

Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo de projeto. *Aquiel Braunschweig*
 Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo de projeto. *João Ferreira*

NetworkManager
-_network: Network
+ load (filename: String): void
+ save(): void
+ saveAs (filename: String): void
+ importFile (filename: String): void
+ exportFile (filename: String): void
+ saveFile (filename: String): void

Network
- «final» serialVersion UID: long
- clients: List<Client>
- terminals: List<Terminal>
- communication: List<Communication>
+ importFile (name: String): void
+ disableClientNotifications (key: String): void
+ enableClientNotifications (key: String): void
+ registerClient (key: String): void
+ showAllClients(): List<String>
+ showClient (key: String): String
+ showClientPaymentsAndDebt (key: String): String
+ showAllCommunications (key: String): List<String>
+ showClientsWithDebt(): List<String>
+ showClientsWithoutDebt(): List<String>
+ showCommunicationFromClient (key: String): List<String>
+ showCommunicationToClient (key: String): List<String>
+ showTerminalsWithPositiveBalance(): List<String>
+ showUnusedTerminals(): List<String>
+ showGlobalBalance(): String
+ openManualTerminalConsole (terminalKey: String): void
+ registerTerminal (terminalKey: String, terminalType: String, clientKey: String): void
+ showAllTerminals(): String

«abstract» TerminalState
#- terminal: Terminal
+ «abstract» canSendTextCommunication(): boolean
+ «abstract» canReceiveTextCommunication(): boolean
+ «abstract» canStartInteractiveCommunication(): boolean
+ «abstract» canReceiveInteractiveCommunication(): boolean
+ «abstract» becomeIdle(): void
+ «abstract» becomeSilent(): void
+ «abstract» becomeBusy(): void
+ «abstract» turnOff(): void

«abstract» Tariff
- client: Client
#- textCommunications: int
#- videoCommunications: int
+ increaseTextCommunicationsCount(): void
+ increaseVideoCommunicationsCount(): void
+ resetCommunicationsCount(): void
+ «abstract» getTextCommunicationCost (charactersNumber: int): int
+ «abstract» getInteractiveCommunication (type: CommunicationType): int
+ «abstract» changePlan (clientBalance: double): void

Off
+ canSendTextCommunication(): boolean
+ canReceiveTextCommunication(): boolean
+ canStartInteractiveCommunication(): boolean
+ canReceiveInteractiveCommunication(): boolean
+ becomeIdle(): void
+ becomeSilent(): void
+ becomeBusy(): void
+ turnOff(): void

Silent
+ canSendTextCommunication(): boolean
+ canReceiveTextCommunication(): boolean
+ canStartInteractiveCommunication(): boolean
+ canReceiveInteractiveCommunication(): boolean
+ becomeIdle(): void
+ becomeSilent(): void
+ becomeBusy(): void
+ turnOff(): void

«interface» NotificationMechanismOfDelinquency
+ isDefault(): boolean
+ sendNotification(): void

RegisterOnApp
+ isDefault(): boolean
+ sendNotification(): void

PostCard
+ isDefault(): boolean
+ sendNotification(): void

SMS
+ isDefault(): boolean
+ sendNotification(): void

Email
+ isDefault(): boolean
+ sendNotification(): void

Busy
+ canSendTextCommunication(): boolean
+ canReceiveTextCommunication(): boolean
+ canStartInteractiveCommunication(): boolean
+ canReceiveInteractiveCommunication(): boolean
+ becomeIdle(): void
+ becomeSilent(): void
+ becomeBusy(): void
+ turnOff(): void

Idle
+ canSendTextCommunication(): boolean
+ canReceiveTextCommunication(): boolean
+ canStartInteractiveCommunication(): boolean
+ canReceiveInteractiveCommunication(): boolean
+ becomeIdle(): void
+ becomeSilent(): void
+ becomeBusy(): void
+ turnOff(): void

PlatinumPlan

```
+ getTestCommunicationCost(characterNumber: int): int
+ getInteractiveCommunicationCost(type: CommunicationType): type
+ changePlan(clientBalance: double): void
```

GoldPlan

```
+ getTestCommunicationCost(characterNumber: int): int
+ getInteractiveCommunicationCost(type: CommunicationType): type
+ changePlan(clientBalance: double): void
```

NormalPlan

```
+ getTestCommunicationCost(characterNumber: int): int
+ getInteractiveCommunicationCost(type: CommunicationType): type
+ changePlan(clientBalance: double): void
```

Notification

```
- << final >> serialVersionUID: long
- client: Client
- meansOfDelivery: MeansOfDelivery
- type: NotificationType
+ isRequestedOnApp(): boolean
```

« interface » Serializable

```
TestCommunication
- messageSent: String
+ getCost(tariff: Tariff): void
```

« enumeration » NotificationType

OAS
OBI
SBI
BBI

« enumeration » CommunicationStatus

ONGOING
ENDED

InteractiveCommunication

```
- duration: int
- isVideoCommunication: boolean
+ setCost(tariff: Tariff): void
```

Terminal

```
- << final >> serialVersionUID: long
- key: String
- type: TerminalType
- state: TerminalState
- communications: List<Communication>
```

```
- payments: double
- debts: double
- friends: List<Terminal>
- notificationsToBeDelivered: List<Notification>
- communicationOngoing: InteractiveCommunication
```

```
+ canEndCurrentCommunication(): boolean
+ canStartCommunication(): boolean
+ addFriend(friend: Terminal): void
+ endInteractiveCommunication(duration: int): void
+ performPayment(communicationKey: int): void
+ removeFriend(terminalKey: String): void
+ sendTestCommunication(terminalToKey: String, message: String): void
+ showOngoingCommunication(): String
+ showTerminalBalance(): String
+ idleTerminal(): void
+ silenceTerminal(): void
+ startInteractiveCommunication(terminalToKey: String, communicationType: String): void
+ turnOffTerminal(): void
+ turnOnTerminal(): void
```

« abstract »

Communication

```
- << final >> serialVersionUID: long
- key: int
- terminalFrom: Terminal
- terminalTo: Terminal
- status: CommunicationStatus
- cost: int
+ setCost(tariff: Tariff): void
```

Client

```
- << final >> serialVersionUID: long
- key: String
- name: String
- rif: int
- tariff: Tariff
- terminals: List<Terminal>
- payments: double
- debts: double
- canReceiveFailedCommunications: boolean
+ getBalance(): int
+ addTerminal(terminal: Terminal): void
+ debtPaid(payment: int): void
+ addDebt(debt: int): void
+ enableFailedCommunicationsNotifications(): void
+ disableFailedCommunicationsNotifications(): void
+ receiveNotification(notification: Notification): void
+ clearNotifications(): void
```



