

CLICK N' BUILD

EMPOWERING YOUR VISION

Class Diagram v0.1

Class Diagram 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: Υπόλοιποι

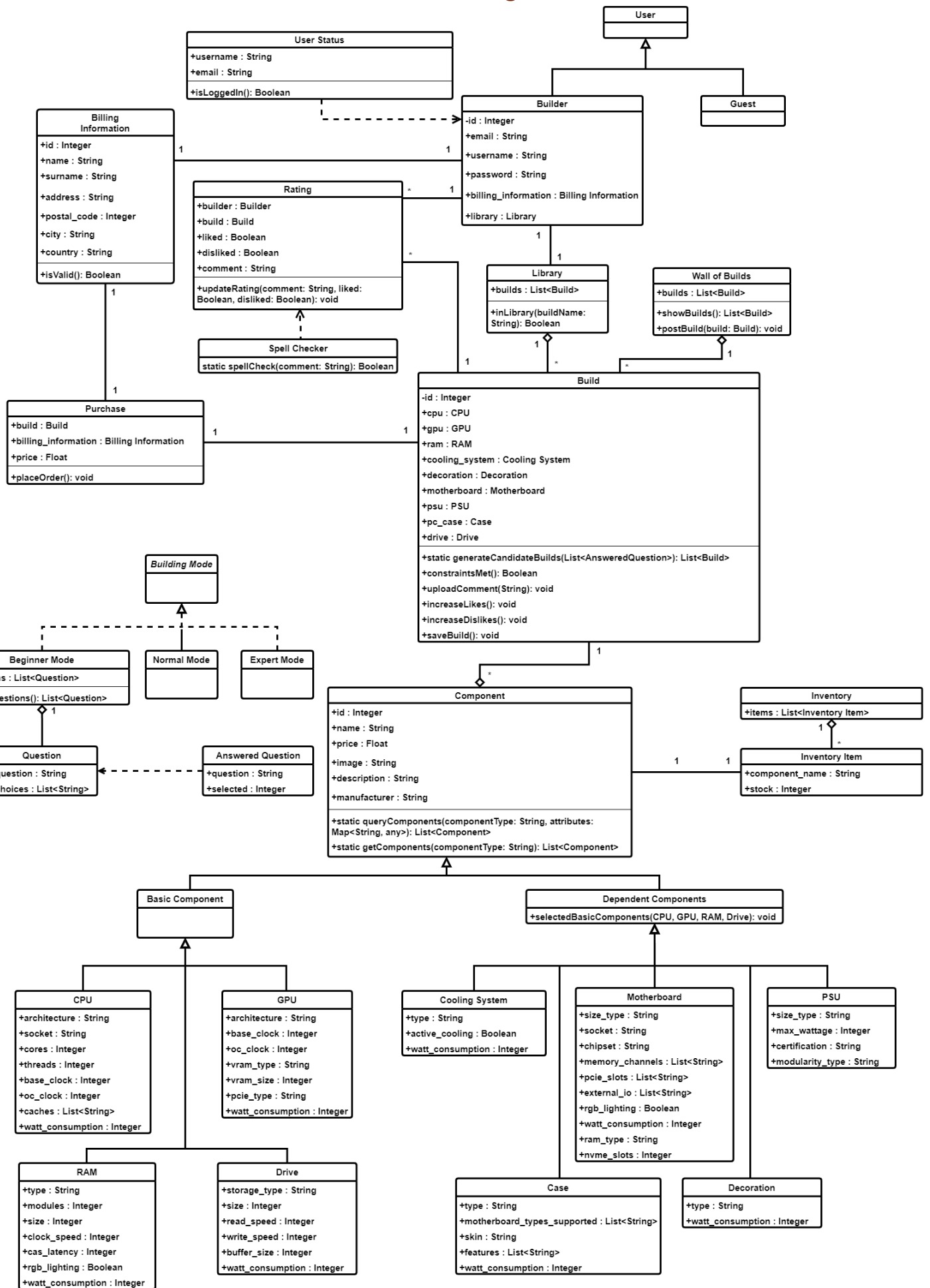
Παραγωγός Διαγραμμάτων : Μενέλαος Παναγιώτης Παπαστεργίου, Βαλεντίν Πασκάρι.

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.

Παρακάτω παρατίθεται διάγραμμα για το **class diagram**. Στη συνέχεια ακολουθούν σύντομες περιγραφές των κλάσεων και των μεθόδων τους, 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.

Class Diagram



Αναλυτική Περιγραφή Κλάσεων και των Μεθόδων τους

Κλάσεις:

1. **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 επιθυμούν.

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

2. **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.
3. **Wall of Builds:** Υπάρχει ένας μοναδικός “τοίχος” στον οποίο προβάλλονται πολλά builds τα οποία κατέχουν μεγάλη αξιολόγηση από τους χρήστες.
4. **Component:** Ένα build αποτελείται από πολλαπλά κομμάτια (components) ενός υπολογιστή. Τα κομμάτια αυτά είναι:
 - **CPU**
 - **GPU**
 - **RAM**
 - **PSU**
 - **Motherboard**
 - **Cooling System**
 - **Drive**
 - **Case**
 - **Lighting**
5. **Rating:** Στους builders θα δίνεται η δυνατότητα να αξιολογούν builds άλλων χρηστών μέσω του Wall of Builds. Αυτό σημαίνει τόσο ότι ένας χρήστης μπορεί να έχει αξιολογήσει πολλά builds όσο και να έχει λάβει πολλές αξιολογήσεις για δικά του.
6. **Billing Information:** Από την στιγμή που έχει δημιουργηθεί λογαριασμός χρήστη και ο χρήστης αυτός φτάσει 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: **Credit (Πίστωση)**

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**

7. **Inventory:** Αποτελεί την αποθήκη του “καταστήματος” μας. Περιλαμβάνει λίστα με όλα τα components τα οποία παρέχουμε.
8. **Inventory Item:** Αποτελεί ένα αντικείμενο αποθήκης με χαρακτηριστικά component και επιπλέον την ένδειξη για την ποσότητα σε απόθεμα.
9. **Library:** Αφορά τα builds που δημιούργησε κάθε χρήστης (δηλ. προσωπική βιβλιοθήκη builds).
10. **Questions:** Αποτελεί λίστα με ερωτήσεις που θα πρέπει να απαντηθούν από τους χρήστες που θα έχουν επιλέξει το Beginner Mode έτσι ώστε να παραχθεί το επιθυμητό τους build.
11. **Spell Checker:** Αφορά την διαχείριση των σχολίων που θα υποβάλλονται σε κάποιο build και είναι υπεύθυνη για την αποτροπή χρήσης ακατάλληλης φρασεολογίας κατά την αξιολόγηση από κάποιον χρήστη.
12. **Build Generator:** Είναι υπεύθυνη για την παραγωγή builds βασισμένη στα στοιχεία που δόθηκαν από εκάστοτε χρήστη.

Επιπλέον, η κλάση Components διασπάστηκε σε δύο υποκλάσεις με σκοπό να κατηγοριοποιηθούν. Η κατηγοριοποίηση αυτή αφορά τους χρήστες που έχουν μπει σε Normal Mode και τους δίνεται η δυνατότητα να επιλέξουν ότι επιθυμούν όσον αφορά τα βασικά components τα οποία δεν απαιτείται να έχουν κάποια συμβατότητα μεταξύ τους (CPU, GPU, RAM, Drive). Στην υποκλάση Dependent Components έχουν μπει όλα τα υπόλοιπα:

- **Basic Components**
- **Dependent Components**

Επιπλέον κλάσεις που προστέθηκαν κατά την διάρκεια της διαμόρφωσης του Class Diagram:

13. **User Status:** Κληρονομεί τα στοιχεία της (username, email) από την κλάση Builder. Χρησιμεύει για να αποθηκεύει προσωρινά τα στοιχεία των χρηστών που είναι logged in για τις περιπτώσεις που απαιτείται να ελεγχθεί η κατάσταση του χρήστη.
14. **Answered Questions:** Κληρονομεί στοιχεία από την κλάση Question και αποθηκεύει την απάντηση που επιλέχθηκε για μία συγκεκριμένη ερώτηση.

Περιγραφή χαρακτηριστικών κλάσεων:

- *Χαρακτηριστικά κλάσης **User**:*
 - **Guests:** Καθώς το είδος χρήστη αυτό δεν έχει κάνει εγγραφή στο σύστημά μας, διατηρούμε προσωρινά ένα id έως ότου ολοκληρώσει τις ενέργειές του με στόχο να μην χαθεί η πρόοδός του κατά την περιήγηση στην εφαρμογή.
 - **Builders:** Για τους εγγεγραμμένους χρήστες μας, πέραν των απαραίτητων στοιχείων για εγγραφή (id, email, password) διατηρούνται επίσης στοιχεία που αφορούν την καταχώρηση παραγγελίας (Billing Information) και μία λίστα με όλα τα builds που δημιούργησαν.

- *Χαρακτηριστικά κλάσης **Build**:*

Ένα build θα περιλαμβάνει:

- Μοναδικό id.
- Λίστα με όλα τα components που το σχηματίζουν.

Σημαντική αλλαγή στα χαρακτηριστικά της κλάσης Build: Διαπιστώθηκε πως αποθηκεύοντας μια απλή λίστα με όλα τα components από τα οποία αποτελείται ένα build δεν θα είναι εύκολη η υλοποίηση ελέγχου συμβατότητας των components μεταξύ τους. Συνεπώς, αποφασίστηκε πως κάθε component θα αποτελεί ξεχωριστό attribute της κλάσης Build.

- **Χαρακτηριστικά κλάσης *Component*:**
Στην κλάση *Component* δόθηκαν χαρακτηριστικά που ουσιαστικά το καθιστούν προϊόν (δηλ. όνομα, εικόνα, περιγραφή, κατασκευαστής, τιμή)
- **Χαρακτηριστικά κάθε *Component* ξεχωριστά:**
Στα *components* προστέθηκαν όλα τα χαρακτηριστικά που αφορούν επιδόσεις, εμφάνιση κ.α. και αυτά ουσιαστικά θα καθορίσουν την συμβατότητα μεταξύ των *component* για να καθοδηγήσουν τον χρήστη στο επιθυμητό του *build*.
- **Χαρακτηριστικά κλάσης *Rating*:**
Η κλάση αυτή θα περιλαμβάνει στοιχεία για το ποιος *builder* και για ποιο *build* έκανε την αξιολόγηση, επιπλέον εάν πατήθηκε *Like/Dislike* και τα σχόλια που καταχωρήθηκαν.
- **Χαρακτηριστικά κλάσης *Billing Information*:**
Από την στιγμή που καταχωρήθηκε μία παραγγελία από κάποιον *Builder* αποθηκεύονται τα στοιχεία διεύθυνσης του χρήστη αυτού (π.χ. όνομα, επώνυμο, διεύθυνση, Τ.Κ., χώρα, πόλη, κλπ.).
- **Χαρακτηριστικά κλάσης *Purchase*:**
Από την στιγμή που καταχωρήθηκε μία παραγγελία είτε από κάποιον *Guest* είτε *Builder* αποθηκεύονται στην κλάση αυτή στοιχεία για το ποιο *build* αγοράστηκε, τα στοιχεία διεύθυνσης του πελάτη και το τελικό ποσό της παραγγελίας.
- **Χαρακτηριστικά κλάσης *Question*:**
Θα περιλαμβάνει το ερώτημα, τις πιθανές απαντήσεις και το πόσες επιλέχθηκαν.

Μέθοδοι:

- **User Status:**
 - **isLoggedIn():** Ελέγχει εάν ο χρήστης έχει συνδεθεί στην σελίδα με τα *credentials* του.
- **Billing Information:**
 - **isValid():** Ελέγχει εάν τα στοιχεία που έχουν συμπληρωθεί από τον χρήστη είναι ορθά και έχουν την κατάλληλη μορφή.
- **Purchase:**
 - **placeOrder():** Μόλις έχουν συμπληρωθεί τα απαραίτητα στοιχεία και ο χρήστης είναι έτοιμος να ολοκληρώσει την παραγγελία του καλείται η μέθοδος αυτή.
- **Library:**
 - **inLibrary():** Για *build* το οποίο επέλεξε ο χρήστης ελέγχει εάν έχει αποθηκευτεί στην βιβλιοθήκη του έτσι ώστε να δοθεί η δυνατότητα στον

χρήστη να προχωρήσει στην διαδικασία δημοσίευσης στον τοίχο της σελίδας (Wall of Builds).

- **Rating:**

- **updateRating():** Μόλις κάποιος χρήστης έχει προσθέσει σε κάποιο build κάποιου είδους αξιολόγησης, καλείται η μέθοδος αυτή για να ενημερώσει την κατάσταση του build ως προς τις αξιολογήσεις.

- **Spell Checker:**

- **spellCheck():** Ελέγχει την ορθότητα του προς υποβολή σχόλιο.

- **Wall of Builds:**

- **showBuilds():** Όταν κάποιος χρήστης επιθυμήσει να οδηγηθεί στον τοίχο με τα δημοσιευμένα builds καλείται η μέθοδος αυτή για να τα εμφανίσει.
- **postBuild():** Όταν κάποιος χρήστης επιθυμήσει να δημοσιεύσει το build που δημιουργείται καλείται η μέθοδος αυτή.

- **Beginner Mode:**

- **showQuestions():** Μόλις κάποιος χρήστης επιλέξει πως θέλει να δημιουργήσει build για αρχάριους καλείται η μέθοδος αυτή για να εμφανίσει στον χρήστη το ερωτηματολόγιο που πρέπει να συμπληρώσει.

- **Build:**

- **generateCandidate():** Η μέθοδος αυτή αφορά το Beginner Mode και καλείται μόλις ο χρήστης έχει απαντήσει το ερωτηματολόγιο και βάσει αυτού δημιουργούνται υποψήφια builds από τα οποία μπορεί να επιλέξει.
- **constraintsMet():** Η μέθοδος αυτή αφορά το Normal και το Expert Mode όπου ο χρήστης έχει μεγαλύτερη ελευθερία κατά την διαμόρφωση του build ωστόσο, πρέπει να τηρούνται κάποιοι περιορισμοί, οι οποίοι ελέγχονται μέσω της μεθόδου αυτής και είτε θα επιτρέψει είτε θα απαγορεύει τον χρήστη από το να προχωρήσει στο επόμενο βήμα.
- **uploadComment():** Μόλις κάποιος χρήστης γράψει κάποιο σχόλιο για ένα συγκεκριμένο build, καλείται η μέθοδος αυτή για να ενημερώσει την λίστα των σχολίων.
- **increaseLikes():** Μόλις κάποιος χρήστης πατήσει like σε ένα συγκεκριμένο build, καλείται η μέθοδος αυτή για να ενημερώσει την ποσότητα των likes στο build αυτό.
- **increaseDislikes():** Όμοια με increaseLikes() αλλά για τα dislikes.
- **saveBuild():** Μόλις κάποιος χρήστης είναι ευχαριστημένος με το build που διαμόρφωσε και επιθυμεί να το αποθηκεύσει στην βιβλιοθήκη (library) του, καλείται η μέθοδος αυτή.

- **Component:**

- **queryComponents():** Αφορά το Beginner και το Normal Mode. Πραγματοποιεί αναζήτηση components με βάση τον τύπο και τα χαρακτηριστικά τους.
- **getComponents():** Αφορά και τα τρία Modes και επιστρέφει όλα τα components ενός τύπου.

- **Dependent Components:**

- **selectBasicComponents():** Αφορά το Normal Mode. Προσδιορίζει ποια basic components έχουν επιλεγθεί στην προηγούμενη φάση του mode αυτού.