

Domain-model-v0.2

Domain-model-v0.2

HELLO BABY



Ομάδα

Όνομα	Επώνυμο	ΑΜ	email
Κρίστελ	Ντουλαΐ	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

Αλλαγές έκδοσης v0.2 σε σχέση με την v0.1:

- Οι αλλαγές σημειώνονται παρακάτω με μπλε χρώμα στο κείμενο και οφείλονται κυρίως στη δημιουργία των robustness διαγραμμάτων με τα οποία έγινε πιο ξεκάθαρη η ροή του κώδικα και κυρίως το GUI μέρος του οι οθόνες του οποίου φαίνονται στην εικόνα με ανοιχτό γκρι.
- Όλες οι κλάσεις που είναι controllers είναι service classes και περιέχουν μεθόδους που χρειάζεται στο σύστημα για την επικοινωνία με τις άλλες κλάσεις. Στο διάγραμμα μας υπάρχουν κάποιες μέθοδοι ενδεικτικά που έχουν εντοπιστεί, όμως χωρίς τα sequence διαγράμματα δεν μπορούμε να αναφέρουμε σίγουρα ότι αυτή η μορφή είναι η τελική.

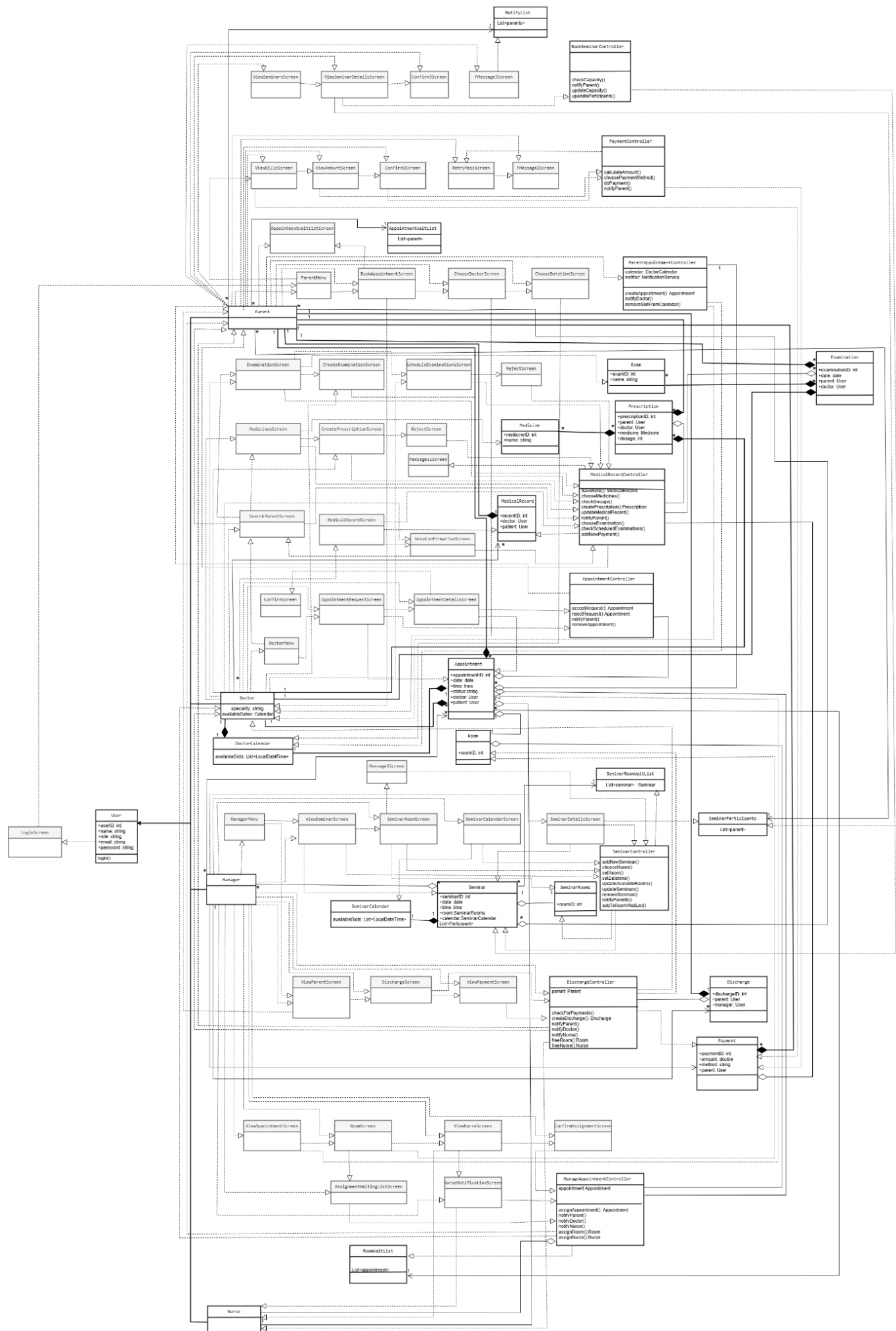
ΠΑΡΑΤΗΡΗΣΕΙΣ:

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 7 και 8. Για αυτό τον λόγο δεν έχουν προστεθεί στο domain διάγραμμα οι κλάσεις του νοσηλευτικού προσωπικού (nurse).

Εργασία

- Το σχήμα 1 παράχθηκε με την βοήθεια του draw.io.

Σχήμα 1 :



Domain Model Description

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

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) 0 - *(ένα η κανένα με 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 συσχέτιση(association) * - 1 (πολλοί με ένα).

Ο *Parent* με την κλάση *NotifyList* συνδέονται με απλή συσχέτιση(association) και ισχύει * - 1(πολλά με ένα), αφού πολλοί γονείς μπορούν να ανήκουν στην ίδια λίστα.

Η κλάση *Parent* με την κλάση *AppointmentWaitList* συνδέονται με συσχέτιση(association) με * - 1, αφού πολλοί γονείς μπορούν να βρίσκονται στην λίστα αναμονής.

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 – 1(ένα προς ένα), 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, 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) * - *(πολλά με πολλά).

Ο *Manager* συνδέεται με το *Payment* με συσχέτιση(association) γιατί μπορεί να βλέπει τις πληρωμές αλλά όχι να τις επεξεργάζεται.

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 σύνθεση(composition) 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) * - 1 (πολλοί με ένα).

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).