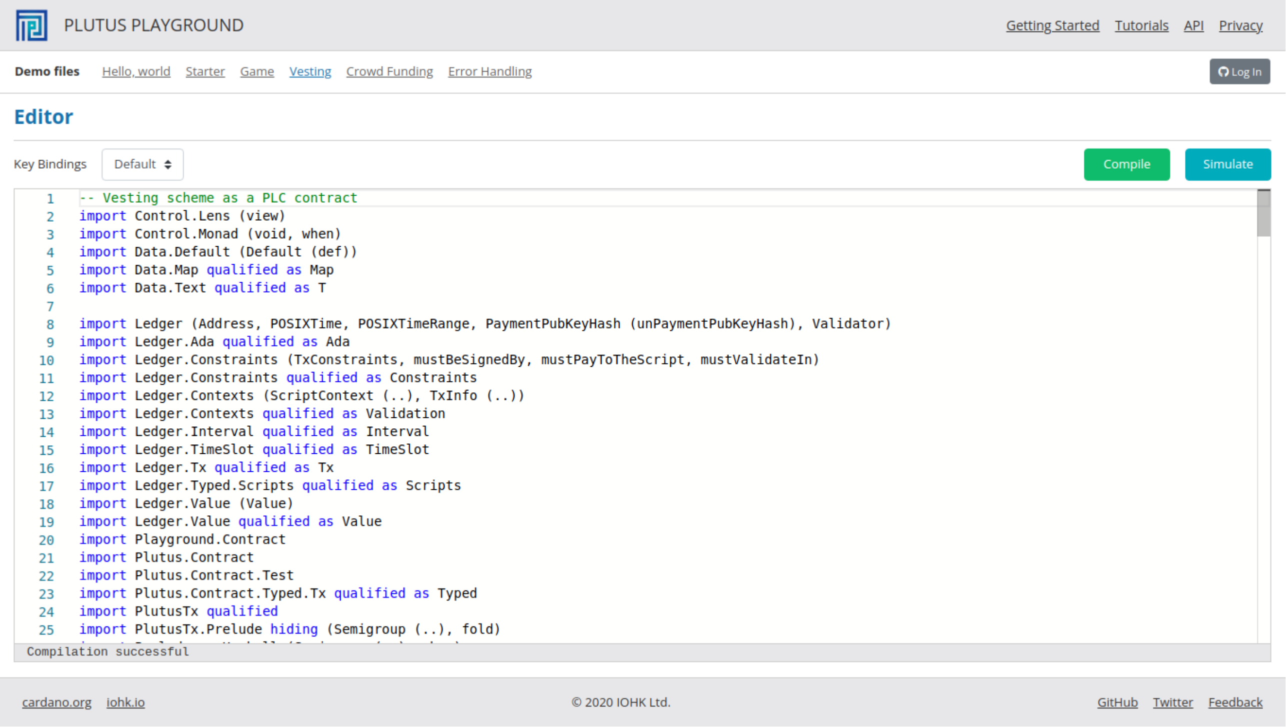
Task: With this in mind, think of a way you can implement a smart contract that can help eliminate any of these ills, then implement it using Plutus.

Possible solution: Giving acces to funds without using the so called "Middleman". We use "Smart Contracts" for (financial) aid or microcredits, powered by the Cardano Blockchain. Lock Ada funds on the blockchain, redeem funds with the correct redeemer.

In this way no intermediary is used. This keeps human error to a minimum. Based on predetermined rules written in Haskell and Plutus.

- No middlemen involved;
- Low transaction costs;
- 24/7 available;
- Safe, secure and fast

Script: The ***Gift.hs*** can be used for this exam and can be tested at the Plutus Playground;
https://playground.plutus.iohkdev.io/



```haskell
-- Vesting scheme as a PLC contract
import Control.Lens (view)
import Control.Monad (void, when)
import Data.Default (Default (def))
import Data.Map qualified as Map
import Data.Text qualified as T

import Ledger (Address, POSIXTime, POSIXTimeRange, PaymentPubKeyHash (unPaymentPubKeyHash), Validator)
import Ledger.Ada qualified as Ada
import Ledger.Constraints (TxConstraints, mustBeSignedBy, mustPayToTheScript, mustValidateIn)
import Ledger.Constraints qualified as Constraints
import Ledger.Contexts (ScriptContext (..), TxInfo (..))
import Ledger.Contexts qualified as Validation
import Ledger.Interval qualified as Interval
import Ledger.TimeSlot qualified as TimeSlot
import Ledger.Tx qualified as Tx
import Ledger.Typed.Scripts qualified as Scripts
import Ledger.Value (Value)
import Ledger.Value qualified as Value
import Playground.Contract
import Plutus.Contract
import Plutus.Contract.Test
import Plutus.Contract.Typed.Tx qualified as Typed
import PlutusTx qualified
import PlutusTx.Prelude hiding (Semigroup (..), fold)
```

```
Compilation successful
```

Language: Haskell and Plutus