Εικόνα που περιέχει γραμματοσειρά, κείμενο, γραφικά, στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

­­­

Domain Model v0.1

|  |  |  |
| --- | --- | --- |
| Ονοματεπώνυμο | E-Mail | AM |
| Μενέλαος Παναγιώτης Παπαστεργίου | up1072638@upnet.gr | 1072638 |
| Βαλεντίν Πασκάρι | up1075214@upnet.gr | 1075214 |
| Γιάννης Γεωργούλης | up1059440@upnet.gr | 1059440 |
| Ηλιάνα Πανταζή | up1072642@upnet.gr | 1072642 |

Ρόλοι:  
 Συντάκτης: Ηλιάνα Πανταζή

Βοηθός Συντάκτη: Μενέλαος Παναγιώτης Παπαστεργίου

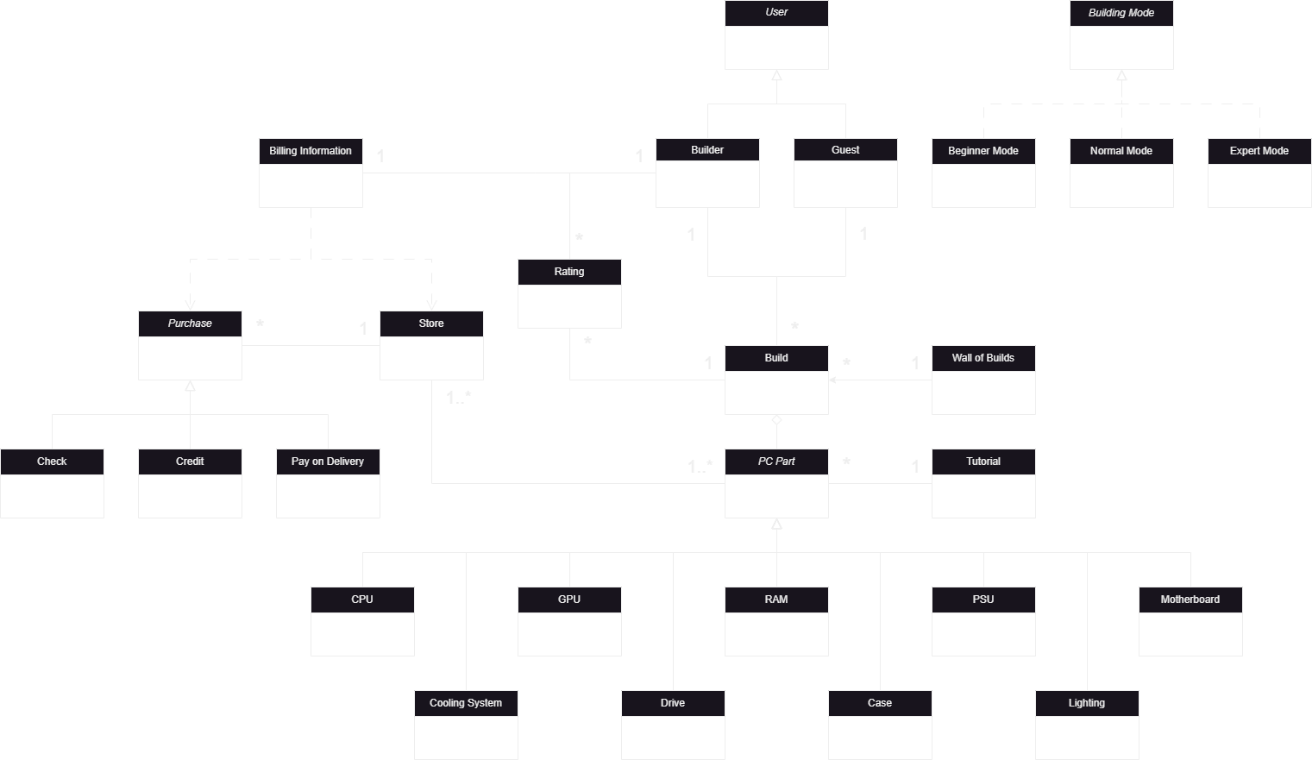
Παραγωγός Διαγραμμάτων : Μενέλαος Παναγιώτης Παπαστεργίου.

Εισαγωγή

Η εφαρμογή που θα υλοποιηθεί σχεδιάζεται ως μια web υπηρεσία, την οποία σκοπεύουμε να hostάρουμε σε cloud provider σε μελλοντικό χρόνο. Συνεπώς, σε αντίθεση με αμιγώς αντικειμενοστραφείς υλοποιήσεις, το domain πρέπει να υλοποιηθεί είτε με τις ίδιες κλάσεις είτε ελαφρώς παραποιημένο τόσο στο front end όσο και στο back end. Επίσης, κατά τη σχεδίαση του domain, σε τέτοιες περιπτώσεις, πρέπει να συνυπολογίζεται ότι μερικές από τις κλάσεις αυτές θα αποτελέσουν models για πίνακες βάσης δεδομένων. Στη συγκεκριμένη περίπτωση ειδικά, μιας και για την σχεδίαση της βάσης θα χρησιμοποιηθεί ORM (Object Relational Mapper), η σχεδίαση του domain ουσιαστικά θα καθορίσει σε μεγάλο βαθμό τη δομή της βάσης και κατά συνέπεια του μεγαλύτερου τμήματος της υλοποίησης.

Παρακάτω παρατίθεται διάγραμμα για το **domain model**. Στη συνέχεια ακολουθούν σύντομες περιγραφές των κλάσεων, συνοδευόμενες από επεξήγηση των συσχετίσεων που επιλέχθηκαν μεταξύ τους. Η περιγραφή αυτή δεν είναι δεσμευτική και ενδέχεται να προκύψουν αλλαγές, οι οποίες θα παρουσιαστούν σε μελλοντικές εκδόσεις.

Domain Model



Αναλυτική Περιγραφή Κλάσεων

* **User:** Αναφέρεται στους χρήστες που θα χρησιμοποιούν την υπηρεσία και χωρίζονται σε δύο κατηγορίες:
* **Guests:** Οι χρήστες αυτοί δεν θα έχουν λογαριασμό στην σελίδα.
* **Builders:** Οι χρήστες αυτοί έχουν δημιουργήσει λογαριασμό στην σελίδα και για αυτό, τους δίνεται η δυνατότητα να συμμετέχουν με επιπλέον υπηρεσίες εκτός της βασικής που αφορά την διαμόρφωση/παραγγελία του υπολογιστή που επιθυμεί.

(Αυτά τα δύο είδη χρηστών κληρονομούν χαρακτηριστικά από την υπερκλάση User)

* **Build:** Αναφέρεται ουσιαστικά, στο τελικό στάδιο διαμόρφωσης ενός υπολογιστή. Κάθε χρήστης builder μπορεί να αποθηκεύσει πολλά builds στον λογαριασμό του. Ωστόσο, στην περίπτωση του guest δεν υπάρχει δυνατότητα αποθήκευσης αυτών και για αποθήκευση απαιτείται δημιουργία χρήστη builder.
* **Wall of Builds:** Υπάρχει ένας μοναδικός “τοίχος” στον οποίο προβάλλονται πολλά builds τα οποία κατέχουν μεγάλη αξιολόγηση από τους χρήστες.
* **Component:** Ένα build αποτελείται από πολλαπλά κομμάτια (components) ενός υπολογιστή. Τα κομμάτια αυτά είναι:
* **CPU**
* **GPU**
* **RAM**
* **PSU**
* **Motherboard**
* **Cooling System**
* **Drive**
* **Case**
* **Lighting**
* **Tutorial:** Σε περίπτωση που ο χρήστης δεν έχει τις απαραίτητες γνώσεις για να προχωρήσει στο στάδιο διαμόρφωσης ενός υπολογιστή, του δίνεται δυνατότητα να παρακολουθήσει επεξηγηματικά tutorials (διαφόρων μορφών) όσον αφορά είτε ολόκληρα builds είτε και μεμονωμένα parts.
* **Rating:** Στους builders θα δίνεται η δυνατότητα να αξιολογούν builds άλλων χρηστών. Αυτό σημαίνει τόσο ότι ένας χρήστης μπορεί να έχει αξιολογήσει πολλά builds όσο και να έχει λάβει πολλές αξιολογήσεις για δικά του.
* **Store:** Αφορά τα καταστήματα τα οποία θα μας παρέχουν τα κομμάτια τα οποία είναι απαραίτητα για το στήσιμο ενός υπολογιστή. Κάθε κατάστημα μπορεί να προμηθεύει διάφορα components (αλλά τουλάχιστον ένα) και κάθε component πρέπει να πωλείται από τουλάχιστον ένα κατάστημα.
* **Billing Information:** Από την στιγμή που έχει δημιουργηθεί λογαριασμός χρήστη και ο χρήστης αυτός καταφτάσει στο στάδιο καταχώρησης μιας παραγγελίας είναι απαραίτητο να αποθηκευτούν στοιχεία πληρωμής και παράδοσης παραγγελίας.
* **Purchase:** Από την στιγμή που έχουν καταχωρηθεί τα στοιχεία πληρωμής και ο χρήστης προχώρησε στο στάδιο ολοκλήρωσης μιας παραγγελίας, η παραγγελία αυτή έχει αποθηκευτεί. Για μία αγορά είναι απαραίτητο να γνωρίζουμε τον τρόπο πληρωμής. ~~Οι τρόποι θα είναι ένας από τους τρεις παρακάτω:~~
* **~~Check (Επιταγή)~~**
* **Credit (Πίστωση)**
* **~~Pay on Delivery (Αντικαταβολή)~~**

Αποφασίσαμε τελικά, πως σαν τρόπο πληρωμής θα χρησιμοποιήσουμε μόνο πληρωμή με κάρτα καθώς αυτό μας φάνηκε ο αποδοτικότερος τρόπος για τις απαιτήσεις της εφαρμογής/επιχείρησής μας.

Υπάρχει dependency της κλάσης billing information προς την κλάση purchase, το οποίο σημαίνει ότι τα στοιχεία του billing information θα καθορίσουν το τελικό ποσό της παραγγελίας, όπως και άλλα χαρακτηριστικά.

Τέλος, καθώς μέλημά μας είναι η υπηρεσία μας να είναι προσιτή σε όσο πιο ευρύ κοινό είναι δυνατό, μέσω ενός interface **(Building Mode)** θα δημιουργηθούν τρία διαφορετικά είδη διαδικασίας διαμόρφωσης υπολογιστή. Κάθε κλάση θα περιλαμβάνει λεπτομέρειες υλοποίησης των διαδικασιών, ειδικές για κάθε επίπεδο χρήστη, ωστόσο όλες οι διαδικασίες θα πρέπει να υλοποιούν κάποια ορισμένη λειτουργικότητα που ορίζεται από το interface. Τα modes που θα υλοποιηθούν είναι:

* **Beginner Mode**
* **Normal Mode**
* **Expert Mode**