Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

**Τεχνολογία Λογισμικού**

Class Diagram - v0.1

Εικόνα που περιέχει κείμενο, υπογραφή, υπαίθριος

Περιγραφή που δημιουργήθηκε αυτόματα

**Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών & Πληροφορικής**

1 Ιουνίου 2022

**Ονόματα και ΑΜ των μελών της ομάδας:**

Ελένη Βασιλάκη, **ΑΜ:** 1070939, 4ο έτος

Βασίλης Κυριάκος, **ΑΜ:** 1070915, 4ο έτος

Κυριακή Λιούμη, **ΑΜ:** 1067410, 4ο έτος

Θεόδωρος Χατζημιχαήλ, **ΑΜ:** 1067538, 4ο έτος

**Editors:**

Βασίλης

Ελένη

Κυριακή

Θεόδωρος

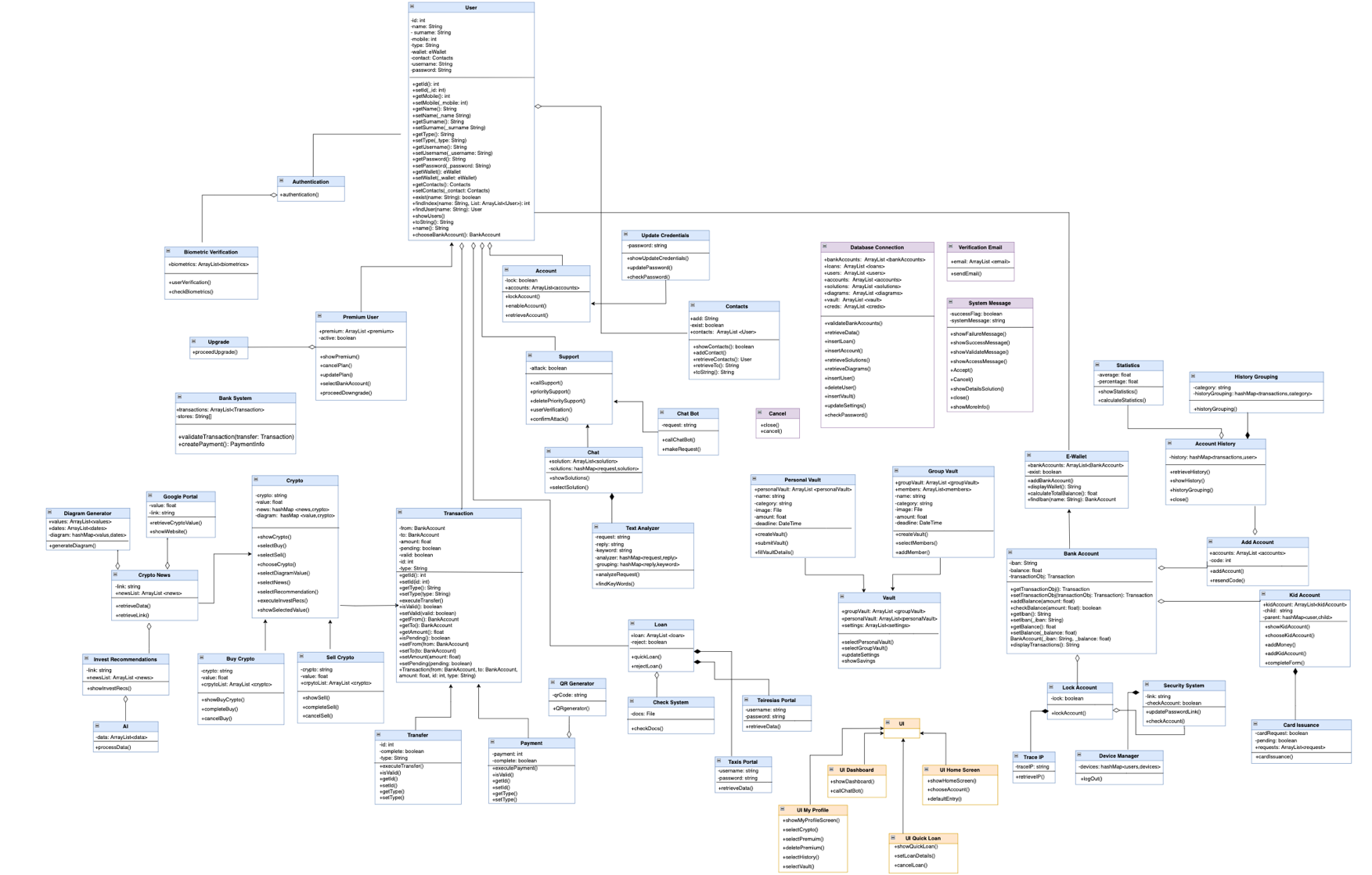
**Εργαλεία:**

Microsoft Word

draw.io

**GitHub:**

<https://github.com/VasilisKyriakos/financial-wallet>

**Class Diagram**

**Περιγραφή Κλάσεων**

1. **User (Χρήστης):** Οντότητα που περιέχει τα στοιχεία των χρηστών του συστήματος.

**Attributes:**

* -id: int
* -name: String
* -surname: String
* -mobile: int
* -type: String
* -wallet: eWallet
* -contact: Contacts
* -username: String
* -password: String

**Methods:**

* +getId(): int
* +setId(\_id: int)
* +getMobile(): int
* +setMobile(\_mobile: int)
* +getName(): String
* +setName(\_name String)
* +getSurname(): String
* +setSurname(\_surname String)
* +getType(): String
* +setType(\_type: String)
* +getUsername(): String
* +setUsername(\_username: String)
* +getPassword(): String
* +setPassword(\_password: String)
* +getWallet(): eWallet
* +setWallet(\_wallet: eWallet)
* +getContacts(): Contacts
* +setContacts(\_contact: Contacts)
* +exist(name: String): boolean
* +findIndex(name: String, List: ArrayList<User>): int
* +findUser(name: String): User
* +showUsers()
* +toString(): String
* +name(): String
* +chooseBankAccount(): BankAccount

1. **Premium User (Premium Χρήστης):** Ειδικότερη περίπτωση Χρήστη, που περιλαμβάνει χρήστες, οι οποίοι έχουν πραγματοποιήσει αναβάθμιση συνδρομής. Οι χρήστες αυτοί λαμβάνουν ειδικά προνόμια.

**Attributes:**

* +premium: ArrayList <premium>
* -active: Boolean

**Methods:**

* +showPremium()
* +cancelPlan()
* +updatePlan()
* +selectBankAccount()
* +proceedDowngrade()

1. **Authentication (Επαλήθευση Χρήστη):** Οντότητα η οποία διαχειρίζεται την επαλήθευση χρηστών.

**Method:**

* +authentication()

1. **Biometric Verification (Βιομετρική Επαλήθευση):** Οντότητα η οποία διαχειρίζεται τα απαραίτητα συστήματα ταυτοποίησης μέσω βιομετρικών ελέγχων, των συσκευών.

**Attributes:**

* +biometrics: ArrayList<biometrics>

**Methods:**

* +userVerification()
* +checkBiometrics()

1. **Transaction (Συναλλαγές):** Η γενική οντότητα που αναφέρεται στις δυνατότητες συναλλαγών που προσφέρονται στον χρήστη.

**Attributes:**

* -from: BankAccount
* -to: BankAccount
* -amount: float
* -pending: boolean
* -valid: boolean
* -id: int
* -type: String

**Methods:**

* +getId(): int
* +setId(id: int)
* +getType(): String
* +setType(type: String)
* +executeTransfer()
* +isValid(): boolean
* +setValid(valid: boolean)
* +getFrom(): BankAccount
* +getTo(): BankAccount
* +getAmount(): float
* +isPending(): boolean
* +setFrom(from: BankAccount)
* +setTo(to: BankAccount)
* +setAmount(amount: float)
* +setPending(pending: boolean)
* +Transaction(from: BankAccount, to: BankAccount,
* amount: float, id: int, type: String)

1. **Payment (Πληρωμή):** Ειδική περίπτωση Συναλλαγής, η οποία διαχειρίζεται τις διαδικασίες πληρωμής των χρηστών.

**Attributes:**

* -payment: int
* -complete: boolean

**Methods:**

* +executePayment()
* +isValid()
* +getId()
* +setId()
* +getType()
* +setType()

1. **Transfer (Μεταφορά):** Ειδική περίπτωση Συναλλαγής, η οποία διαχειρίζεται τις διαδικασίες μεταφοράς των χρηστών.

**Attributes:**

* -id: int
* -complete: boolean
* -type: String

**Methods:**

* +executeTransfer()
* +isValid()
* +getId()
* +setId()
* +getType()
* +setType()

1. **QR Generator (Δημιουργία QR):** Οντότητα δημιουργίας μοναδικού αναγνωριστικού κωδικού QR για την ολοκλήρωση πληρωμών.

**Attribute:**

* -qrCode: string

**Method:**

* +QRgenerator()

1. **Crypto (Κρυπτονομίσματα):** Γενική οντότητα υποστήριξης e-Wallet κρυπτονομισμάτων.

**Attributes:**

* -crypto: string
* -value: float
* -news: hashMap <news,crypto>
* -diagram: hasMap <value,crypto>

**Methods:**

* +showCrypto()
* +selectBuy()
* +selectSell()
* +chooseCrypto()
* +selectDiagramValue()
* +selectNews()
* +selectRecommendation()
* +executeInvestRecs()
* +showSelectedValue()

1. **Buy Crypto (Αγορά Κρυπτονομίσματος):** Ειδικότερη οντότητα υποστήριξης αγορών μεταξύ κρυπτονομισμάτων.

**Attributes:**

* -crypto: string
* -value: float
* +crpytoList: ArrayList <crypto>

**Methods:**

* +showBuyCrypto()
* +completeBuy()
* +cancelBuy()

1. **Sell Crypto (Πώληση Κρυπτονομίσματος):** Ειδικότερη οντότητα υποστήριξης πληρωμών με χρήση κρυπτονομισμάτων.

**Attributes:**

* -crypto: string
* -value: float
* +crpytoList: ArrayList <crypto>

**Methods:**

* +showSell()
* +completeSell()
* +cancelSell()

1. **Crypto News (Επικαιρότητα Κρυπτονομισμάτων):** Οντότητα η οποία αναλαμβάνει την συλλογή δεδομένων αναφορικά με κάποιο προεπιλεγμένο κρυπτονόμισμα.

**Attributes:**

* -link: string
* +newsList: ArrayList <news>

**Methods**

* +retrieveData()
* +retrieveLink()

1. **Invest Recommendations (Επενδυτικές Προτάσεις):** Οντότητα η οποία αναλαμβάνει την παρουσίαση επενδυτικών προτάσεων. Η οντότητα αυτή περιλαμβάνει αλγορίθμους, οι οποίοι μέσω Machine Learning επεξεργάζονται τα Crypto News. Το σύστημα αυτό συμβουλευόμενο τις προτιμήσεις του χρήστη εμφανίζει προτάσεις.

**Attributes:**

* -link: string
* +newsList: ArrayList <news>

**Methods:**

* +showInvestRecs()

1. **Contacts (Επαφές):** Οντότητα η οποία περιλαμβάνει τις καταχωρημένες επαφές του χρήστη.

**Attributes:**

* +add: String
* -exist: boolean
* +contacts: ArrayList <User>

**Methods:**

* +showContacts(): boolean
* +addContact()
* +retrieveContacts(): User
* +retrieveTo(): String
* +toString(): String

1. **Support (Τεχνική Υποστήριξη):** Οντότητα η οποία παρέχει υποστήριξη στον χρήστη.

**Attribute:**

* -attack: boolean

**Methods:**

* +callSupport()
* +prioritySupport()
* +deletePrioritySupport()
* +userVerification()
* +confirmAttack()

1. **Chat Bot:** Ειδικότερη οντότητα η οποία παρέχει υποστήριξη στον χρήστη μέσω συνομιλίας σε πραγματικό χρόνο. Η οντότητα αυτή διαχειρίζεται τα ερωτήματα του χρήστη, προτείνοντας τις αντίστοιχες λύσεις.

**Attribute:**

* -request: string

**Methods:**

* +callChatBot()
* +makeRequest()

1. **Loan (Δάνειο):** Οντότητα η οποία διαχειρίζεται την δανειοδότηση ενός χρήστη.

**Attributes:**

* +loan: ArrayList <loan>
* -reject: boolean

**Methods:**

* +quickLoan()
* +rejectLoan()

1. **Check System (Ελεγκτής):** Οντότητα η οποία ελέγχει τα δικαιολογητικά δανειοδότησης, απορρίπτοντας ή εγκρίνοντας την αίτηση.

**Attribute:**

* -docs: File

**Method:**

* +checkDocs()

1. **E-Wallet (Ψηφιακό Πορτοφόλι):** Γενική οντότητα η οποία περιλαμβάνει το σύνολο των διασυνδεδεμένων τραπεζικών λογαριασμών του χρήστη.

**Attributes**

* +bankAccounts: ArrayList<BankAccount>
* -exist: Boolean

**Methods:**

* +addBankAccount()
* +displayWallet(): String
* +calculateTotalBalance(): float
* +findIban(name: String): BankAccount

1. **Bank Account (Τραπεζικός Λογαριασμός):** Ειδικότερηοντότητα η οποία περιλαμβάνει τα στοιχεία ενός τραπεζικού λογαριασμού.

**Attributes:**

* -iban: String
* -balance: float
* -transactionObj: Transaction

**Methods:**

* +getTransactionObj(): Transaction
* +setTransactionObj(transactionObj: Transaction): Transaction
* +addBalance(amount: float)
* +checkBalance(amount: float): boolean
* +getIban(): String
* +setIban(\_iban: String)
* +getBalance(): float
* +setBalance(\_balance: float)
* BankAccount(\_iban: String, \_balance: float)
* +displayTransactions(): String

1. **Vault (Σύστημα Αποταμίευσης):** Γενικότερη οντότητα η οποία διαχειρίζεται την δημιουργία ομαδικού ή προσωπικού χρηματοκιβωτίου.

**Attributes:**

* +groupVault: ArrayList <groupVault>
* +personalVault: ArrayList<personalVault>
* +settings: ArrayList<settings>

**Methods:**

* +selectPersonalVault()
* +selectGroupVault()
* +updateSettings
* +showSavings

1. **Account History (Ιστορικό Λογαριασμού):** Οντότητα η οποία διαχειρίζεται το ιστορικό συναλλαγών ανά τραπεζικό λογαριασμό.

**Attribute:**

* -history: hashMap<transactions,user>

**Methods:**

* +retrieveHistory()
* +showHistory()
* +historyGrouping()
* +close()

1. **History Grouping (Κατηγοριοποίηση Ιστορικού):** Ειδικότερη Οντότητα Ιστορικού, η οποία επεξεργάζεται και κατηγοριοποιεί τις συναλλαγές του χρήστη.

**Attributes:**

* -category: string
* -historyGrouping: hashMap<transactions,category>

**Method:**

* +historyGrouping()

1. **Statistics (Στατιστικά):** Ειδικότερη οντότητα η οποία περιλαμβάνει την παρουσίαση στατιστικών, έπειτα από επεξεργασία των συναλλαγών.

**Attributes:**

* -average: float
* -percentage: float

**Methods:**

* +showStatistics()
* +calculateStatistics()

1. **Lock Account (Κλείδωμα Λογαριασμού):** Οντότητα η οποία διαχειρίζεται την απενεργοποίηση του λογαριασμού σε περιπτώσεις παραβίασης.

**Attribute:**

* -lock: boolean

**Method:**

* +lockAccount()

1. **Kid Account (Παιδικός Λογαριασμός):** Ειδικότερη οντότητα τραπεζικού λογαριασμού, η οποία επιτρέπει την δημιουργία παιδικού λογαριασμού.

**Attributes:**

* +kidAccount: ArrayList<kidAccount>
* -child: string
* -parent: hashMap<user,child>

**Methods:**

* +showKidAccount()
* +chooseKidAccount()
* +addMoney()
* +addKidAccount()
* +completeForm()

1. **Card Issuance (Έκδοση Κάρτας):** Ειδικότερη περίπτωση δυνατότητας έκδοσης κάρτας σε φυσική μορφή.

**Attributes:**

* -cardRequest: boolean
* -pending: boolean
* +requests: ArrayList<request>

**Method:**

* +cardIssuance()

1. **Database connection (Σύνδεση με Βάση):** Γενική οντότητα που παρέχει σύνδεση στην βάση δεδομένων για την φόρτωση των απαραίτητων πληροφοριών.

**Attributes:**

* +bankAccounts: ArrayList <bankAccounts>
* +loans: ArrayList <loans>
* +users: ArrayList <users>
* +accounts: ArrayList <accounts>
* +solutions: ArrayList <solutions>
* +diagrams: ArrayList <diagrams>
* +vault: ArrayList <vault>
* +creds: ArrayList <creds>

**Methods:**

* +validateBankAccounts()
* +retrieveData()
* +insertLoan()
* +insertAccount()
* +retrieveSolutions()
* +retrieveDiagrams()
* +insertUser()
* +deleteUser()
* +insertVault()
* +updateSettings()
* +checkPassword()

1. **Personal Vault (Προσωπικό Χρηματοκιβώτιο):** Ειδικότερη οντότητα η οποία διαχειρίζεται την δημιουργία προσωπικού χρηματοκιβωτίου.

**Attributes:**

* +personalVault: ArrayList <personalVault>
* -name: string
* -category: string
* -image: File
* -amount: float
* -deadline: DateTime

**Methods:**

* +createVault()
* +submitVault()
* +fillVaultDetails()

1. **Group Vault (Ομαδικό Χρηματοκιβώτιο):** Ειδικότερη οντότητα η οποία διαχειρίζεται την δημιουργία ομαδικού χρηματοκιβωτίου.

**Attributes:**

* +groupVault: ArrayList <groupVault>
* +members: ArrayList<members>
* -name: string
* -category: string
* -image: File
* -amount: float
* -deadline: DateTime

**Methods:**

* +createVault()
* +selectMembers()
* +addMember()

1. **Verification Email:** Οντότητα η οποία διαχειρίζεται την αποστολή email σε περιπτώσεις επαλήθευσης χρήστη.

**Attribute**

* +email: ArrayList <email>

**Method:**

* +sendEmail()

1. **Add Account (Προσθήκη Λογαριασμού):** Οντότητα η οποία διαχειρίζεται την εισαγωγή νέου τραπεζικού λογαριασμού από τον χρήστη.

**Attributes**

* +accounts: ArrayList <accounts>
* -code: int

**Methods:**

* +addAccount()
* +resendCode()

1. **Text Analyzer (Αναλυτής Κειμένου):** Οντότητα η οποία διαχειρίζεται τις ερωτήσεις του χρήστη της οντότητας Chatbot με σκοπό την κατανόηση και την υπόδειξη λύσης.

**Attributes:**

* -request: string
* -reply: string
* -keyword: string
* -analyzer: hashMap<request,reply>
* -grouping: hashMap<reply,keyword>

**Methods**

* +analyzeRequest()
* +findKeyWords()

1. **Cancel (Ακύρωση):** Οντότητα η οποία διαχειρίζεται την πραγματοποίηση ακύρωσης ενέργειας, όπου αυτό απαιτείται.

**Methods:**

* +close()  
  +cancel()

1. **Chat (Συνομιλία):** Οντότητα η οποία διαχειρίζεται την συνομιλία Χρήστη – Συστήματος στην κλάση Chat Bot.

**Attributes:**

* +solution: ArrayList<solution>
* -solutions: hashMap<request,solution>

**Methods:**

* +showSolutions()
* +selectSolution()

1. **Crypto News:** Οντότητα η οποία διαχειρίζεται τη σύνθεση ειδήσεων για κρυπτονομίσματα.

**Attributes:**

* -link: string
* +newsList: ArrayList <news>

**Methods:**

* +retrieveData()
* +retrieveLink()

1. **Google Portal:** Ειδικότερη οντότητα η οποία διαχειρίζεται την σύνδεση του συστήματος με το Google για την συλλογή απαραίτητων δεδομένων.

**Attributes:**

* -value: float
* -link: string

**Methods:**

* +retrieveCryptoValue()
* +showWebsite()

1. **Diagram Generator:** Οντότητα η οποία διαχειρίζεται την δημιουργία διαγραμμάτων διακύμανσης τιμής για κρυπτονομίσματα.

**Attributes:**

* +values: ArrayList<values>
* +dates: ArrayList<dates>
* -diagram: hashMap<valus,dates>

**Method:**

* +generateDiagram()

1. **ΑΙ:** Οντότητα η οποία διαχειρίζεται την δεδομένα και μέσω AI αλγορίθμων καταλήγει σε συμπεράσματα όπως για παράδειγμα την δημιουργία Επενδυτικών Προτάσεων της κλάσης Invest Recommendations.

**Attribute:**

* -data: ArrayList<data>

**Method:**

* +processData()

1. **Upgrade:** Ειδικότερη οντότητα η οποία διαχειρίζεται την αναβάθμιση συνδρομής ενός χρήστη σε Premium.

**Method:**

* +proceedUpgrade()

1. **Taxis Portal:** Ειδικότερη οντότητα η οποία αναλαμβάνει την αυτόματη σύνδεση ενός χρήστη στο σύστημα Taxis Net και την λήψη των απαραίτητων δικαιολογητικών για έγκριση δανειοδότησης.

**Attributes:**

* -username: string
* -password: string

**Method:**

* +retrieveData()

1. **Teiresias Portal:** Ειδικότερη οντότητα η οποία αναλαμβάνει την εξέταση οφειλών ενός χρήστη μέσω σύνδεσης στην πλατφόρμα Τειρεσίας η οποία είναι αρμόδια για τις οφειλές-δάνεια ενός χρήστη.

**Attributes:**

* -username: string
* -password: string

**Method:**

* +retrieveData()

1. **Trace IP:** Ειδικότερη οντότητα η οποία αναλαμβάνει την ανίχνευση ΙP διεύθυνσης του χρήστη.

**Attribute:**

* -traceIP: string

**Method:**

* +retrieveIP()

1. **Account:** Γενικότερη οντότητα η οποία διαχειρίζεται όλα τους συνδεδεμένους λογαριασμούς του συστήματος.

**Attributes:**

* -lock: boolean
* +accounts: ArrayList<accounts>

**Methods**

* +lockAccount()
* +enableAccount()
* +retrieveAccount()

1. **Device Manager:** Ειδικότερη οντότητα η οποία αποθηκεύει και διαχειρίζεται το πλήθος διασυνδεδεμένων συσκευών του κάθε χρήστη.

**Attribute:**

* -devices: hashMap<users,devices>

**Method:**

* +logOut()

1. **Update Credentials:** Ειδικότερη οντότητα η οποία διαχειρίζεται την ανανέωση των credentials ενός λογαριασμού.

**Attribute:**

* -password: string

**Methods:**

* +showUpdateCredentials()
* +updatePassword()
* +checkPassword()

1. **Security System:** Οντότητα η οποία διαχειρίζεται την διατήρηση της ασφάλειας στο σύστημά μας.

**Attributes:**

* -link: string
* -checkAccount: boolean

**Methods:**

|  |
| --- |
| * +updatePasswordLink() |
| * +checkAccount() |