JackeMoney System

Introduction

JackeMoney is a financial institution that mainly deals with transactions and transfer of money from one customer to another, and works under the regulations of the telecom companies MTN and Warid / Airtel. Offering transfer of money with customers by either committing deposits or making cash withdraws from the attendants whose use a command line application at their kiosks to record the transactions to the database whenever a transaction is performed.

Activities

Kiosk Attendant

The kiosk attendant is able to perform these actions with the command line application

* **Committing transactions**, the kiosk attendant can record up to three transactions at a time which can either be withdraws or deposits for the customers they service.
* **Viewing float,** the kiosk attendant can see another attendant’s float and selects who has more float than he or she has and sends a request to that attendant
* **Requesting,** attendants / agents can request another attendant to perform a transaction on their behalf given that they have more float than he or she does by providing the customer number and the amount to the desired agent.
* **Viewing requests**, kiosk attendants can see who requested the to perform transactions on their behalf in case the requesting attendant does not have enough float on their simcards
* **Servicing requests**, after the attendant has seen the requests he or she has can then go ahead to service the request using the customer number, amount, request id and after he or she has serviced the request it is recorded as a normal deposit transaction in the repository.

Administrator (Jack in this case)

The administrator Jack is able to perform the common system administrative activities using the web interface without leaving;

* **Adding agents**, this requires jack to add new users / agents, he gives them initial float for both service providers, the name of the kiosk they are to operate, the user name they are to use for logging in, both simcards they are to use for the transactions.
* **Paying commission**, when the agents make transactions they get a small amount that the service provider pays to them and accumulates as commission and jack can login to his web interface and decides to pay back the commission to the agent who has insufficient float for example.

The architecture

Client-Server architecture

JackeMoney system is built on a client server architecture where the client is the command line application (JackeMoneyAgent) that is installed for the kiosk attendants to send requests in form of transactions, requests for helping an agent on transactions and views for databases information for example float to the server (JackeMoneyServer) whose job is to interact with the database and retrieve to the client the intended results and also in case of errors it should be able to notify the client.

The Client (JackeMoneyAgent)

The attendant is expected to first login to the systems by providing the user name that was given to him or her by the administrator jack and connects to the server then he or she can type in recognized commands in the command line window.

JackeMoneyAgent accepts only 5 commands, these are the only activities aa agent can do with his command line application and are summarized in the below.

**commit command**

***commit deposit*** *customer\_Number1 amount1 agent\_number****,withdraw*** *customer\_Number2**amount2 agent\_number****, deposit*** *customer\_Number3 amount3 agent\_number*

and enables the agent to send the sales information to the database

this command expects the attendant to enter not less than two transactions and not more than three transactions

example

***commit deposit*** *0789141512 100000 0772457812****,withdraw*** *0753141415**14000 0752457812*

the above command means the agent is recording two transactions a an **Mtn** deposit of 100000 on customer number 0789141512 using agent number 0772457812 and a **Warid** withdraw of 14000 from customer number 0753141415 using agent number 0752457812.

**view requests command**

***view requests***

This command enables the agent to see the requests or transfers that are directed to him or her from other agents who did not have enough float to service the customers. It returns the requests of the logged in agent with their request ids.

**service request command**

***service request*** *request\_id customer\_Number* *amount**agent\_Number*

With this command, the agent can service the request using the request id, the amount, customer number and his or her agent number and upon submitting this command such a pending request is now recorded as a served transaction and is recorded in the repository

Example

***service request*** *1 0788127845 10000 0786325898*

This means the agent is servicing a request with request id *1* and records this transaction as a deposit of *10000* to customer number *0788127845* using her or his agent number *0786325898*

**view all floats command**

***view all floats*** *service\_Provider*

This command enables the agent to check out which other agent has more float for the specific service provider either MTN or Warid than he or she has before she can request as she or he must make sure that the agent requested is having enough float to service the request.

Example

***view all floats*** *warid*

This will return the warid float for other agents then it is upon the agent to choose the agent of his or her choice with enough warid float to service the request.

**request command**

***request*** *user\_Name customer\_Number amount*

here the agent requests another agent whose user name is specified in the command ***reques***t to perform the transaction on his or her behalf having seen that the other agent has enough float to service the request.

Example

***request*** *kato 0789812356 45000*

this means the agent is sending another agent *kato* in this case a request to help him deposit *45000* on customer number *0789812356*

**logout command**

***logout***

this command logs out the attendant and closes the whole application