

Class-Diagram-v1.0

parkRadar



Σύνθεση ομάδας

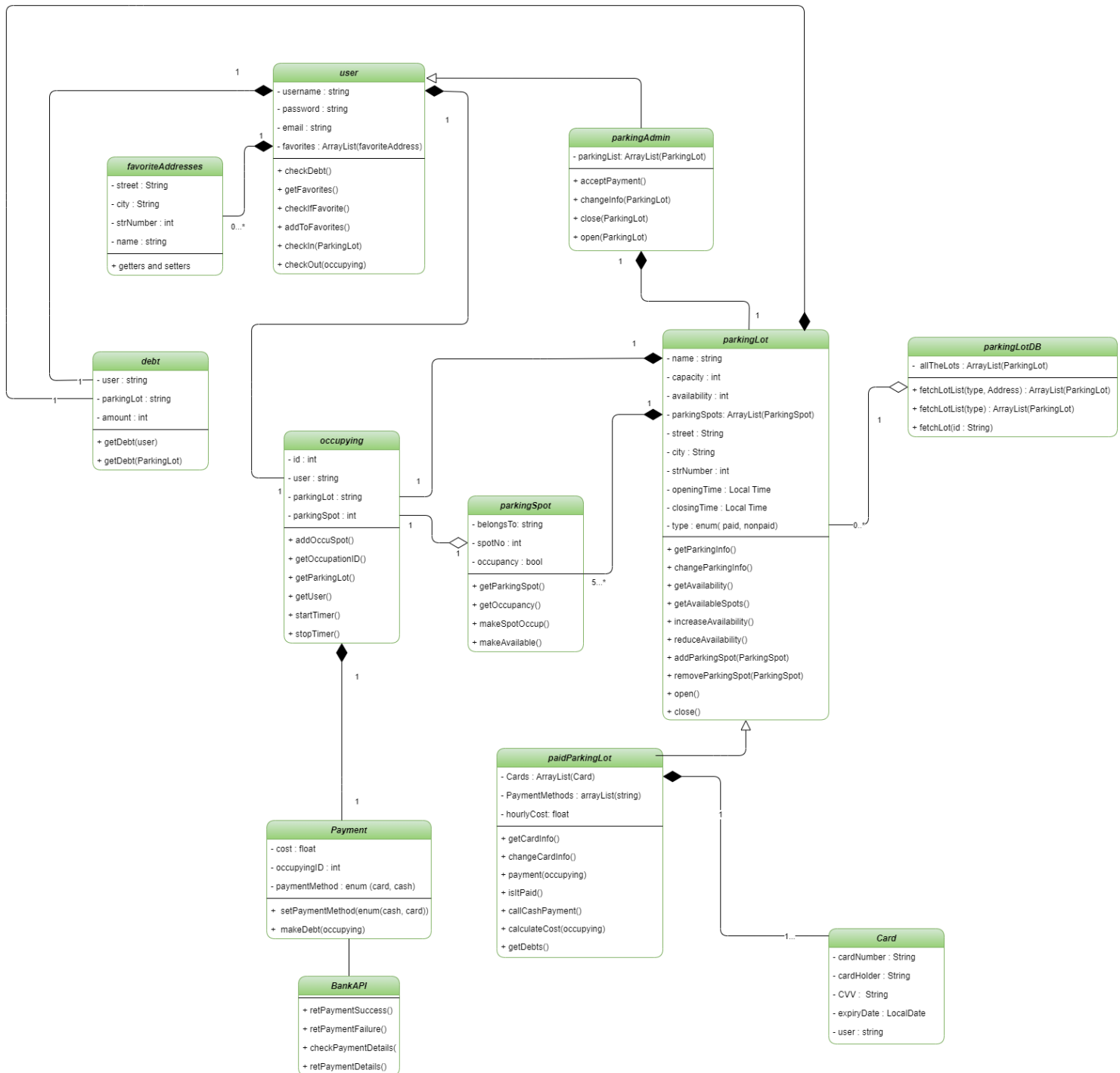
ΔΕΜΟΣ ΔΗΜΗΤΡΗΣ, 1051329, 8^ο ΕΤΟΣ

ΛΙΟΠΕΤΑ ΔΗΜΗΤΡΑ, 1054373, 7^ο ΕΤΟΣ

ΜΠΑΖΑΚΑΣ ΛΑΖΑΡΟΣ, 1054289, 7^ο ΕΤΟΣ

ΞΕΝΟΥ ΕΜΜΑΝΟΥΕΛΑ, 1054286, 7^ο ΕΤΟΣ

Class Diagram



Διάγραμμα 1 Class Diagram για την εφαρμογή ParkRadar

FULL VIEW

Επεξήγηση

parkingLot

+ close()

Κάνει το *parkingLot.availability* 0, δηλαδή κάνει το *parkingLot* μη διαθέσιμο για δέσμευση θέσης και στάθμευση. Η μέθοδος αυτή αντιστοιχίζεται με την μέθοδο *close(parkingLot)* της οντότητας *parkingAdmin*

+ open()

Αν το *parkingLot* είναι κλειστό/μη διαθέσιμο, η μέθοδος αυτή το επαναφέρει στα διαθέσιμα parking. Η μέθοδος αυτή αντιστοιχίζεται με την μέθοδο *open(parkingLot)* της οντότητας *parkingAdmin*.

PaidParkingLot

Θηγατρική οντότητα του *parkingLot*, αντιπροσωπεύει τα parking επί πληρωμή και αποθηκεύει τις επιπλέον πληροφορίες

Cards : *arrayList* όπου αποθηκεύονται αντικείμενα τύπου *Card*,

hourlyCost: κόστος δέσμευσης θέσης στο εν λόγω parking ανά ώρα.

PaymentMethods: οι διαθέσιμοι τρόποι πληρωμής στο εν λόγω parking.

+calculateCost(occupying)

Μέθοδος που υπολογίζει το κόστος για μια συγκεκριμένη δέσμευση θέσης (αντικείμενο τύπου *occupying*) στο εν λόγω *paidparkingLot*, βάση του χρόνου που το *parkingSpot* έμεινε δεσμευμένο.

+getDebts()

Μέθοδος που εμφανίζει όλα τα αντικείμενα τύπου *debt* για το εν λόγω *parkingLot*. Η μέθοδος αυτή αντιστοιχίζεται με την μέθοδο *getDebt(parkingLot)* της οντότητας *debt*.

Card

Οντότητα για να αποθηκεύονται οι κάρτες-υποδοχείς πληρωμής κάθε parking.

ParkingLotDB

Βάση Δεδομένων που διατηρεί όλα τα Parking. Χρήση της για την εύρεση Parking από τον χρήστη.

+FetchLotList(type, Address)

Ανακαλεί από την βάση Δεδομένων *ParkingLotDB* όλα τα parking συγκεκριμένου τύπου κοντά σε μια συγκεκριμένη διεύθυνση. (UseCase "Search Parking")

+FetchLotList(type)

Ανακαλεί από την βάση Δεδομένων *ParkingLotDB* όλα τα parking συγκεκριμένου τύπου (UseCase "View Parkings By Type")

+FetchLot(id: string)

Ανακαλεί από την βάση Δεδομένων *ParkingLotDB* ένα αντικείμενο *parkingLot* του οποίου το χαρακτηριστικό *name* αντιστοιχεί στο string εισόδου *id*.

Payment

Διατηρεί τις πληροφορίες της πληρωμής σε ένα parking, δηλαδή την μέθοδο πληρωμής, το ποσό αλλά και την δέσμευση στην οποία αντιστοιχεί η πληρωμή.

+makeDebt(occupying)

Σε περίπτωση που ο *user* δεν δύναται να ολοκληρώσει την πληρωμή, δημιουργείται αντικείμενο *debt*, όπου αποθηκεύεται το κόστος (*total*) στην μεταβλητή *amount*. Από το στιγμιότυπο *occupying*, λαμβάνονται τα στοιχεία του *user* και του *parkingLot* στο οποίο οφείλει το ποσό.

Occupying

Οντότητα που διατηρεί τις πληροφορίες μιας δέσμευσης θέσης.

+startTimer()

Η μέθοδος αυτή ξεκινάει την μέτρηση του χρόνου για τον οποίο παραμένει δεσμευμένη η θέση, ώστε να είναι δυνατός ο υπολογισμός του κόστους, αν αυτό υπάρχει, της δέσμευσης θέσης.

+stopTimer()

Αντίστοιχα η μέθοδος αυτή σταματάει την μέτρηση του χρόνου δέσμευσης θέσης.

Βασικά Εργαλεία

Microsoft Word για την συγγραφή των τεχνικών κειμένων.

Diagrams.net για τη δημιουργία του Class Diagram.

Github ως αποθετήριο της ατομικής δουλειάς του κάθε μέλους της ομάδας, αλλά και ως εργαλείο οργάνωσης της συλλογικής δουλειάς.

Discord για τις συναντήσεις, την επικοινωνία, ανταλλαγή ιδεών και συζήτηση πάνω στον τρόπο εργασίας μας.

Java ως την γλώσσα προγραμματισμού για την ανάπτυξη της εφαρμογής

Android Studio ως Integrated Development Environment