

## **Domain-model-v0.3**

Domain-model-v0.3

## **HELLO BABY**



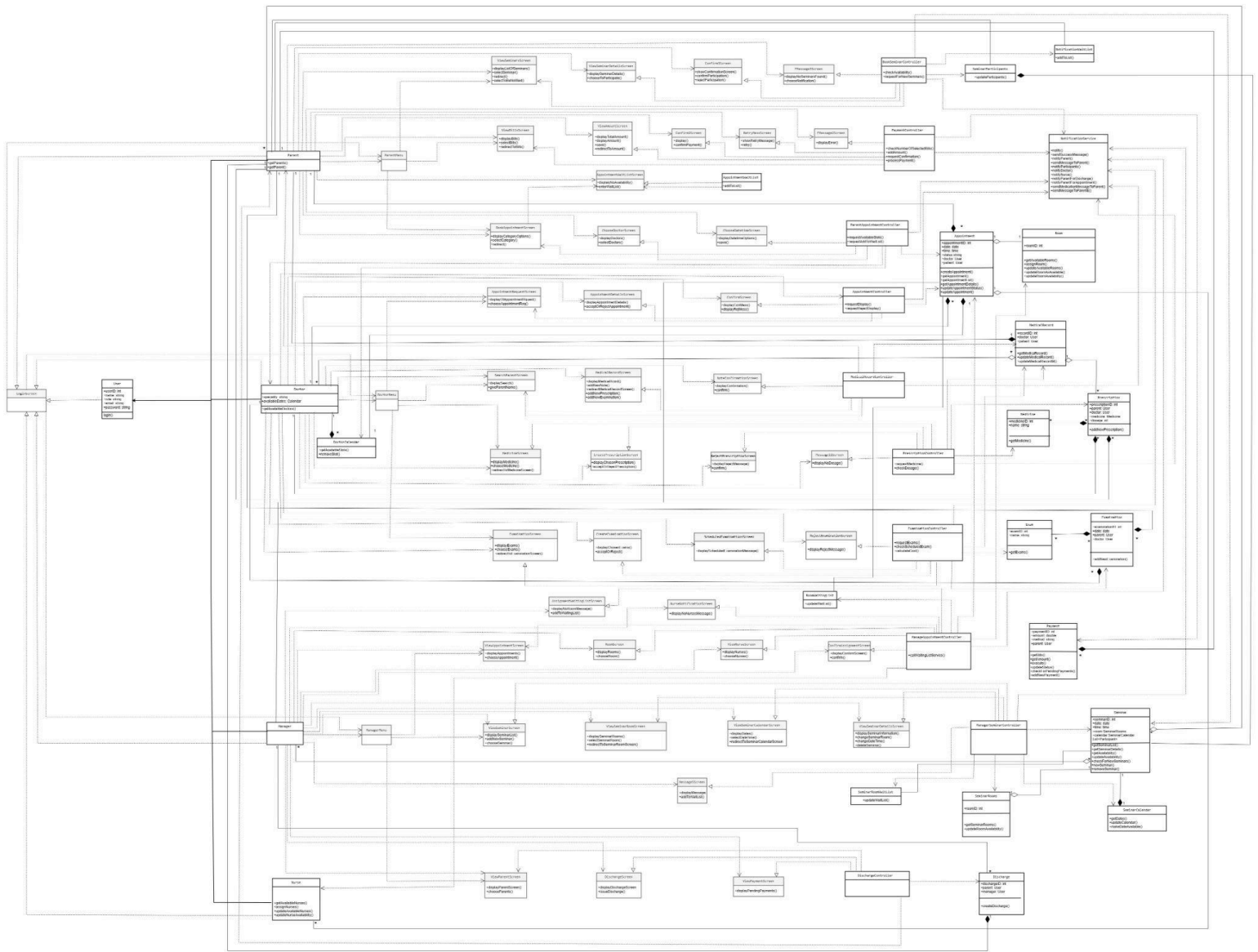
## Ομάδα

Όνομα	Επώνυμο	ΑΜ	email
Κρίστελ	Ντουλάϊ	1064498	up1064498@ac.upatras.gr
Ανδρέας	Γιαννικόπουλος	1067393	up1067393@ac.upatras.gr
Κλεοπάτρα	Πελέκη	1072557	up1072557@ac.upatras.gr
Ιωάννα	Νικολακοπούλου	1072609	up1072609@ac.upatras.gr
Βασιλική	Παγώνη	1072497	up1072497@ac.upatras.gr

### **ΠΑΡΑΤΗΡΗΣΕΙΣ:**

- Όπως αναφέρθηκε και στην προηγούμενη έκδοση του αρχείου τα ζητούμενα use cases πρέπει να είναι 10 για αυτό τον λόγο αποφασίσαμε να μην ασχοληθούμε με τα use case του νοσηλευτικού προσωπικού (nurse) δηλαδή τα use case A και B(με την νέα αρίθμηση). Για αυτό τον λόγο δεν έχουν προστεθεί ούτε στο v0.3 domain διάγραμμα οι κλάσεις του νοσηλευτικού προσωπικού (nurse).
- Για καλύτερη ανάγνωση του διαγράμματος παραθέτουμε και το link για το github όπου είναι αναρτημένη μαζί με το αρχείο domain\_model\_v0.3, με τίτλο Domain\_Model.jpg.

ΔΙΑΓΡΑΜΜΑ:



## Domain Model Description

Η κλάση User συνδέεται με κληρονομικότητα(inheritance) με τις κλάσεις Parent, Manager, Nurse, Doctor.

Το κάθε use case έχει από ένα controller όπου εκπληρώνει τις λογικές λειτουργίες του συστήματος. Από εκεί καλούνται τα interfaces όπου εμφανίζεται η πληροφορία. Για αυτό οι controllers κάθε use case συνδέονται με τις κλάσεις του interface με τη σχέση υλοποίησης (implementation).

Σε αυτό το παραδοτέο η κλάση MedicalRecordController χωρίστηκε σε τρεις διαφορετικές. Η MedicalRecordController αφορά την ενημέρωση ιατρικού φακέλου από τον γιατρό, η κλάση PrescriptionController αφορά την συνταγογράφηση, και η κλάση ExaminationController αφορά τις εξετάσεις.

Ο κάθε χρήστης συνδέεται με τις κλάσεις interface με απλή σύνδεση.

Όλες οι συνδέσεις οι οποίες ήταν με βέλος συσχέτισης(association) κατευθυνόμενο στο διάγραμμα από την έκδοση v0.2 σε αυτή την έκδοση έγιναν μη κατευθυνόμενες συσχετίσεις.

## Η κλάση *Parent*:

Η κλάση *Parent* συνδέεται με σύνθεση (composition) με το *Appointment*. Τα ραντεβού δεν υπάρχουν χωρίς τον γονέα. Ένας γονέας μπορεί να έχουν πολλά ραντεβού άρα η συσχέτιση (association) είναι 1 - \* (ένα με πολλά).

Ο *Parent* μπορεί να έχει πολλαπλές *Payment* μπορεί και καθόλου άρα συνδέεται με συσχέτιση (dependency) 1 - \* (ένα με κανένα ή πολλά). Συνδέεται επίσης και με σύνθεση (composition) γιατί για να υπάρχουν πληρωμές πρέπει να υπάρχει ο γονέας.

Ο *Parent* και το *Seminar* συνδέονται με συνάθροιση (aggregation) και έχουν συσχέτιση \* - \* (πολλά προς πολλά).

Ο *Parent* συνδέεται με το *Discharge* με σύνθεση (composition) και συσχέτιση 1-1 (ένα προς ένα), αφού ένα *Discharge* αντιστοιχεί σε ένα *Parent*.

Ο *Parent* με το *Prescription* και το *Examination* συνδέεται με σύνθεση (composition) γιατί δεν μπορεί να γίνει συνταγογράφηση ή εξέταση χωρίς γονέα αλλά ένας *parent* μπορεί να έχει πολλά άρα 1 - \* (ένας σε πολλά).

Ο *Parent* με την κλάση *SeminarParticipants* συνδέονται με απλή συσχέτιση (association).

Ο *Parent* με την κλάση *NotificationWaitList* (σε αυτό το παραδοτέο v0.2 το *NotificationWaitList* είχε το όνομα *NotifyList* το οποίο άλλαξε) συνδέονται με απλή συσχέτιση (association).

Η κλάση *Parent* με την κλάση *AppointmentWaitList* συνδέονται με συσχέτιση (association).

Το *MedicalRecord* ανήκει αποκλειστικά σε έναν *Parent*, συνδέονται με σύνθεση (composition) και είναι 1 - 1 αφού ένας *parent* έχει ένα ιατρικό φάκελο.

## Η κλάση *Doctor*:

Η κλάση *Doctor* συνδέεται με σύνθεση (composition) με το *Appointment*. Τα ραντεβού δεν υπάρχουν χωρίς τον γιατρό. Ένας γιατρός μπορεί να έχουν πολλά ραντεβού άρα η συσχέτιση (association) είναι 1 - \* (ένα με πολλά).

Ο *Doctor* συνδέεται με το *Prescription* και το *Examination* με σύνθεση (composition) γιατί δεν γίνεται να υπάρχει συνταγογράφηση και εξέταση χωρίς τον γιατρό και αφού ένας γιατρός μπορεί να έχει πολλές συνταγογραφήσεις και εξετάσεις είναι 1 - \* (ένα προς πολλά).

Ο *Doctor* με την κλάση *DoctorCalendar* συνδέονται με σύνθεση (composition) και με συσχέτιση 1 - \* (ένα με πολλά), αφού δεν γίνεται να υπάρχει ημερολόγιο γιατρού χωρίς τον γιατρό.

Ο *Doctor* δεν συνδέεται με τον *MedicalRecord* με σύνθεση (composition) γιατί δεν ανήκει αποκλειστικά σε ένα γιατρό. Αν ο γιατρός διαγραφεί ο φάκελος μένει. Δεν συνδέεται επίσης με συνάθροιση (aggregation) γιατί ο γιατρός ενημερώνει τον ιατρικό φάκελο αλλά ο φάκελος δεν εξαρτάται από την ύπαρξη του γιατρού, μπορεί να ενημερωθεί από πολλούς γιατρούς. Έχει απλή συσχέτιση (association) με πολλαπλότητα \*-\*(πολλά με πολλά). Ομοίως και για την σχέση *Nurse* και *MedicalRecord*, αλλά συνδέεται με απλή συσχέτιση (association).

## Η κλάση *Manager*:

Ο *Manager* μπορεί να διαχειρίζεται τα *Appointment* αλλά δεν είναι μέρος αυτών, άρα συνδέονται με συσχέτιση (association).

Ο *Manager* συνδέεται με το *Seminar* με συνάθροιση (aggregation) αφού ο *manager* φτιάχνει τα *seminars* αλλά και να διαγραφεί αυτός αυτά παραμένουν συσχέτιση \* - \* (πολλά προς πολλά)

Ο *Manager* συνδέεται με το *Discharge* με απλή συσχέτιση (association) (όχι σύνθεση (composition) όπως είχαμε βάλει στο v0.2) το *discharge* δεν μπορεί να υπάρχει χωρίς τον *manager* και η συσχέτιση είναι 1 - \* (ένα προς πολλά)

Η κλάση *Prescription* συνδέεται με την κλάση *Medicine* με σύνθεση(composition) και είναι \* - \* (πολλά με πολλά), αφού πολλές prescriptions μπορούν να έχουν πολλά φάρμακα και ένα φάρμακο μπορεί να υπάρχει σε πολλές συνταγές.

Η κλάση *Examination* συνδέεται με την κλάση *Exam* με σύνθεση(composition) και είναι \* - \* (πολλά με πολλά).

Η κλάση *Seminar* με την κλάση *SeminarRoomWaitList* συνδέονται με συσχέτιση(association).

Η κλάση *Appointment* με την κλάση *Room* συνδέονται με συνάθροιση(aggregation) και η συσχέτιση 1 – 1(ένα προς ένα).

Η κλάση *Appointment* με την κλάση *RoomWaitList* συνδέονται με συσχέτιση(association) \* - 1 (πολλά προς ένα).

Το *Seminar* συνδέεται με το *SeminarRoom* με συνάθροιση(aggregation) και η συσχέτιση είναι 1 – 1(ένα προς ένα), ομοίως και για το *Appointment* με το *Room*.

Το *SeminarCalendar* συνδέεται με το *Seminar* με σύνθεση(composition) και συσχέτιση 1 – 1 (ένα προς ένα).

Η κλάση *Appointment* συνδέεται με το *DoctorCalendar* με σύνθεση(composition) αφού χωρίς ημερομηνίες δεν μπορεί να κλειστεί ραντεβού και με πολλαπλότητα \* - \* (πολλά με πολλά).

Η κλάση *Seminar* συνδέεται με την κλάση *SeminarParticipants* συνδέονται με σύνθεση(composition) γιατί χωρίς το seminar δεν μπορεί να υπάρχει λίστα συμμετεχόντων.

#### **GitHub Link για τη φωτογραφία:**

<https://github.com/eimai-o-andreas/Software-Engineering-Project-CEID>

#### **Εργαλεία**

- Το διάγραμμα παράχθηκε με την βοήθεια του draw.io.