



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΣΤΕΡΕΑΣ ΕΛΛΑΔΑΣ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΙΑΤΡΙΚΗ**

**Πληροφοριακά Συστήματα Αποδοτικής Διαχείρισης  
Ασθενών και Πόρων σε Μονάδα Φροντίδας Υγείας**

**Σαρδέλλας Αντώνιος**

**A.M. 6**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ  
Υπεύθυνος  
Dr. Φουντούκης Σπυρίδων  
ΠΔ407/80**

**Λαμία, 2013**

**«Πληροφοριακά Συστήματα Αποδοτικής Διαχείρισης  
Ασθενών και Πόρων σε Μονάδα Φροντίδας Υγείας»**

## Περίληψη

Η πτυχιακή εργασία με τίτλο «Πληροφοριακά Συστήματα Αποδοτικής Διαχείρισης Ασθενών και Πόρων σε Μονάδα Φροντίδας Υγείας» προσπαθεί να μελετήσει τα Πληροφοριακά συστήματα και ειδικότερα αυτά της διαχείρισης πόρων (ERP), στον ιδιαίτερα σημαντικό και ευαίσθητο τομέα της Υγείας. Η επιτυχής τους εισαγωγή στον επιχειρηματικό κόσμο, μας επιδεικνύει τα οφέλη που μπορούν να αποκομισθούν από την εφαρμογή τους στο τομέα της Υγείας.

Αρχικά, στην παρούσα εργασία, περιγράφονται τα τμήματα ενός τέτοιου συστήματος, τα χαρακτηριστικά του και οι λειτουργίες του. Αναλύονται τα πλεονεκτήματα που προσφέρει, αλλά και οι δυσκολίες που συναντούνται στην εφαρμογή τους. Ακόμα, αναφέρονται υπάρχοντα πληροφοριακά συστήματα που εφαρμόζονται ήδη σε μονάδες φροντίδας Υγείας.

Στη συνέχεια εστιάζουμε σε ένα συγκεκριμένο υποπρόβλημα που αφορά την κινητικότητα των ασθενών στον ίδιο νοσοκομειακό οργανισμό. Με τη χρήση τεχνολογιών αιχμής, παρουσιάζετε ένα υποσύστημα αυτόματης διεκπεραίωσης ραντεβού, που χρησιμοποιεί κατάλληλα του πόρους των εργαστηριακών μονάδων ενός οργανισμού, για την ταχύτερη εξυπηρέτηση των ασθενών.

Τέλος, παρουσιάζετε αναλυτικά η κατασκευή της εφαρμογής, και αναφέρονται οι τεχνολογίες που χρησιμοποιήθηκαν και δίνεται έμφαση στην εύκολη επέκταση και ενσωμάτωση του υποσυστήματος σε ένα ολοκληρωμένο Πληροφοριακό Σύστημα Διαχείρισης Ασθενών και Πόρων.

## **Abstract**

The present work titled "Information Systems of Efficient resource and patient management in a Health Unit" is trying to study the Information systems, notably the Enterprise Resource Systems (ERP), in the sensitive and important Health domain. Their successful implementation in the business world indicates the benefits that can be drawn by introducing them in Health.

At first, in this essay, we describe the sections of said system, their characteristics and their functions. We analyze its benefits, but also the difficulties in implementing them. We state existing Systems that are used in Health Units.

Then, we focus in a particular problem that regards to the mobility of patients in one Health organization. Using cutting-edge technologies, we present a subsystem that handles the automated dating process, which uses the laboratory resources appropriately to speed up the patient service.

Finally, we present thoroughly, the making of the application, we analyze the technologies used, and we focus on the ease of integrating the subsystem in an already operational Information system.

## Πίνακας περιεχομένων

Πίνακας εικόνων .....	7
1 Εισαγωγή .....	8
2 Πληροφοριακά συστήματα .....	9
2.1 Ορισμοί .....	9
2.2 Τα τμήματα ενός πληροφοριακού συστήματος.....	10
3 Ολοκληρωμένα συστήματα διαχείρισης πόρων .....	11
3.1 Ιστορία των "ERP" .....	12
3.2 Χαρακτηριστικά των ERP- Λειτουργικές μονάδες.....	14
3.3 Εγκατάσταση και λειτουργία ERP .....	16
3.4 Πλεονεκτήματα.....	17
3.5 Μειονεκτήματα .....	19
4 Πληροφοριακά συστήματα διαχείρισης πόρων υγείας .....	20
4.1 Σκοπός .....	20
4.1.1 Διοικητική Μέριμνα .....	21
4.1.2 Κλινική Μέριμνα .....	26
4.2 Υπάρχοντα πληροφοριακά συστήματα υγείας .....	28
4.2.1 Πληροφοριακά συστήματα βασισμένα στο μοντέλο του ERP 30	
4.3 Συμπεράσματα .....	34
5 Test case .....	36
5.1 Εισαγωγή .....	36
5.2 Περιγραφή .....	36

5.3	Σχεδιασμός του συστήματος.....	37
6	Υλοποίηση του συστήματος.....	43
6.1	Λογισμικό και τεχνολογίες που έχουν υιοθετηθεί .....	43
6.1.1	Μεθοδολογία ανάπτυξης Μοντέλου - Όψης - Ελεγκτή 44	
6.1.1.1	Σχέσεις μεταξύ των συνιστωσών.....	46
6.1.1.2	Πλεονεκτήματα από τη χρήση του MVC .....	49
6.1.1.3	Μειονεκτήματα του MVC.....	50
6.1.1.4	Αρχιτεκτονική MVC – Πλατφόρμες MVC.....	50
6.1.2	MySQL.....	53
6.1.3	Γλώσσα προγραμματισμού Java .....	55
6.1.4	Appfuse .....	56
6.1.5	Spring MVC .....	58
6.2	Πλεονεκτήματα από την υιοθέτηση τεχνολογιών αιχμής	59
6.3	Παρουσίαση της εφαρμογής .....	61
7	Επίλογος.....	68
8	Παράρτημα- Source Code .....	72
8.1	Controllers .....	74
8.2	Models.....	97
8.3	Manager Classes.....	106
8.4	Views .....	108

## Πίνακας εικόνων

Εικόνα 1: User-Action Diagram .....	39
Εικόνα 2: Διάγραμμα ροής για την ενέργεια κλεισίματος ραντεβού .....	41
Εικόνα 3: Το σχήμα της βάσης που προκύπτει .....	42
Εικόνα 4: Γενική αναπαράσταση του μοντέλου MVC .....	45
Εικόνα 5 MVC δομή .....	48
Εικόνα 6: Γραφική Διεπαφή – Μονάδα πιστοποίησης χρηστών ..	61
Εικόνα 7: Γραφική Διεπαφή – Κεντρικό μενού .....	62
Εικόνα 8: Επεξεργασία προφιλ χρήστη .....	63
Εικόνα 9: Λίστα ασθενών .....	64
Εικόνα 10: Προσθήκη / Επεξεργασία ασθενή .....	64
Εικόνα 11: Κατάλογος ιατρείων/εργαστηρίων .....	65
Εικόνα 12: Προσθήκη/επεξεργασία ιατρείου/εργαστηρίου .....	66
Εικόνα 13: Εισαγωγή/Επεξεργασία Εξέτασης .....	66
Εικόνα 14: Κατάλογος διαθέσιμων εξετάσεων ανά εργαστήριο/ιατρείο .....	67

# 1 Εισαγωγή

Τα πληροφοριακά συστήματα αποδοτικής διαχείρισης ασθενών και πόρων υγείας αποτελούν μια από τις σημαντικότερες απαιτήσεις στα συστήματα υγείας που σταδιακά μεταμορφώνονται όλα τα σύγχρονα δημόσια και ιδιωτικά συστήματα υγείας. Η εισαγωγή των ηλεκτρονικών υπολογιστών σε όλα τα στάδια της λειτουργίας τους κρίνεται απαραίτητη για την βελτίωση των παρεχόμενων υπηρεσιών, την καλύτερη εμπειρία των ασθενών και την αποδοτικότερη λειτουργία.

Τα τελευταία χρόνια γίνεται μια ευρεία προσπάθεια στην Ευρώπη, για την διείσδυση των τεχνολογιών πληροφορικής και επικοινωνίας σε κάθε μορφή και έκφανση της καθημερινότητας, με ιδιαίτερη προσοχή στον τομέα της υγείας. Αρκετές ελληνικές και διεθνείς προσπάθειες έχουν καταγραφεί, προς την παροχή ολοκληρωμένων συστημάτων υποστήριξης οργανισμών υγείας [1], ενώ ο διεθνής οργανισμός υγείας έχει προχωρήσει στην συστηματική καταγραφή της διείσδυσης των Τεχνολογιών Πληροφοριών και Επικοινωνιών (ΤΠΕ) στον τομέα της υγείας στις χώρες μέλη [2].



## **2 Πληροφοριακά συστήματα**

Τα πληροφοριακά συστήματα έχουν μπει στη ζωή του σύγχρονου ανθρώπου και αποτελούν αναπόσπαστο τμήμα της καθημερινότητάς του, ακόμη κι όταν αυτό δεν είναι τόσο εμφανές. Πριν προχωρήσουμε όμως, στην έννοια και την λειτουργία των πληροφοριακών συστημάτων, ας ξεκινήσουμε με τους ορισμούς που θα χρειαστούμε στην συνέχεια.

### **2.1 Ορισμοί**

Ένα πληροφοριακό σύστημα συγκροτείται γύρω από δύο βασικούς πυλώνες: τα δεδομένα και την πληροφορία.

Δεδομένα είναι κάθε στοιχείο, οποιασδήποτε μορφής (κείμενο, αριθμοί κλπ) τα οποία παράγονται στους υπολογιστές, ή συλλέγονται αυτοματοποιημένα από εξωτερικές πηγές. Τα δεδομένα είναι τα πρωτογενή στοιχεία που χρησιμοποιούνται μέσα σε ένα πληροφοριακό σύστημα.

Η πληροφορία εξάγεται μετά από κατάλληλη επεξεργασία των δεδομένων και έχει μορφή η οποία είναι αντιληπτή και χρήσιμη στον τελικό χρήστη.

Το Πληροφοριακό Σύστημα αποτελείται από οντότητες και έχει ως αποκλειστικούς σκοπούς τη συλλογή, την αποθήκευση, την ανάλυση και επεξεργασία των δεδομένων και την παρουσίαση της παραγόμενης πληροφορίας στον τελικό χρήστη.

Οι οντότητες του πληροφοριακού συστήματος αποτελούνται από οντότητες εισόδου, επεξεργασίας, αποθήκευσης και παρουσίασης.

## **2.2 Τα τμήματα ενός πληροφοριακού συστήματος**

Από τον ορισμό του πληροφοριακού συστήματος μπορούμε εύκολα να διακρίνουμε και τα τμήματά του:

- **Τμήμα εισαγωγής δεδομένων:** Το τμήμα αυτό αποτελείται από πολλαπλά συστήματα τα οποία τροφοδοτούν το πληροφοριακό σύστημα με τα απαραίτητα δεδομένα. Τα συστήματα αυτά έχουν μεγάλο εύρος και μπορούν να ποικίλλουν από κοινές συσκευές εισόδου, όπως το πληκτρολόγιο, έως και πολυσύνθετα συστήματα λήψης δεδομένων από διάφορες αισθητήρες (όπως θερμοκρασίας, οπτικούς, ηχητικούς και υπέρυθρους)
- **Τμήμα αποθήκευσης δεδομένων:** Το τμήμα αποθήκευσης των δεδομένων στις περισσότερες περιπτώσεις πληροφοριακών συστημάτων ανατίθεται σε ένα σύστημα διαχείρισης βάσεων δεδομένων, το οποίο διευκολύνει την πρόσβαση τους, με πληθώρα ερωτημάτων.
- **Τμήμα ανάλυσης/επεξεργασίας:** Το τμήμα ανάλυσης δεδομένων αποτελεί το κύριο τμήμα λογισμικού του πληροφοριακού συστήματος
- **Τμήμα παρουσίασης πληροφορίας**

### **3 Ολοκληρωμένα συστήματα διαχείρισης πόρων**

Ένα ολοκληρωμένο σύστημα διαχείρισης πόρων (Enterprise Resource Planning - ERP) είναι ένα τύπος υπερ-συστήματος ο οποίος αξιοποιείται από τις επιχειρήσεις και αποτελεί μια ολοκληρωμένη λύση λογισμικού κατάλληλη για κάθε είδους επιχείρηση.

Ένα ERP αποτελείται από διακεκριμένες ενότητες λογισμικού κάθε μια από τις οποίες υποστηρίζει μια ξεχωριστή εσωτερική επιχειρηματική διαδικασία της εταιρείας, ενώ ταυτόχρονα μοιράζονται την κοινή βάση δεδομένων του ERP. Έτσι, είναι σε θέση δίνει σε πραγματικό χρόνο για μια εταιρεία μια ολοκληρωμένη εποπτεία των δραστηριοτήτων της, όπως για παράδειγμα την παραγωγή, την επεξεργασία παραγγελιών και τη διαχείριση των αποθεμάτων.

Ένα από τα σχεδιαστικά πλεονεκτήματα των συστημάτων ERP αποτελεί το γεγονός ότι διευκολύνει τη ροή πληροφοριών μεταξύ όλων των επιχειρηματικών λειτουργιών στο εσωτερικό μιας

επιχείρησης, και διαχειρίζεται τις συνδέσεις με εξωτερικούς φορείς [3].

Οι σύγχρονες επιχειρήσεις (μεγάλου μεγέθους κυρίως) θεωρούν τα συστήματα ERP ζωτικά εργαλεία για την οργάνωσή τους, καθώς ενσωματώνουν ποικίλα οργανωτικά συστήματα και επιτρέπει την απρόσκοπτη και άμεση ροή των δεδομένων μεταξύ των επιμέρους τμημάτων της επιχείρησης [4].

### **3.1 Ιστορία των "ERP"**

Τα ERP συστήματα γνώρισαν ταχεία ανάπτυξη στη δεκαετία του 1990, με αφορμές κυρίως το πρόβλημα του έτους 2000 και την εισαγωγή του ευρώ στην Ευρωπαϊκή, και κατ' επέκταση στην παγκόσμια αγορά, τα οποία κατέστησαν ορισμένα παλαιά συστήματα μη λειτουργικά. Πολλές εταιρείες άδραξαν την ευκαιρία που τους δόθηκε και επένδυσαν κατά την αντικατάσταση του μη λειτουργικού τους λογισμικού με ένα σύστημα ERP. [5]

Τα συστήματα ERP αρχικά στόχευαν στην αυτοματοποίηση των εσωτερικών λειτουργιών των επιχειρήσεων που δεν επηρεάζουν άμεσα τους πελάτες και το ευρύ κοινό (Back-Office διαδικασίες). Σταδιακά, στη συνέχεια άρχισε η ενσωμάτωση front office λειτουργιών, όπως η διαχείριση πελατειακών σχέσεων (Customer

Relation Management, CRM), διαδικασίες πώλησης και τιμολόγησης, λειτουργίες e-business, όπως e-commerce, e-Telecom και e-finance, αξιοποιώντας παράλληλα και τις νέες τεχνολογικές καινοτομίες που έχει εισάγει η διαδικτυακή επανάσταση της προηγούμενης δεκαετίας.

Τα συστήματα ERP νέας γενιάς αποτελούν web-based λογισμικό που παρέχει στους εργαζόμενους και τους εταίρους (όπως οι προμηθευτές και πελάτες) σε πραγματικό χρόνο πρόσβαση σε ERP συστήματα. Με τα νέας γενιάς ERP συστήματα (αναφέρονται και ως συστήματα ERP II) επεκτείνεται η παραδοσιακή βελτιστοποίηση των πόρων του ERP και στην επεξεργασία των συναλλαγών. Αντί απλά να διαχειρίζεται την αγορά, την πώληση, ή οποιαδήποτε άλλη ενέργεια μέσα στην εταιρεία, τα συστήματα ERP νέας γενιάς αξιοποιούν τις πληροφορίες των πόρων υπό τη διαχείρισή του για να βοηθήσουν την επιχείρηση και σε άλλους τομείς, όπως κυρίως η συνεργασία με άλλες επιχειρήσεις.

Τα συστήματα ERP II είναι πιο ευέλικτα συγκριτικά με τα πρώτης γενιάς ERP, καθώς οι δυνατότητες τους δεν περιορίζονται στο στενό ενδοεπιχειρησιακό περιβάλλον, αλλά επεκτείνονται πέρα από τα εταιρικά τείχη και διευκολύνεται σημαντικά η αλληλεπίδρασή τους με άλλα συστήματα. Προς την κατεύθυνση αυτή έχει αρχίσει και η εμφάνιση των ERP δύο επιπέδων, στοχεύοντας κυρίως σε ομίλους επιχειρήσεων και σε πολυεθνικές εταιρείες με πολλαπλές θυγατρικές ανά τον κόσμο. Στο μοτίβο

των ERP δύο επιπέδων, ουσιαστικά λειτουργούν παράλληλα πολλά ERP συστήματα, ένα για κάθε επιχείρηση του ομίλου, τα οποία συνεργάζονται απρόσκοπτα με το ERP του ομίλου [6].

### **3.2 Χαρακτηριστικά των ERP– Λειτουργικές μονάδες**

Τα συστήματα ERP περιλαμβάνουν συνήθως τα ακόλουθα χαρακτηριστικά :

- Ένα ολοκληρωμένο σύστημα που λειτουργεί σε πραγματικό χρόνο (ή σχεδόν σε πραγματικό χρόνο). Αυτή η παράμετρος είναι σημαντική καθώς επιτρέπει τα αρμόδια στελέχη της επιχείρησης να έχουν πλήρη εικόνα των τεκταινομένων εντός της επιχείρησης.
- Τα συστήματα ERP μοιράζονται μια κοινή βάση δεδομένων, η οποία υποστηρίζει όλες τις επιμέρους λειτουργικές μονάδες του συστήματος και εξωτερικές εφαρμογές.
- Ενιαία εμφάνιση και συνέπεια στη λειτουργικότητα σε όλο το ERP ανεξάρτητα από ενότητα.

Η αρχιτεκτονική των συστημάτων ERP είναι συνήθως «σπονδυλωτή» (modular) γεγονός που επιτρέπει την αξιοποίηση των ενότητων εκείνων που είναι απαραίτητες για την δραστηριότητα της εταιρείας. Σε ένα σύστημα ERP γενικά

μπορούμε να εισάγουμε πολλαπλές ενότητες, από τις οποίες αναφέρουμε ενδεικτικά τις ενότητες της χρηματοοικονομικής λογιστικής, της διοικητικής λογιστικής, της διαχείρισης εφοδιαστικής αλυσίδας, της οργάνωσης της παραγωγής, της διαχείρισης έργου και της διαχείρισης πελατειακών σχέσεων [7]. Επίσης, οι μεγάλοι οργανισμοί που κατασκευάζουν ERP προσφέρουν δυνατότητες προσθήκης ιδιαίτερων ενοτήτων κατά παραγγελία, για να καλύψουν ιδιαίτερες ανάγκες που προκύπτουν σε διάφορα επιχειρηματικά σχήματα.

Τα περισσότερα συστήματα ERP ενσωματώνουν βέλτιστες πρακτικές, δηλαδή κατάλληλα τμήματα λογισμικού τα οποία αντανakλούν την ερμηνεία του κατασκευαστή σχετικά με τον πιο αποτελεσματικό τρόπο που μπορεί να εφαρμοσθεί κατά την εκτέλεση κάθε επιχειρηματικής διαδικασίας, ενώ ορισμένα από αυτά παρέχουν επιπλέον τη δυνατότητα να τροποποιήσουν την συγκεκριμένη λογική [8]. Έχει παρατηρηθεί ότι εταιρείες που εφαρμόζονται βέλτιστες πρακτικές, έχουν τη δυνατότητα να μειώσουν σημαντικά χρονοβόρες διεργασίες και μάλιστα με σημαντικά χαμηλό ρίσκο [9].

### 3.3 Εγκατάσταση και λειτουργία ERP

Η εισαγωγή ενός συστήματος ERP σε μια εταιρεία συνήθως συνεπάγεται σημαντικές αλλαγές στις μεθόδους εργασίας του προσωπικού. Για μια τέτοια μετάβαση τρεις τύποι έχουν προταθεί για την ομαλή εφαρμογή των αλλαγών: consulting, υποστήριξη και παραμετροποίηση [7]. Η πρώτη παράμετρος απαιτεί τη συνδρομή εξειδικευμένου προσωπικού στη χρήση του ERP, το οποίο θα μεταφέρει τη γνώση του, θα εκπαιδεύσει το προσωπικό της εταιρείας, και θα το υποστηρίζει για αρκετό καιρό μετά από την αρχική εγκατάσταση. Επίσης, η κατάλληλη και στοχευμένη παραμετροποίηση είναι δυνατό να βοηθήσει το ERP να συμπεριφέρεται παρόμοια με το προηγούμενο λογισμικό της εταιρείας, ώστε να διευκολύνει τη μετάβαση.

Ο χρόνος που απαιτείται για μια εταιρεία ώστε να μεταβεί από ένα απλό λογισμικό σε ένα σύστημα ERP είναι άμεση συνάρτηση του μεγέθους της επιχείρησης, του αριθμού των ενοτήτων οι οποίες θα εγκατασταθούν, της πολυπλοκότητας που διαθέτει η επιθυμητή παραμετροποίηση, του μεγέθους των αλλαγών που θα πραγματοποιηθούν, καθώς και της ετοιμότητας του πελάτη να αναλάβει και αξιοποιήσει το ERP. Στα σπονδυλωτά συστήματα ERP, οι αλλαγές μπορούν να εφαρμοστούν σταδιακά εισάγοντας την νέα τεχνολογία ανά ενότητα. Ο χρόνος εγκατάστασης και λειτουργίας ενός ERP σε μια μεγάλη επιχείρηση μπορεί να είναι αρκετά μεγάλος (14 μήνες αναφέρεται ως τυπικός χρόνος), ενώ



η ύπαρξη ενοτήτων ή υποενοτήτων κατά παραγγελία καθώς και εργασίες προσαρμογής του ERP μπορεί να αυξήσει σημαντικά τους χρόνους υλοποίησης [7].

Ωστόσο, τα πολυσύνθετα εργαλεία ERP είναι συνήθως ακριβά, και όχι πρακτικά για μικρομεσαίες επιχειρήσεις. Για την αντιμετώπιση αυτού του ζητήματος, ορισμένες εταιρείες λογισμικού αναπτύσσουν απλούστερα, και φθηνότερα εργαλεία ειδικά για τις μικρότερες εταιρείες.

### **3.4 Πλεονεκτήματα**

Το βασικό πλεονέκτημα των ERP εντοπίζεται στο γεγονός ότι επιτυγχάνεται η ενσωμάτωση όλων των διαδικασιών των επιχειρήσεων σε ένα ενιαίο σύστημα που εξοικονομεί μακροπρόθεσμα χρόνο και χρήμα. Η διοίκηση της επιχείρησης έχει τη δυνατότητα να παίρνει αποφάσεις πιο γρήγορα και με λιγότερα λάθη. Τα δεδομένα από όλα τα τμήματα είναι προσβάσιμα άμεσα από ολόκληρη την διοικητική δομή της επιχείρησης. Ορισμένα από τα πλεονεκτήματα περιλαμβάνουν

- Πρόβλεψη πωλήσεων, η οποία επιτρέπει τη βελτιστοποίηση της αποθήκης
- Ιστορικό κάθε συναλλαγής, που επιτυγχάνεται από τη συλλογή δεδομένων κάθε τομέα της επιχείρησης.
- Παρακολούθηση των παραγγελιών, από την αποδοχή της μέχρι και την ολοκλήρωσή της

- Παρακολούθηση των εσόδων, εξόδων και κερδών, από την ανάλυση στα παραστατικά πωλήσεων, αγορών, καθώς και τις εισπράξεις.

Στα ERP συστήματα συγκεντρώνονται τα στοιχεία των επιχειρήσεων, με αποτέλεσμα να παρατηρούνται τα ακόλουθα:

- Εξαλείφεται η ανάγκη του συγχρονισμού και της ενοποίησης των δεδομένων από τα διάφορα υποσυστήματα της εταιρείας
- Διευκολύνει τη νομιμότητα και τη διαφάνεια των στοιχείων
- Διευκολύνει την εφαρμογή προτυποποίησης στην επιχείρηση
- Παρέχει μια ολοκληρωμένη εικόνα των επιχειρήσεων, επιτρέποντας σε πραγματικό χρόνο τη διαχείριση της εταιρείας και την σημαντική συνδρομή στη λήψη αποφάσεων
- Προστατεύει τα ευαίσθητα δεδομένα από την ενοποίηση πολλαπλών συστημάτων ασφαλείας σε μια ενιαία δομή [36]

Ένα σύστημα ERP μπορεί να βελτιώσει σημαντικά την ποιότητα και την αποτελεσματικότητα της επιχείρησης, καθώς διατηρώντας την εσωτερική επιχειρηματική της διαδικασία, το σύστημα ERP μπορεί να οδηγήσει σε καλύτερα αποτελέσματα προς όφελός της, όπως η εξυπηρέτηση πελατών. Το ERP παρέχει λειτουργική στήριξη στο ανώτερο επίπεδο διοίκησης παρέχοντας κρίσιμες πληροφορίες για τη λήψη αποφάσεων. Η στήριξη αυτή επιτρέπει στο ανώτερο επίπεδο διοίκησης να λαμβάνει τις κατάλληλες

διαχειριστικές επιλογές οι οποίες ενισχύουν συνολικά την επιχείρηση. Το ERP προσδίδει στην εταιρεία την απαραίτητη ευελιξία που της επιτρέπει να προσαρμόζεται καλύτερα και ευκολότερα στις αλλαγές της δυναμικής παγκοσμιοποιημένης αγοράς.

### **3.5 Μειονεκτήματα**

Τα μειονεκτήματα από την χρήση των ERP μέσα σε μία εταιρεία είναι τα ακόλουθα:

- Η προσαρμογή στο περιβάλλον του ERP ενδέχεται να είναι προβληματική.
- Υπάρχει πιθανότητα να απαιτηθεί επανασχεδιασμός ορισμένων επιχειρηματικών διαδικασιών ώστε να συμβαδίζουν με το σύστημα ERP, το οποίο είναι δυνατό να βλάψει την ανταγωνιστικότητα ή να απομακρύνει την προσοχή της εταιρείας από άλλες κρίσιμες δραστηριότητες.
- Το κόστος του συστήματος ERP μπορεί να είναι σημαντικά μεγαλύτερο συγκριτικά με λιγότερο ολοκληρωμένες λύσεις.
- Το λειτουργικό κόστος ενός ERP ευρείας κλίμακας είναι δυνατό να είναι υψηλότερα όσον αφορά την λειτουργική υποστήριξη, την συντήρηση καθώς και τα έξοδα αναβάθμισης.

- Το κόστος εκπαίδευσης χρήσης ενός ERP συστήματος είναι σημαντικά υψηλότερο σε σύγκριση με άλλες λύσεις.

Η εναρμόνιση των ERP συστημάτων μπορεί να είναι ένα τιτάνιο έργο (ειδικά για τις μεγάλες επιχειρήσεις) και έχει εξαιρετικά αυξημένο χρόνο εφαρμογής και αυξημένο κόστος. [38]

## **4 Πληροφοριακά συστήματα διαχείρισης πόρων υγείας**

Τα πληροφοριακά συστήματα διαχείρισης πόρων υγείας αποτελούν την αιχμή του δόρατος των σύγχρονων συστημάτων υγείας, καθώς αποτελούν ολοκληρωμένα συστήματα πλήρους διαχείρισης τόσο μιας μικρής ιατρικής μονάδας, όσο και ενός καταναμημένου και πολυδιάστατου συστήματος όπως του γνωστού πλέον Ενιαίου Οργανισμού Παροχής Υπηρεσιών Υγείας (ΕΟΠΥΥ).

### **4.1 Σκοπός**

Ο σκοπός των ολοκληρωμένων πληροφοριακών συστημάτων υγείας είναι πολύπλευρος και πολυδιάστατος. Στη συνέχεια θα προχωρήσουμε στην ανάλυση των βασικών αξόνων στα οποία στηρίζεται ένα τέτοιο πληροφοριακό σύστημα, καθώς και το πρακτικό αντίκρισμα στην παροχή υπηρεσιών υγείας στο ευρύ κοινό.

Καταρχήν, ένα ολοκληρωμένο πληροφοριακό σύστημα υγείας μπορεί να διαθέτει πληθώρα ημί-αυτόνομων μονάδων, οι οποίες σε άλλες περιπτώσεις μπορεί να αποτελούν και αυτόνομες εφαρμογές. Τις μονάδες αυτές μπορούμε να τις διαχωρίσουμε σε δύο μεγάλες κατηγορίες: Τις μονάδες που αποσκοπούν κυρίως στην διοικητική μέριμνα ενός συστήματος υγείας, και τις μονάδες που αφορούν την πληροφορική υποστήριξη κατά την κλινική διαδικασία. Παρόλο που ο διαχωρισμός φαίνεται αυστηρός, στην πράξη, όπως θα φανεί και στη συνέχεια, μερικές φορές είναι εξαιρετικά δύσκολος, καθώς ορισμένες μονάδες μπορούν να εξυπηρετήσουν και τους δύο σκοπούς.

#### **4.1.1 Διοικητική Μέριμνα**

Τα υποσυστήματα ενός ολοκληρωμένου πληροφοριακού συστήματος υγείας που μπορούν πολύ εύκολα να καταταχθούν ως υποσυστήματα υποστήριξης της διοικητικής μέριμνας είναι:

- Σύστημα διαχείρισης ανθρώπινων πόρων: Η καρδιά ενός συστήματος υγείας, σε κάθε περίπτωση, είναι το προσωπικό του. Το σύστημα διαχείρισης ανθρώπινων πόρων αναλαμβάνει όλες εκείνες τις διαδικασίες που χρειάζονται τόσο για τυπικούς όσο και για ουσιαστικούς λόγους, ώστε το σύστημα υγείας να λειτουργεί εύκολα, απρόσκοπτα και με την μεγαλύτερη δυνατή ευελιξία, χωρίς από την άλλη πλευρά το προσωπικό να είναι δυσαρεστημένο, κουρασμένο ή τοποθετημένο σε ακατάλληλη θέση. Η αξία του

συστήματος διαχείρισης ανθρώπινων πόρων γίνεται αντιληπτή σε μεγάλες ή/και κατανεμημένες μονάδες.

- Σύστημα οικονομικής διαχείρισης: Το επόμενο βασικό ζήτημα, από διοικητική πλευρά, σε ότι αφορά την εύρυθμη λειτουργία ενός συστήματος υγείας είναι ο οικονομικός τομέας. Το σύστημα οικονομικής διαχείρισης παραλαμβάνει από όλα τα υπόλοιπα υποσυστήματα αυτοματοποιημένα όλες τις πληροφορίες που απαιτούνται για την σωστή κατάρτιση οικονομικών αναφορών, αποτύπωση οικονομικών μεγεθών, τάσεων, προβλέψεων κλπ. Η περιπλοκότητα των συστημάτων υγείας καθιστά το συγκεκριμένο σύστημα απαραίτητο: Από τη μία πλευρά οι συνηθισμένες συναλλαγές με ασθενείς, προμηθευτές, ασφαλιστικές εταιρείες συστήματα δημόσιας υγείας καθώς και από την άλλη πλευρά η διαχείριση μισθοδοσίας, και συναφών διαδικασιών (ΙΚΑ), φορολογικών υποχρεώσεων και κοινωνικής εταιρικής ευθύνης. Όλα αυτά καθιστούν το σύστημα αυτό απαραίτητο, όχι μόνο σε ένα μεγάλο οργανισμό υγείας, αλλά ακόμη και σε μια απλή τοπική μονάδα πρωτοβάθμιας φροντίδας.
- Σύστημα διασφάλισης ποιότητας: Το σύστημα διασφάλισης ποιότητας έχει διπλό ρόλο: από τη μία είναι αρμόδιο για τον καθορισμό όλων εκείνων των πρωτοκόλλων που απαιτούνται για την διασφάλιση της ποιότητας των παρεχόμενων υπηρεσιών, σύμφωνα με τις νομικές

υποχρεώσεις και την διεθνή πρακτική, ενώ από την άλλη πρέπει να ελέγχει την ορθή εφαρμογή τους από την μεριά του προσωπικού.

- Σύστημα διαχείρισης υλικού: Το σύστημα διαχείρισης υλικού, γνωστότερο στο ευρύ κοινό ως «αποθήκη», αποτελεί ένα ακόμη σύστημα διοικητικής μέριμνας για ένα πληροφοριακό σύστημα υγείας. Πιο συγκεκριμένα, στο σύστημα αυτό καταγράφονται οι απαιτούμενες πληροφορίες για κάθε υλικό αντικείμενο που απαιτείται ή χρησιμοποιείται μέσα στο σύστημα υγείας, από απλό χαρτί A4 για τους εκτυπωτές της γραμματείας, έως και εξειδικευμένες συσκευές εντός εργαστηρίων ή χειρουργείων (όπως μαγνητικός τομογράφος). Οι πληροφορίες που καταγράφονται διαφέρουν συνήθως ανάλογα με τον τύπο του υλικού. Έτσι, για παράδειγμα, στα αναλώσιμα και στα είδη που διακινούνται μεγάλες ποσότητες καταγράφεται το απόθεμα, οι εισροές και εκροές με τη συναφή αιτιολογία, ο χώρος στον οποίο βρίσκονται και πιθανόν, σε μερικά ευπαθή υλικά (όπως φάρμακα, φιλμ ακτινογραφίας και χημικά αντιδραστήρια), οι ημερομηνίες παραγωγής και λήξης. Αντίθετα, σε εξειδικευμένο ιατρικό, ή μηχανολογικό εξοπλισμό, αποθηκεύονται περισσότερες πληροφορίες για το ίδιο το υλικό (όπως ο σειριακός αριθμός, η ημερομηνία κατασκευής και η εταιρεία παραγωγής), αλλά και για την συντήρησή του.

- Σύστημα διαχείρισης ασθενών: Το σύστημα διαχείρισης ασθενών είναι στην ουσία η καρδιά σε ένα ολοκληρωμένο πληροφοριακό σύστημα υγείας. Στην πραγματικότητα το σύστημα αυτό έχει δύο όψεις: την διοικητική και την ιατρική. Από διοικητικής πλευράς, το σύστημα αυτό περιέχει όλες εκείνες τις πληροφορίες που είναι απαραίτητες από διοικητικής πλευράς, για την γνώση του ασθενούς όπως, ονοματεπώνυμο, ημερομηνία γέννησης, συγγενείς, ασφάλιση, χρεώσεις, πληρωμές κοκ. Η ιατρική πλευρά, η οποία θα μπορούσε να χαρακτηριστεί πιο κρίσιμη, αποτελεί την αποτύπωση του «πλήρους» ιατρικού φακέλου του ασθενή. Ο ιατρικός φάκελος του ασθενή αποτελεί ίσως το σημαντικότερο εργαλείο στα χέρια του ιατρικού και νοσηλευτικού προσωπικού, καθώς αποτελεί έναν καθρέφτη του ασθενή: οι πληροφορίες που περιλαμβάνονται είναι χρήσιμες κατά την εξέταση, νοσηλεία, συνταγογράφηση και γενικά κάθε συναφή ιατρική ή νοσηλευτική ενέργεια, παρέχοντας γνώση για αλλεργίες, ευπάθειες ή ιδιαίτερες καταστάσεις, οι οποίες χρήζουν διαφορετικής αντιμετώπισης. Ένα απλοϊκό, αλλά χαρακτηριστικό παράδειγμα αποτελεί η απλή ασπιρίνη: η ασπιρίνη προκαλεί σημαντικά αιματολογικά προβλήματα σε ορισμένους ασθενείς με γενετική έλλειψη ενός ενζύμου (G6PD). Με την απουσία του ηλεκτρονικού ιατρικού φακέλου, είναι πιθανό να χορηγηθεί ασπιρίνη σε ασθενή με έλλειψη G6PD, το οποίο ενδέχεται να προκαλέσει επιδείνωση της υγείας του. Στο μέλλον οι ερευνητές οραματίζονται έναν πραγματικά πλήρη ιατρικό φάκελο, ο οποίος θα δημιουργείται κατά την



γέννηση του ανθρώπου, θα ενημερώνεται σχολαστικά και επισταμένα, και θα τον ακολουθεί μέχρι και την αποδημία του από τα εγκόσμια. Είναι προφανές, ότι στη σημερινή εποχή, η έννοια «πλήρης ιατρικός φάκελος» μόνο ως ουτοπία μπορεί να εκληφθεί, για αυτό και τα εισαγωγικά.

#### **4.1.2 Κλινική Μέριμνα**

Εκτός από τον ηλεκτρονικό ιατρικό φάκελο, ο οποίος υπάγεται στο σύστημα διαχείρισης ασθενών, ένα ολοκληρωμένο πληροφοριακό σύστημα υγείας στον τομέα της κλινικής μέριμνας περιλαμβάνει και τα ακόλουθα στοιχεία:

- Συστήματα εργαστηριακής υποστήριξης: Πρόκειται για τα επιμέρους εξειδικευμένα υποστηρικτικά πληροφοριακά συστήματα που εγκαθίστανται στα εξειδικευμένα εργαστήρια, για εκτέλεση συγκεκριμένων λειτουργιών. Τα συστήματα αυτά διαχειρίζονται τον εξοπλισμό και ενημερώνουν αυτόματα τον ιατρικό φάκελο του υπό εξέταση ασθενούς με τα αποτελέσματα, μαζί με τις σχετικές ιατρικές γνωματεύσεις. Η συγκεκριμένη διαδικασία μειώνει δραματικά το λειτουργικό κόστος (καθώς δεν είναι πλέον απαραίτητη η εκτύπωση των αποτελεσμάτων), και αυξάνει τον χρήσιμο χρόνο του εργαστηριακού προσωπικού (καθώς δεν θα ασχολείται πλέον με εργασίες γραφειοκρατικής υποστήριξης)
- Συστήματα ιατρικής παρακολούθησης: Τα συστήματα ιατρικής παρακολούθησης αναλαμβάνουν την αυτοματοποιημένη παρακολούθηση ασθενών που βρίσκονται σε συγκεκριμένα τμήματα του νοσοκομείου. Παρατηρούν συγκεκριμένα μετρήσιμα μεγέθη όπως πίεση και καρδιαγγειακούς παλμούς και εγείρουν συναγερμό σε περίπτωση ιατρικής ανάγκης

- Σύστημα φαρμακείου: Το σύστημα παρακολούθησης φαρμακείου αποτελεί μια μετεξέλιξη των συστημάτων διατήρησης αποθήκης και κινήσεων αυτής, κατάλληλα προσαρμοσμένο στις ανάγκες του νοσοκομειακού περιβάλλοντος.

Όλα τα παραπάνω συστήματα, είναι πιθανό να συνυπάρχουν μέσα σε ένα ολοκληρωμένο πληροφοριακά σύστημα υγείας, αλλά δεν είναι όλα απαραίτητα.

## **4.2 Υπάρχοντα πληροφοριακά συστήματα υγείας**

Η εισαγωγή των πληροφοριακών συστημάτων υγείας στο νοσοκομειακό περιβάλλον, ξεκίνησε αρχικά από τις διοικητικές λειτουργίες και στη συνέχεια έχει επεκταθεί σταδιακά και με αργούς ρυθμούς σε ολόκληρο το φάσμα της λειτουργίας τους.

Οι πρώτες προσπάθειες στον τομέα των πληροφοριακών συστημάτων υγείας αφορούσαν κυρίως τον τομέα του ηλεκτρονικού φακέλου του ασθενούς [10], ενώ παράλληλα άρχισε και η εισαγωγή των τεχνολογιών πληροφορικής στο τμήμα της διοίκησης, για εξυπηρέτηση εσωτερικών κυρίως αναγκών.

Στη συνέχεια ξεκίνησε μια προσπάθεια για παροχή περισσότερο ολοκληρωμένων λύσεων. Για παράδειγμα, στο Ηνωμένο Βασίλειο, το πληροφοριακό σύστημα διαχείρισης πόρων υγείας αναπτύσσεται και συντηρείται από τοπικούς φορείς παροχής υπηρεσιών [11], με τη μορφή ιδιόκτητου λογισμικού. Στις ΗΠΑ, έχει αναπτυχθεί μια εξαιρετική λύση, το πληροφοριακό σύστημα για τα νοσοκομεία των βετεράνων - VISTA, το οποίο διασυνδέει άμεσα ή έμμεσα ένα μικρό αριθμό νοσοκομείων εντός των ΗΠΑ, που εξυπηρετεί συνολικά 30 εκατομμύρια ασθενείς [12]. Παρόλα αυτά, η διείσδυση των πληροφοριακών συστημάτων υγείας από τα υπόλοιπα νοσοκομεία είναι μάλλον περιορισμένη [13].

Μια προσπάθεια που αξίζει ιδιαίτερη μνεία είναι το σύστημα OpenMRS [14]. Το Open Medical Record System (OpenMRS®) είναι ένα έργο το οποίο δημιουργήθηκε το 2004 ως μια πλατφόρμα ανοιχτού λογισμικού, η οποία υλοποιεί ένα ιατρικό σύστημα καταγραφής για τις αναπτυσσόμενες χώρες. Η πλατφόρμα OpenMRS έχει αναπτυχθεί και συντηρείται από μια μη κερδοσκοπικού χαρακτήρα συνεργασία ιδρυμάτων με επικεφαλή το Ινστιτούτο Regenstrief και το ίδρυμα "Συνεργάτες στην Υγεία", μια φιλανθρωπική οργάνωση της Βοστώνης.

Το OpenMRS είναι μια πλατφόρμα λογισμικού και μια εφαρμογή αναφοράς που επιτρέπει το σχεδιασμό ενός συστήματος καταγραφής ιατρικών δεδομένων χωρίς γνώσεις προγραμματισμού (αν και οι ιατρικές γνώσεις κρίνονται μάλλον απαραίτητες). Το σύστημα βασίζεται σε μια εννοιολογική δομή στο επίπεδο της βάσης δεδομένων η οποία δεν εξαρτάται από τους πραγματικούς τύπους των ιατρικών πληροφοριών που απαιτείται να συλλέγονται ή στις ιδιαίτερες μορφές συλλογής δεδομένων και έτσι μπορεί να προσαρμοστεί για διαφορετικές χρήσεις. Το OpenMRS βασίζεται στην αρχή ότι οι πληροφορίες πρέπει να αποθηκεύονται κατά τέτοιο τρόπο ώστε η σύνοψη, η ανάλυση και η επεξεργασία των δεδομένων να είναι εύκολη διαδικασία. Στον πυρήνα του υπάρχει ένα λεξικό εννοιών που αποθηκεύει όλες τις διαγνώσεις, δοκιμές, τις διαδικασίες, τη φαρμακευτική αγωγή που έχει ακολουθηθεί και άλλες πληροφορίες σχετικές με το ιατρικό ιστορικό. Το OpenMRS είναι μια πλατφόρμα βασισμένη στο μοντέλο client - server , που

σημαίνει ότι έχει σχεδιαστεί για να λειτουργεί σε ένα περιβάλλον όπου πολλοί υπολογιστές-πελάτες έχουν πρόσβαση στις ίδιες πληροφορίες του κεντρικού διακομιστή.

Στη συνέχεια θα εξειδικευθούμε σε μελέτες για πληροφοριακά συστήματα υγείας τα οποία βασίζονται εξολοκλήρου στην τεχνολογία του ERP. Η συγκεκριμένη εξειδίκευση θεωρείται μάλλον επιβεβλημένη, γεγονός που θα εξηγηθεί στη συνέχεια.

#### **4.2.1 Πληροφοριακά συστήματα βασισμένα στο μοντέλο του ERP**

Ένα σύστημα «προγραμματισμού επιχειρηματικών πόρων» (Enterprise Resource Planning systems – ERP) αποτελεί ένα ολοκληρωμένο σύστημα επιχειρηματικής διαχείρισης το οποίο συμπεριλαμβάνει μέσα του ένα σύνολο συνεργαζόμενων υποσυστημάτων τα οποία εξυπηρετούν όλες τις πλευρές της επιχείρησης ή του οργανισμού τον οποίο υποστηρίζουν. Τα συστήματα ERP ακολουθούν συνήθως σπονδυλωτή αρχιτεκτονική, όπου βασιζόμενα σε μια κεντρική πλατφόρμα συνδέονται όλα τα επιμέρους υποσυστήματα. Στη συνηθισμένη περίπτωση τα επιμέρους υποσυστήματα είναι ανεξάρτητα μεταξύ τους, ή έχουν μη κρίσιμες διασυνδέσεις. Για τον λόγο αυτό, κάθε εγκατάσταση συστήματος ERP είναι δυνατό να περιλαμβάνει μόνο ένα υποσύνολο από τα διαθέσιμα υποσυστήματα.

Σε μια εμπορική επιχείρηση ένα σύστημα ERP συμπεριλαμβάνει κατάλληλα υποσυστήματα για την διαχείριση της επιχείρησης, την υποστήριξη της γραμμής παραγωγής, το δίκτυο των πωλήσεων και για την εμπορική πολιτική. Ένα από τα σημαντικότερα εμπορικά συστήματα ERP είναι το SAP, ενώ τα τελευταία χρόνια έχουμε ανάλογες προσπάθειες από την κοινότητα του ανοιχτού λογισμικού, με χαρακτηριστικότερο εκπρόσωπο το OpenERP.

Σε ένα νοσηλευτικό ίδρυμα το σύστημα ERP έχει ορισμένες βασικές διαφοροποιήσεις από τα συστήματα που αναπτύσσονται για χρήση σε εμπορικές επιχειρήσεις. Αυτές είναι οι διαφοροποιήσεις βασίζονται στο γεγονός ότι ορισμένοι από τους πόρους που διαχειρίζεται έχουν σχέση με την ανθρώπινη υγεία και αξιοπρέπεια. Συνεπώς, υπάρχουν αυξημένες απαιτήσεις ασφαλείας και εμπιστευτικότητας. Επιπρόσθετα, εξαιτίας της κρισιμότητας της υγείας, υπάρχουν επίσης αυξημένες απαιτήσεις σχετικά με τη συνέπεια και την αξιοπιστία. Οι υπόλοιπες λειτουργικότητες όμως, δεν διαφέρουν ιδιαίτερα από τις λειτουργικότητες που απαντώνται στις περισσότερες εμπορικές επιχειρήσεις, όπως το σύστημα διαχείρισης ανθρώπινων πόρων (προσωπικό, ιεραρχία αρμοδιότητες), σύστημα οικονομικής διαχείρισης, σύστημα διαχείρισης υλικού (μια διαφορετική μορφή του συστήματος αποθήκης των εμπορικών υπηρεσιών), σύστημα διαχείρισης ασθενών (πελάτες – προμηθευτές στις εμπορικές επιχειρήσεις).

Όσον αφορά τον τομέα των πληροφοριακών συστημάτων υγείας, πρόσφατες μελέτες έχουν συνταχθεί για την υιοθέτηση της ενσωμάτωσης των τεχνολογιών ERP ώστε να είναι εφαρμόσιμες στις μονάδες υγείας. Τα πλεονεκτήματα μιας τέτοιας επιλογής είναι πολλαπλά:

- Τα συστήματα που βασίζονται στο μοντέλο ERP είναι εκ φύσεως σπονδυλωτά (modular) και διαμοιράζονται τα ίδια δεδομένα. Συνεπώς, είναι μια σημαντική προσπάθεια λειτουργίας διαφορετικών συστημάτων σε κοινή βάση, με κοινά δεδομένα τα οποία αλληλεπιδρούν κατάλληλα μεταξύ τους. Ως εκ τούτου, καθιστούν ευκολότερη την διασύνδεση τους
- Υποστηρίζουν εξαρχής όλες τις διαδικασίες που χρειάζεται ένας οργανισμός παροχής υπηρεσιών υγείας, προσφέροντας ενιαίο περιβάλλον λειτουργίας.
- Τα δεδομένα μεταξύ των μονάδων είναι όλα αποθηκευμένα σε κοινή βάση δεδομένων, συνεπώς εξοικονομείται σημαντικός χρόνος για την αντιγραφή τους από το ένα σύστημα στο άλλο.
- Τα συστήματα με κοινή βάση δεδομένων προσφέρονται για την εφαρμογή τεχνικών υπολογιστικής νοσημοσύνης για την εξαγωγή χρήσιμων συμπερασμάτων, τόσο ιατρικής όσο και γενικότερης φύσης αλλά και στη δημιουργία νέας γνώσης



- Αποτελούν επίσης, αρχείο με ιδιαίτερη αξία και σημασία, καθώς μεταφέρει την γνώση μεταξύ του ιατρικού και νοσηλευτικού προσωπικού.

Στη μελέτη [15] αναλύεται η διαδικασία εφαρμογής των ERP συστημάτων στα νοσοκομεία, ως οργανισμός με διαφορετικές και ετερογενείς λειτουργικές περιοχές. Ο στόχος της μελέτης είναι να επισημανθεί η δυνατότητα εξυπηρέτησης από ένα ERP σύστημα των βασικών στόχων που τίθενται για την ικανοποιητική λειτουργία των οργανισμών αυτών, καθώς και να επισημανθούν οι ελλείψεις που έχουν. Τα αποτελέσματα μπορούν να παρέχουν καθοδήγηση για την επίλυση προβλημάτων κατά την εφαρμογή των συστημάτων ERP σε οργανισμούς στους οποίους υπάρχουν λειτουργικές περιοχές οι οποίες διαφοροποιούνται σε μεγάλο βαθμό. Τα αποτελέσματα της έρευνας έδειξαν ότι στην ανάλυση του βαθμού ικανοποίησης των χρηστών του συστήματος ERP, στην περίπτωση του κλινικού προσωπικού αγγίζει σχεδόν το 100% των προσδοκιών τους, ενώ οι υπεύθυνοι για τη διαχείριση του νοσοκομείου είναι μάλλον λιγότερο ικανοποιημένοι, δεδομένου ότι δεν έχουν δει τις αρχικές τους προσδοκίες να πληρούνται όσον αφορά το σύστημα, καθώς οι διαδικασίες λογιστικής και τιμολόγησης ήταν ανεπαρκείς. Μια επιπλέον πιθανή εξήγηση βασίζεται στην ετερογένεια των δύο αυτών ομάδων.

Μια παρόμοια μελέτη διεξάγεται και στην εργασία [16], χρησιμοποιώντας μια μεγάλη νοσοκομειακή μονάδα για εφαρμογή του γνωστού SAP. Η μελέτη χρησιμοποιεί μια ποιοτική μεθοδολογία, με τη βοήθεια φοιτητών και του προσωπικού του νοσοκομείου. Οι φοιτητές ήταν εξοικειωμένοι με συστήματα ERP. Το ERP της συγκεκριμένης μελέτης έχει επιφέρει θεμελιώδεις αλλαγές στη δομή του νοσοκομείου, την νοοτροπία εργασίας μέσα σε αυτό, καθώς και τις πρακτικές διαχείρισης του. Το ERP παρουσίασε πολλά πλεονεκτήματα σε σχέση με τις άλλες προσεγγίσεις.

### **4.3 Συμπεράσματα**

Οι τεχνολογίες πληροφορικής και επικοινωνιών, με αρκετά αργά, αλλά σταθερά βήματα, εισάγονται σταδιακά στον τομέα της υγείας και της περίθαλψης. Η συγκεκριμένη αργοπορία, αν και από ορισμένους τεχνολάτρεις μπορεί να θεωρηθεί ως συντηρητισμός, αποτελεί φυσιολογική αντίδραση στην αλλαγή, καθώς τα δεδομένα αποδεικνύουν μια εντελώς διαφορετική φιλοσοφία. Η ιατρική πληροφορική είναι ένας κλάδος που έχει αναπτυχθεί παράλληλα με αυτόν της πληροφορικής, και αρχίζει σχετικά άμεσα τις διαδικασίες υιοθέτησης ορισμένων νέων τεχνολογιών. Αυτό φαίνεται από το γεγονός ότι, τουλάχιστον σε διεθνές επίπεδο, τα νοσοκομειακά περιβάλλοντα, στη διοικητική πλευρά, υιοθέτησαν σχεδόν από την αρχή τις τεχνολογίες πληροφορικής. Στον τομέα βέβαια της παροχής υγείας, δεν πρέπει να λησμονούμε ότι απαιτούνται ιδιαίτερα επίπονες

προσπάθειες για την επίτευξη της επιθυμητής ακρίβειας και αξιοπιστίας που πρέπει να υπακούν τα συγκεκριμένα συστήματα (η οποία είναι αρκετά πιο αυστηρή σε σύγκριση με τα συνηθισμένα εμπορικά συστήματα).

Ένα μεγάλο πρόβλημα που εντοπίζεται αυτήν την περίοδο στα πληροφοριακά συστήματα υγείας εντός νοσοκομειακών περιβαλλόντων είναι η ελλείψεις επικοινωνία μεταξύ διαφορετικών συστημάτων. Οι περισσότεροι οργανισμοί έχουν αναπτύξει ανεξάρτητα πληροφοριακά συστήματα σε διαφορετικές χρονικές περιόδους και συνήθως ασύμβατα μεταξύ τους. Συνεπώς, η πρόκληση στο εγγύς μέλλον είναι η πλήρης ενοποίηση των διαφόρων νοσοκομειακών συστημάτων, σε ένα ενιαίο σύστημα το οποίο θα μπορεί να ακολουθεί τον κάθε άνθρωπο από την ημέρα της γέννησής του, με ένα πλήρες και ενημερωμένο ιατρικό ιστορικό.

Οι τεχνολογίες που υιοθετούνται στα συστήματα ERP τείνουν προς τον στόχο αυτό, καθώς αποτελούν μια σημαντική προσπάθεια ενοποίησης ετερογενών συστημάτων εντός του ίδιου νοσοκομειακού περιβάλλοντος.

## **5 Test case**

### **5.1 Εισαγωγή**

Το test case στο οποίο θα στηριχθούμε στην συγκεκριμένη εργασία αφορά μια πρώτη προσπάθεια ενοποίησης του συστήματος παραπεμπτικών. Η παρούσα κατάσταση στα περισσότερα Ελληνικά δημόσια νοσοκομεία παρουσιάζεται μη-ηλεκτρονική. Η προσπάθεια που καταβάλλεται στο επίπεδο αυτό, αν και είναι σε πρωταρχικό στάδιο, υλοποιεί εκείνες τις τεχνολογίες αιχμής που θα επιτρέψουν με εύκολο τρόπο και με ταχεία υλοποίηση όποια επέκταση μπορεί να απαιτηθεί.

### **5.2 Περιγραφή**

Το σενάριο το οποίο θα υλοποιήσουμε στην παρούσα εργασία αφορά τον σχεδιασμό και την υλοποίηση ενός συστήματος αυτόματης διεκπεραίωσης ραντεβού για περαιτέρω εξετάσεις μέσα στον ίδιο φορέα υγείας. Πιο συγκεκριμένα, υπάρχει μια πληθώρα ασθενών σε ένα φορέα υγείας (όπως για παράδειγμα ένα νοσοκομείο) όπου χρειάζεται να περάσουν από μια αλληλουχία εξετάσεων. Έτσι, απαιτείται ένα κατάλληλο σύστημα διαχείρισης, επεξεργασίας και αναπροσαρμογής όλων των απαραίτητων σταδίων από την προσέλευση του ασθενή στο νοσοκομείο μέχρι και την τελική αποχώρησή τους.

Μέσα σε αυτή τη διαδικασία, οι ασθενείς διέρχονται από γιατρό και εν συνεχεία είναι πιθανό να υποβληθούν σε εξετάσεις και να επανεξεταστούν από τους ίδιους ή άλλους γιατρούς. Το σύνολο της συγκεκριμένης διαδικασίας μπορεί να διεκπεραιώνεται σε ορισμένες ώρες, ή είναι πιθανό να διαρκεί πολλές εβδομάδες (ιδιαίτερα σε σοβαρές ή χρόνιες περιπτώσεις), ενώ είναι συνηθισμένο το φαινόμενο να χρειάζεται πληθώρα εξετάσεων, οι οποίες εν γένει μπορούν να πραγματοποιηθούν παράλληλα.

Αντικειμενικός στόχος του συστήματος είναι ο σχεδιασμός και η υλοποίηση ενός συστήματος το οποίο θα συνδράμει τη σχετική διαδικασία

### **5.3 Σχεδιασμός του συστήματος**

Ο σχεδιασμός του συστήματος αρχίζει με τη συλλογή των απαιτήσεων της εφαρμογής της οποίας επιθυμούμε την ανάπτυξη. Πιο συγκεκριμένα, οι απαιτήσεις είναι οι ακόλουθες:

- Με την είσοδό του στο νοσοκομείο, ο ασθενής λαμβάνει μοναδικό αναγνωριστικό αριθμό (πιθανόν με χρήση κάρτας), ο οποίος τον συνοδεύει μέχρι το πέρας των όλων των διαδικασιών
- Αρχικώς, ο ασθενής προωθείται σε έναν από τους γενικούς ιατρούς, οι οποίοι κρίνουν για την πιθανή περαιτέρω παραπομπή, νοσηλεία ή έξοδο. Στο σημείο αυτό απαιτείται

κατάλληλο σύστημα προτεραιοτήτων ώστε τα πραγματικά επείγοντα περιστατικά να εξυπηρετούνται άμεσα, ενώ παράλληλα ο κάθε ασθενής να έχει ελάχιστο χρόνο αναμονής

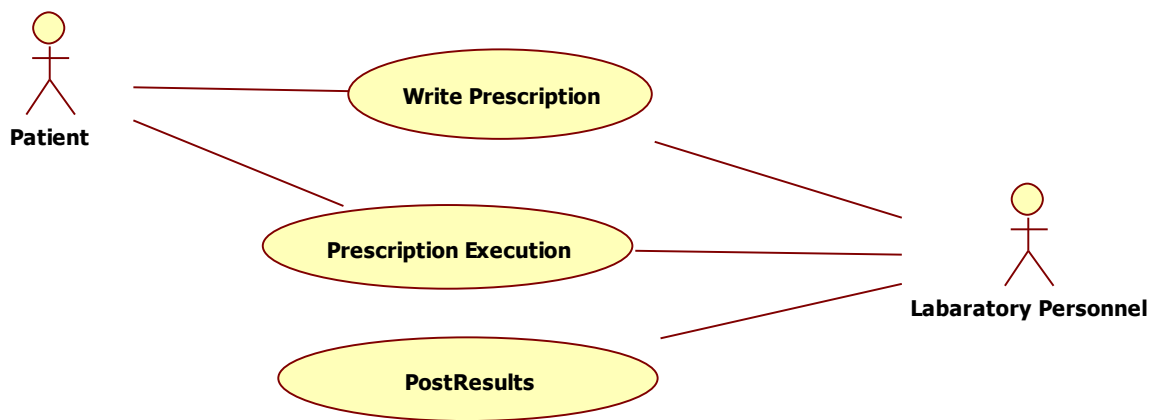
- Ο γενικός ιατρός συγγράφει στο ηλεκτρονικό αρχείο του ασθενούς (ηλεκτρονικός ιατρικός φάκελος), όλα εκείνα τα στοιχεία που θεωρεί σημαντικά και, αν είναι απαραίτητο, παραπέμπει τον ασθενή σε ειδικευμένο ιατρό ή για εξετάσεις, ή και τα δύο. Σε κάθε περίπτωση, κατά την σύνταξη του παραπεμπτικού, αυτόματα δίνεται στον ασθενή η προτεραιότητά του στο ραντεβού για κάθε ειδικό ιατρό ή εξέταση. Στην περίπτωση πολλαπλής παραπομπής, πρέπει να λαμβάνεται κατάλληλη μέριμνα ώστε να μην υπάρχει επικάλυψη.

Με βάση τις παραπάνω απαιτήσεις, προχωρούμε στον σχεδιασμό της βάσης δεδομένων που χρειάζεται για την συγκεκριμένη εφαρμογή.

Αρχικά δημιουργούμε τους τρεις πίνακες που απαιτούνται από το σύστημα διαχείρισης και ταυτοποίησης χρηστών, για τους χρήστες, για τους ρόλους και τη σύνδεση μεταξύ τους: User, Role και User\_Role.

Το νοσοκομείο το χωρίζουμε σε Εργαστήρια και Εξετάσεις. Όλες οι ενέργειες επί του ασθενούς μπορούν να χαρακτηριστούν ως κάποιου είδους εξέταση. Έτσι, όταν ο ασθενής εισέρχεται στο νοσοκομείο, εγγράφεται από τη γραμματεία (επεξεργασία του Ασθενή - πίνακας Person), και γράφεται αυτόματα ένα «παραπεμπτικό» για εξέταση από γενικό ιατρό. Κάθε παραπομπή από εκεί και πέρα, πραγματοποιείται μέσω του ίδιου συστήματος.

Στην Εικόνα 1 παρουσιάζεται το διάγραμμα ενεργειών χρήστη σε μορφή UML, το οποίο συνοψίζει τις παραπάνω διαδικασίες



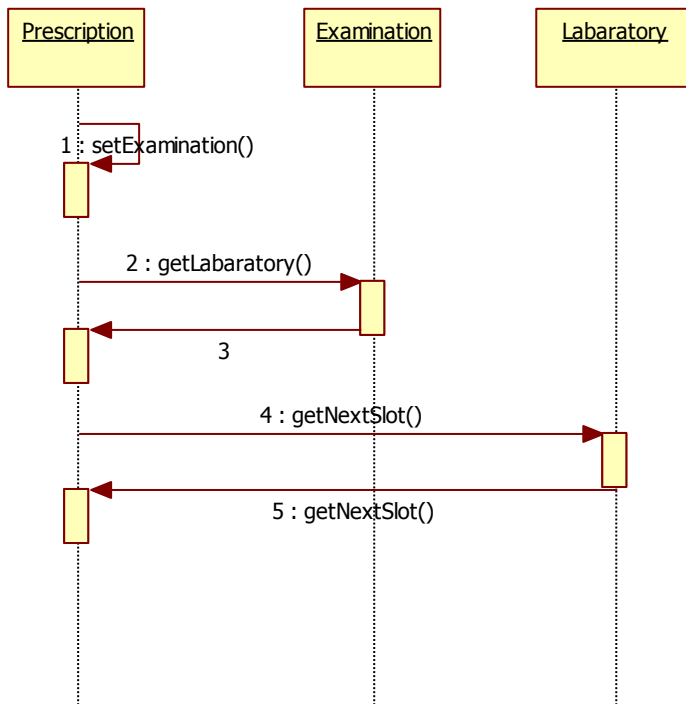
**Εικόνα 1: User-Action Diagram**

Αναλύοντας τις παραπάνω λειτουργίες, οι βασικές οντότητες που προκύπτουν είναι οι ακόλουθες:

- **Οντότητα ασθενή:** Η οντότητα του ασθενή περιλαμβάνει όλα τα χαρακτηριστικά που χρειάζεται για την απρόσκοπτη λειτουργία του συστήματος. Στα πλαίσια της παρούσας σχεδίασης, η βασική καρτέλα του ασθενή περιλαμβάνει μόνο τα βασικά και απαραίτητα στοιχεία του, ενώ το ιστορικό του οικοδομείται σταδιακά μέσα από τις εξετάσεις που θα εκτελεί.
- **Οντότητα Εργαστηρίου:** Η οντότητα του εργαστηρίου περιλαμβάνει λεπτομέρειες του εργαστηρίου, όπως το ωράριο λειτουργίας, και μια λίστα από τις εξετάσεις που μπορεί να εκτελέσει το εργαστήριο.
- **Οντότητα παραπεμπτικού:** Το παραπεμπτικό περιλαμβάνει όλες τις απαραίτητες πληροφορίες για την απαιτούμενη εξέταση και εσωκλείει επιπλέον και τα αποτελέσματά της. Μέσα από τα αποτελέσματα των παραπεμπτικών, οικοδομείται σταδιακά το ιστορικό του ασθενούς.

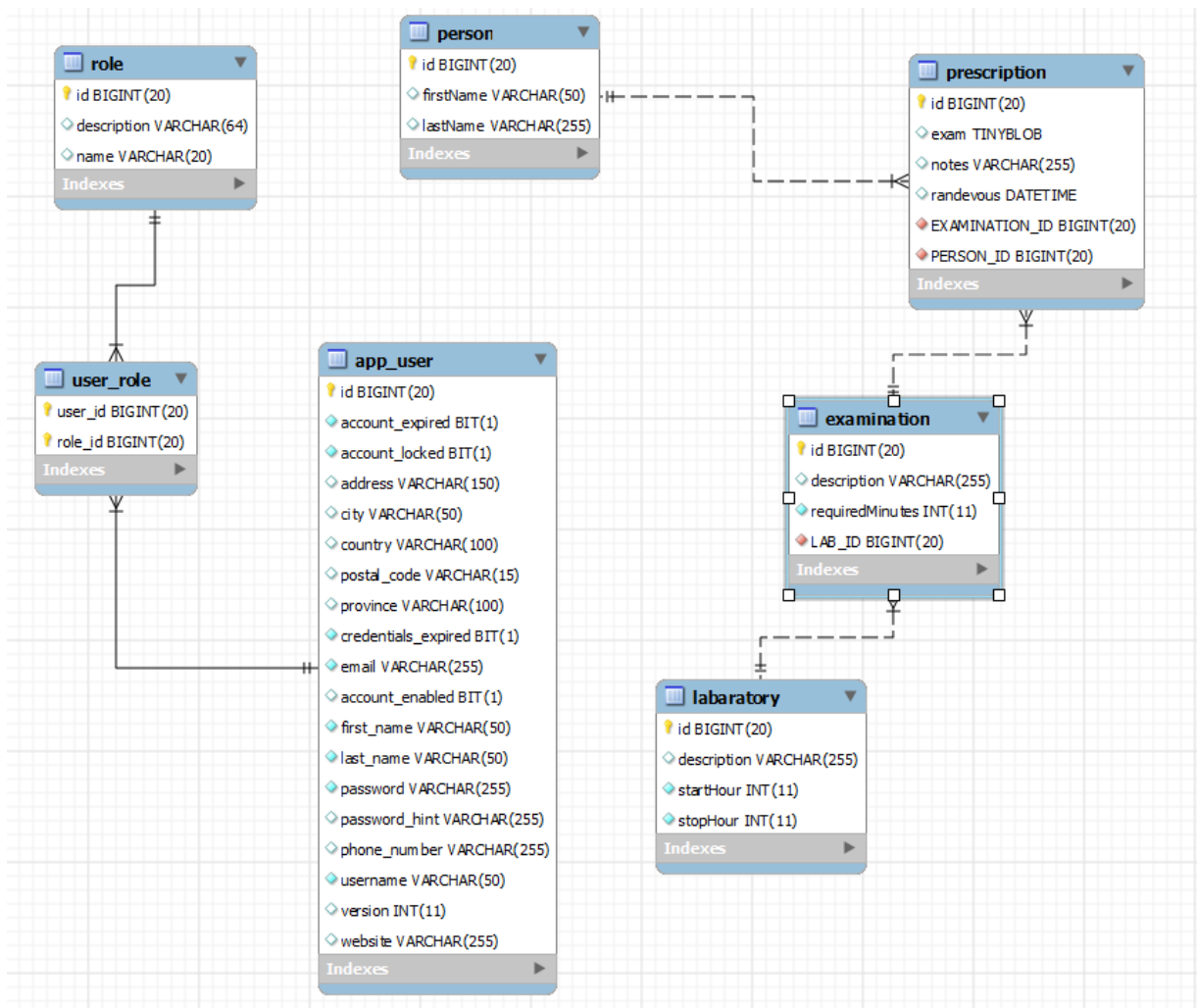
Η βασική λειτουργία της εφαρμογής είναι το κλείσιμο του ραντεβού. Στην Εικόνα 2 παρουσιάζεται αναλυτικά το διάγραμμα ροής της συγκεκριμένης λειτουργίας. Αφού ο γιατρός αποφασίσει την εξέταση που πρέπει να υποβληθεί ο ασθενής, το σύστημα αυτόματα λαμβάνει μέσω του συνδεδεμένου εργαστηρίου, την επόμενη δυνατή ημερομηνία πραγματοποίησης της εξέτασης.





**Εικόνα 2: Διάγραμμα ροής για την ενέργεια κλεισίματος ραντεβού**

Το διάγραμμα στην Εικόνα 3 παρουσιάζει το σχήμα της βάσης που εξυπηρετεί το παραπάνω πρόβλημα. Το διάγραμμα είναι κανονικοποιημένο σύμφωνα με την τρίτη κανονική μορφή. Το συγκεκριμένο διάγραμμα αποτελεί και το διάγραμμα κλάσεων του μοντέλου της εφαρμογής σύμφωνα με το πρότυπο MVC που θα αναπτύξουμε στη συνέχεια.



**Εικόνα 3:Το σχήμα της βάσης που προκύπτει**

## **6 Υλοποίηση του συστήματος**

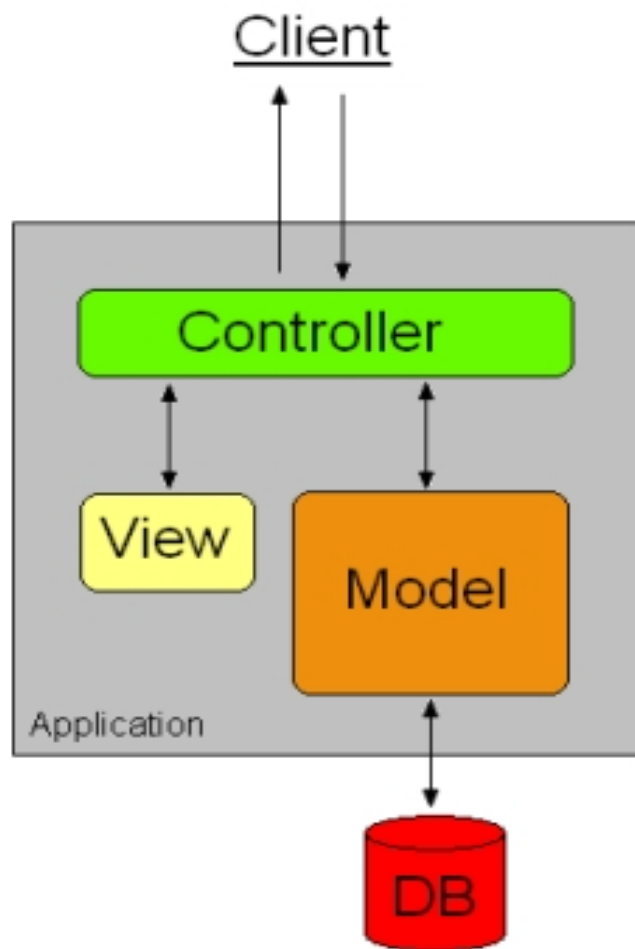
Πριν παρουσιάσουμε την υλοποίηση του συστήματος, θα επικεντρωθούμε αρχικά στις τεχνολογίες που έχουν υιοθετηθεί κατά την δημιουργία της συγκεκριμένης εφαρμογής. Στη συνέχεια παρουσιάζεται η υλοποίηση του συστήματος

### **6.1 Λογισμικό και τεχνολογίες που έχουν υιοθετηθεί**

Η βασική τεχνολογία πάνω στην οποία έχει στηριχθεί η ανάπτυξη της εφαρμογής είναι η Μεθοδολογία ανάπτυξης Μοντέλου - Όψης - Ελεγκτή, η οποία χρησιμοποιείται ευρέως στην ανάπτυξη διαδικτυακών εφαρμογών. Η μεθοδολογία αυτή έχει υλοποιηθεί με χρήση της γλώσσας Java και του πλαισίου Spring MVC. Η αρχικοποίηση της εφαρμογής γίνεται με το Appfuse, ενώ η υποστηρικτική βάση αποθηκεύεται σε MySQL.

### **6.1.1 Μεθοδολογία ανάπτυξης Μοντέλου – Όψης – Ελεγκτή**

Η μεθοδολογία Model - View - Controller (MVC) είναι ένα αρχιτεκτονικό μοτίβο που χρησιμοποιείται στην τεχνολογία λογισμικού για το σχεδιασμό και την ανάπτυξη των εφαρμογών. Η τεχνική αυτή χρησιμοποιείται συχνά από εφαρμογές που απαιτούν τη δυνατότητα να παρουσιάζουν στο χρήστη πολλαπλές προβολές των ίδιων δεδομένων. Το MVC περιγράφηκε για πρώτη φορά το 1979 αρχικά για τις διεπαφές χρήστη σε Smalltalk από τον Trygve Reenskaug [17]. Στόχος του MVC είναι ένας ευέλικτος σχεδιασμός του προγράμματος που θα διευκολύνει τις μελλοντικές τροποποιήσεις ή επεκτάσεις του και θα δίνει την δυνατότητα επαναχρησιμοποίησης όλων ή ορισμένων εξαρτημάτων του. Μια γενική αναπαράσταση του τρόπου διεξαγωγής του σχεδιασμού MVC φαίνεται στην παρακάτω εικόνα:



**Εικόνα 4: Γενική αναπαράσταση του μοντέλου MVC**

Το πρότυπο αυτό διαχωρίζει την εφαρμογή κατά την μοντελοποίηση, σε τρεις διαφορετικές έννοιες οι οποίες περιγράφονται αναλυτικά στη συνέχεια:

- Το μοντέλο (Model): Το μοντέλο διαχειρίζεται την συμπεριφορά και τη διασύνδεση της εφαρμογής με τη βάση δεδομένων, ενώ ανταποκρίνεται σε αιτήματα για λήψη πληροφοριών σχετικά με την κατάσταση του (συνήθως από

την όψη) και στις οδηγίες για αλλαγή των δεδομένων (συνήθως από τον ελεγκτή).

- Την όψη/προβολή (View): Η όψη/προβολή αναφέρεται στην αισθητική εμφάνιση των πληροφοριών.
- Τον ελεγκτή (Controller): Ο ελεγκτής ερμηνεύει τις εισόδους του χρήστη φροντίζοντας να ενημερώνει το μοντέλο και την προβολή ή και τα δυο μαζί για να αλλάξουν ανάλογα με την περίπτωση.

#### **6.1.1.1 Σχέσεις μεταξύ των συνιστωσών**

##### ***Προβολή σχέση ελεγκτή***

Στην παραδοσιακή μεθοδολογία Smalltalk MVC οι όψεις και ελεγκτές είναι στενά συνδεδεμένες. Κάθε όψη συνδέεται με ένα ενιαίο και μοναδικό ελεγκτή και αντιστρόφως. Ο ελεγκτής χρησιμοποιεί την όψη/προβολή για την είσοδο των δεδομένων, ενώ η όψη/προβολή είναι επίσης υπεύθυνη για τη δημιουργία νέων όψεων και των ελεγκτών. Είναι λογικό ότι οι όψεις και ελεγκτές είναι τμήματα που είναι στενά συνδεδεμένα μεταξύ τους όπως ακριβώς η είσοδος και έξοδος σε ένα υπολογιστικό σύστημα. Στις περισσότερες γραφικές διεπαφές (GUI) που έχουν υλοποιηθεί με βάση τις αρχές του MVC, η προβολή και ο ελεγκτής μπορούν απλά να συγχωνευτούν σε ένα αντικείμενο. Αυτό ονομάζεται προβολή εγγράφων.

## **Μοντέλο-Όψη**

Η όψη εξαρτάται από το μοντέλο. Αλλαγές στη διασύνδεση μοντέλο απαιτεί παράλληλες αλλαγές στην όψη. Είναι πολύ δύσκολο να επιτευχθεί ένας αυστηρός διαχωρισμός μεταξύ του μοντέλου. Για παράδειγμα σκεφτείτε την απαίτηση “Εμφάνισε τα αρνητικά υπόλοιπα με κόκκινο χρώμα”. Με μια πρώτη ματιά αυτό φαίνεται να είναι αυστηρά μια απαίτηση εξόδου και ένα τεστ που μπορεί να τοποθετηθεί στην προβολή σε περίπου αυτή την μορφή:

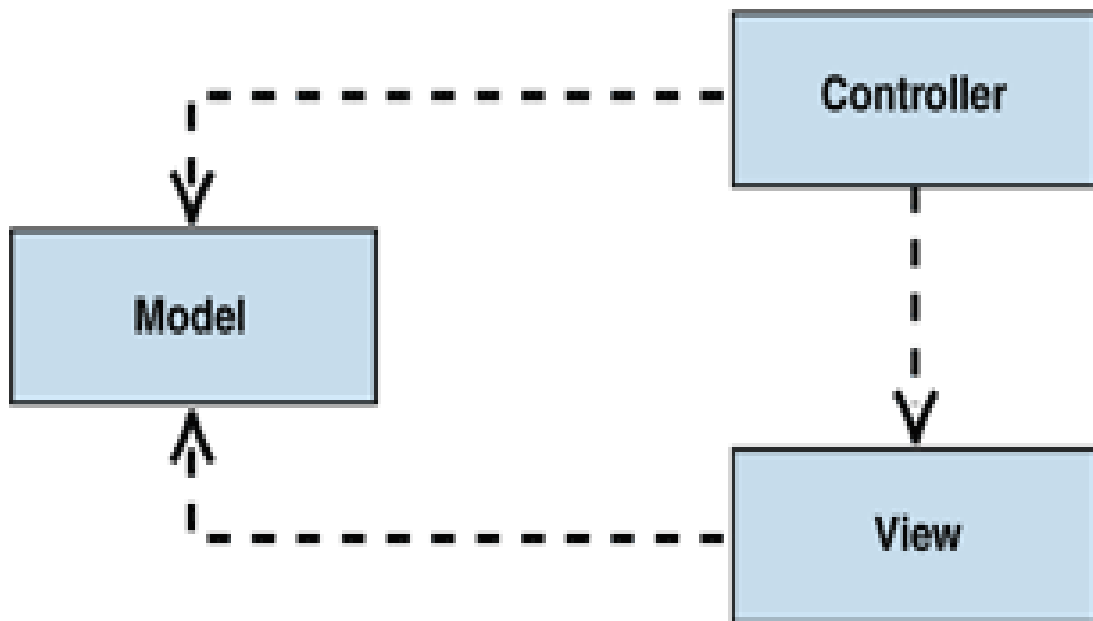
*If balance < 0 then red*

Αυτό θα παραβίαζε τον διαχωρισμό των περιορισμών του προτύπου MVC. Μετά από περεταίρω ανάλυση αποδεικνύεται ότι η πραγματική απαίτηση είναι “εμφάνισε τα ακάλυπτα υπόλοιπα σε κόκκινο” και ο ορισμός του «ακάλυπτος = (ισοζύγιο < 0)» πρέπει να τοποθετηθεί στο μοντέλο.,

## **Μοντέλο σχέση ελεγκτή**

Ο υπεύθυνος επεξεργασίας ελεγκτής έχει άμεση εξάρτηση από το μοντέλο. Έτσι, αλλαγές στη διασύνδεση του μοντέλου μπορεί να προκαλέσει παράλληλες αλλαγές στον ελεγκτή.

Η εικόνα που ακολουθεί απεικονίζει τη μεταξύ τους δομική σχέση των τριών αυτών αντικειμένων.



Εικόνα 5 MVC δομή

Πρέπει να σημειώσουμε στο σημείο αυτό ότι τόσο η όψη όσο και ο ελεγκτής εξαρτώνται από το μοντέλο. Ωστόσο το μοντέλο δεν εξαρτάται ούτε από την όψη ούτε από τον ελεγκτή και αυτό είναι ένα από τα βασικά πλεονεκτήματα του διαχωρισμού αυτού: επιτρέπει στο μοντέλο που θα κατασκευαστεί να δοκιμαστεί ανεξάρτητα από την οπτική παρουσίαση. Ο διαχωρισμός αυτός μεταξύ της όψης και του ελεγκτή δεν είναι αυστηρά καθορισμένος σε πολλές εφαρμογές επιτραπέζιων υπολογιστών. Από την άλλη πλευρά σε εφαρμογές Web ο διαχωρισμός μεταξύ της προβολής και του ελεγκτή είναι πολύ καλά καθορισμένος.



Το Model – View - Controller είναι ένα θεμελιώδες πρότυπο για τον διαχωρισμό της λογικής διεπαφής χρήστη από την λογική της εφαρμογής. Η δημοτικότητα του πρότυπου έχει οδηγήσει σε μια σειρά από διαφορετικές περιγραφές. Ειδικά ο όρος “Ελεγκτής” έχει χρησιμοποιηθεί για να σημαίνει διαφορετικά πράγματα σε διαφορετικά πλαίσια. Ευτυχώς οι εφαρμογές Web έχουν βοηθήσει στην επίλυση ορισμένων ασαφειών, διότι ο διαχωρισμός μεταξύ της όψης και του ελεγκτή είναι πρόδηλη.

#### **6.1.1.2 Πλεονεκτήματα από τη χρήση του MVC**

1. Υποστηρίζει πολλαπλές όψεις. Επειδή η κάθε όψη είναι διαχωρισμένη από το μοντέλο και δεν υπάρχει άμεση εξάρτηση από αυτό, το γραφικό περιβάλλον που παρουσιάζεται στο χρήστη μπορεί να τροποποιείται πολύ απλά, χρησιμοποιώντας πολλαπλές όψεις των ίδιων δεδομένων. Για παράδειγμα πολλές σελίδες σε μια εφαρμογή Web μπορεί να χρησιμοποιούν το ίδιο μοντέλο αντικειμένων. Άλλο παράδειγμα είναι μια εφαρμογή Web που επιτρέπει στον χρήστη την αλλαγή στην εμφάνιση των σελίδων. Αυτές οι σελίδες εμφανίζουν τα ίδια στοιχεία από το κοινό μοντέλο αλλά τα δείχνουν με διαφορετικό τρόπο.
2. Επιτρέπει αλλαγές στις όψεις. Οι απαιτήσεις της διεπαφής χρήστη τείνουν να αλλάζουν ταχύτερα από τη λογική της εφαρμογής. Οι χρήστες μπορούν να επιλέγουν διαφορετικά

χρώματα, γραμματοσειρές, διάταξη της οθόνης, και τα επίπεδα στήριξης για τις νέες συσκευές όπως κινητά τηλέφωνα ή PDAs. Επειδή το μοντέλο δεν εξαρτάται από τις όψεις, προσθέτοντας νέους τύπους όψεων, η λειτουργικότητα του συστήματος δεν τροποποιείται.

#### **6.1.1.3 Μειονεκτήματα του MVC**

1. Στενή διασύνδεση μεταξύ των όψεων και των ελεγκτών για το μοντέλο: Αλλαγές στη διασύνδεση μοντέλο ενδέχεται να προκαλέσουν παράλληλες αλλαγές στην όψη και μπορεί να απαιτηθούν πρόσθετες αλλαγές στον ελεγκτή.
2. Μεγάλη σύζευξη μεταξύ όψης και ελεγκτή: Ο αυστηρός διαχωρισμός είναι δύσκολος αν όχι αδύνατος.

#### **6.1.1.4 Αρχιτεκτονική MVC – Πλατφόρμες MVC**

Παρά το γεγονός ότι το MVC κατά κανόνα συνδέεται με πλατφόρμες είναι ουσιαστικά μια αρχιτεκτονική ανάπτυξης εφαρμογών. Αυτό σημαίνει ότι μπορεί να εφαρμοστεί ακόμη και χωρίς τη χρήση μιας αντικειμενοστραφούς γλώσσας προγραμματισμού ή μιας συγκεκριμένης ιεραρχίας κλάσεων.

Η Smalltalk MVC εφαρμογή ενέπνευσε πολλές MVC πλατφόρμες ανάπτυξης τόσο παραδοσιακών εφαρμογών για επιτραπέζιους υπολογιστές όσο και πλατφόρμες για ανάπτυξη δικτυακών εφαρμογών. Μερικές από τις πιο γνωστές πλατφόρμες που υλοποιούν το πρότυπο MVC είναι οι ακόλουθες:

- AppFlower
- XPages
- GNUstep
- GTK+
- JFace
- Oracle Application
- Microsoft Foundation Class Library
- Microsoft ASP.NET
- Microsoft Composite Block
- Qt
- Java Swing
- Apache Pivot
- Apache Struts
- Apache Tapestry
- Adobe Flex
- CakePHP
- CodeIgniter
- Zend Framework

Η απήχηση του μοντέλου ανάπτυξης εφαρμογών MVC για διαδικτυακές εφαρμογές είναι τεράστια. Οι προαναφερθείσες πλατφόρμες που το εφαρμόζουν καλύπτουν μια τεράστια γκάμα τεχνολογιών, τόσο ανοιχτού κώδικα όσο και εμπορικές, από απλές γλώσσες (Groovy, Visual Basic) έως εξαιρετικά πολύπλοκες σαν τις C# και Java. Μεγάλη αποδοχή φαίνεται να έχει στην κοινότητα της PHP, (υπάρχουν περισσότερα από 6 ενεργά εργαλεία ανοιχτού κώδικα), όσο και στην πλατφόρμα της C#, όπου η Microsoft φαίνεται να επενδύει ολοένα και περισσότερο στην συγκεκριμένη τεχνολογία για ανάπτυξη διαδικτυακών εφαρμογών.

Όσον αφορά τη Java επίσης, σημαντικές συνεισφορές παρατηρούνται. Στα πλαίσια της παρούσας εργασίας, θα επικεντρωθούμε στην πλατφόρμα MVC, η οποία είναι προσανατολισμένη σε Java.

### 6.1.2 MySQL

Η MySQL [18] αποτελεί ίσως την πιο συχνά χρησιμοποιούμενη σχεσιακή βάση δεδομένων (RDBMS) για δικτυακές εφαρμογές σε ολόκληρη την υφήλιο για μικρές και μεσαίες εγκαταστάσεις. Λειτουργεί ως ένας τυπικός διακομιστής (server) ο οποίος έχει τη δυνατότητα να παρέχει ταυτόχρονα πρόσβαση σε πολλούς χρήστες στις βάσεις δεδομένων που εξυπηρετεί.

Η βάση δεδομένων MySQL αναπτύσσεται με διπλή άδεια: η πρώτη μορφή είναι αυτή του ελεύθερου λογισμικού/ λογισμικού ανοιχτού κώδικα (ΕΛ/ΛΑΚ). Η MySQL διατίθεται στο κοινό σύμφωνα με τους όρους που επιβάλλει η άδεια χρήσης GNU General Public License. Η δεύτερη μορφή είναι αυτή της εμπορικής άδειας, όπου παρέχεται και επίσημη υποστήριξη. Η MySQL επί σειρά ετών αποτελούσε ιδιοκτησία μιας μη κερδοσκοπικής εταιρείας, της σουηδικής εταιρείας MySQL AB και η οποία αποτελούσε τον βασικό χρηματοδότη. Σχετικά πρόσφατα, η εταιρεία αυτή εξαγοράστηκε από την Sun και πλέον ανήκει στον όμιλο εταιρειών της Oracle Corporation. Η MySQL χρησιμοποιείται σε πολλά υψηλού προφίλ, μεγάλης κλίμακας World Wide Web προϊόντα, με χαρακτηριστικά παραδείγματα της διάσημης Wikipedia, της Google (όχι όμως για το τμήμα που αναφέρεται στις αναζητήσεις) και των κοινωνικών δικτύων του Facebook και του Twitter.

Η MySQL έχει αναπτυχθεί στις γλώσσες προγραμματισμού C και C++ ενώ ο μεταφραστής της γλώσσας SQL έχει αξιοποιήσει τον yacc. Η MySQL λειτουργεί σε πολλές και διαφορετικές

πλατφόρμες, μεταξύ των οποίων είναι οι ακόλουθες: AIX , BSDi , FreeBSD , HP-UX , eComStation , i5/OS , IRIX , Linux , Mac OS X , Microsoft Windows , το NetBSD , Novell NetWare , OpenBSD , OpenSolaris , OS / 2 Warp, QNX , Solaris , Symbian , SunOS , OpenServer SCO , SCO UnixWare , Sanos και Tru64.

Πολλές γλώσσες προγραμματισμού με ειδικά APIs περιλαμβάνουν τις κατάλληλες βιβλιοθήκες για την πρόσβαση σε βάσεις δεδομένων MySQL, όπως ο MySQL Connector/Net για την παροχή διασύνδεση με το Visual Studio και του οδηγού JDBC για Java. Επιπλέον, υπάρχει και διασύνδεση για ODBC (διεπαφή MyODBC) η οποία επιτρέπει επιπλέον γλώσσες προγραμματισμού να υποστηρίξουν τη MySQL με χρήση διασύνδεσης ODBC. Ο διακομιστής της βάσης δεδομένων MySQL και οι επίσημες βιβλιοθήκες υλοποιούνται κατά κανόνα σε ANSI C / ANSI C++.

Η MySQL χρησιμοποιεί το μεγαλύτερο μέρος του πρότυπου SQL. Η υποστήριξή του όμως, δεν είναι πλήρης και αρκετές παραβιάσεις του έχουν σημειωθεί, η σημαντικότερη εκ των οποίων είναι η απουσία υποστήριξης ξένων κλειδιών.

Συνοψίζοντας, τα σημαντικότερα πλεονεκτήματα της MySQL είναι τα εξής:

- Αποτελεί μια λύση ανοιχτού κώδικα, ο οποίος συντηρείται διαρκώς, από το οποίο προκύπτει ένας δωρεάν διακομιστής σχεσιακών βάσεων με πολύ καλή ποιότητα.

- Το τελευταίο διάστημα πίσω από την MySQL είναι η εταιρεία-κολοσσός στο χώρο των βάσεων δεδομένων (Oracle), συνεπώς υπάρχουν μεγαλύτερες εξασφαλίσεις για ποιοτικότερο προϊόν στο μέλλον
- Ο διακομιστής τρέχει σχεδόν σε κάθε λειτουργικό που υπάρχει αυτή τη στιγμή
- Υποστηρίζει τα πρότυπα της ANSI SQL, συνεπώς δεν διαφέρει κατά πολύ από τα υπόλοιπα εμπορικά συστήματα.

### **6.1.3 Γλώσσα προγραμματισμού Java**

Η Java είναι μια γενικής χρήσης αντικειμενοστραφείς γλώσσα προγραμματισμού που έχει σχεδιαστεί ειδικά για να δημιουργεί εφαρμογές με όσο το δυνατόν λιγότερες εξαρτήσεις. Μια από τις βασικές επιλογές κατά το σχεδιασμό της Java ήταν το μοτίβο "write once, run anywhere" (σύγγραψε κώδικα μια φορά, εκτέλεσέ τον παντού) πράγμα που σημαίνει ότι κώδικας που τρέχει σε μια πλατφόρμα δεν χρειάζεται να αναθεωρείται για να τρέξει σε μια άλλη. Οι εφαρμογές Java συνήθως συγκεντρώνονται σε bytecode (αρχεία class) τα οποία έχουν τη δυνατότητα να εκτελεστούν σε οποιαδήποτε συμβατή εικονική μηχανή Java (JVM), ανεξάρτητα από την αρχιτεκτονική του υπολογιστή.

Η Java είναι μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού σε χρήση, ιδιαίτερα για client-server εφαρμογές web. Αναπτύχθηκε αρχικά από τον James Gosling στη Sun Microsystems (η οποία πλέον έχει συγχωνευθεί με την Oracle Corporation) και κυκλοφόρησε το 1995 από τη Sun Microsystems. Η γλώσσα αντλεί μεγάλο μέρος της σύνταξης από C και C++, αν και φαινομενικά στερείται των δυνατών αλλά και επικίνδυνων δεικτών μνήμης (pointer). Παρόλα αυτά, διαθέτει έναν έξυπνο και πανίσχυρο Garbage Collector ο οποίος διασφαλίζει ότι δεν θα καταλαμβάνεται άσκοπα μνήμη από την εφαρμογή.

#### **6.1.4 Appfuse**

Η πλατφόρμα υλοποίησης εφαρμογών AppFuse είναι το δημιούργημα της open source κοινότητας το οποίο μπορεί να χρησιμοποιηθεί για την αρχική δημιουργία επαγγελματικών web εφαρμογών με χρήση γνωστών πλατφόρμων MVC. Ο πιο σημαντικός στόχος της AppFuse είναι να παρέχει μια θεμελιώδη εφαρμογή όπου ο προγραμματιστής απασχολείται με τις ουσιαστικές προδιαγραφές της εφαρμογής, την σχεδίαση του μοντέλου, την εφαρμογή της λογικής και το αισθητικό αποτέλεσμα. Το AppFuse παρέχει άνεση, καθώς ενσωματώνει σχεδόν αυτόματα όλες τις τεχνολογίες αιχμής που διαθέτει η Java.



Η πλατφόρμα υλοποίησης AppFuse είναι εξαιρετικά γενική και προσφέρει τη διασύνδεση με πολλές αρχιτεκτονικές πλατφόρμες MVC, με ενιαίο σχεδόν τρόπο. Στην συγκεκριμένη υλοποίηση, επιλέξαμε την υλοποίηση Spring MVC, (οι άλλες δύο είναι το Struts και το Tapestry) την οποία αναφέρουμε στο επόμενο υποκεφάλαιο.

Τα χαρακτηριστικά που ενσωματώνονται εξ αρχής στην πρότυπη εφαρμογή που παράγει το AppFuse είναι τα ακόλουθα:

- Μονάδα ταυτοποίησης και εξουσιοδότησης χρηστών, με δυνατότητα απομνημόνευσης και ανάκτησης κωδικού: Πρόκειται για την πιο κοινότυπη λειτουργία των περισσότερων συστημάτων που υπάρχουν σήμερα. Η μονάδα αυτή επεκτείνεται με τα εξής:
  - Μονάδα διαχείρισης χρηστών: όπου ο διαχειριστής μπορεί να επεξεργαστεί όλους τους χρήστες, και
  - Εγγραφή νέων χρηστών: όπου δίνεται η δυνατότητα σε κάθε χρήστη να πραγματοποιήσει με εύκολο τρόπο εγγραφή στο σύστημα
- Υποστήριξη SSL: Η δυνατότητα του κρυπτογράφησης των επικοινωνιών με χρήση SSL είναι απαραίτητη σε κάθε σύστημα όπου διακινούνται ευαίσθητες πληροφορίες

- Υποστήριξη E-Mail για την αυτοματοποιημένη επικοινωνία μεταξύ των χρηστών της εφαρμογής, αλλά και της εφαρμογής με τους χρήστες
- URL φιλικά στις μηχανές αναζήτησης
- Εύκολη διαχείριση αισθητικών τμημάτων της εφαρμογής (όψεων – views)

### **6.1.5 Spring MVC**

Το πλαίσιο Spring είναι μια πλατφόρμα ανάπτυξης εφαρμογών που έχει αναπτυχθεί για την γλώσσα προγραμματισμού Java

Το βασικό χαρακτηριστικό του πλαισίου Spring αποτελεί το γεγονός ότι μπορεί να χρησιμοποιηθεί από οποιαδήποτε εφαρμογή Java καθώς και για τη δημιουργία εφαρμογών web με την χρήση της Java EE πλατφόρμας.

Το πλαίσιο Spring διαθέτει την δική του υλοποίηση για την ανάπτυξη εφαρμογών με βάση το MVC πρότυπο, αν και η συγκεκριμένη μορφή δεν είχε προβλεφθεί στον αρχικό σχεδιασμό του. Οι προγραμματιστές του πλαισίου αποφάσισαν να γράψουν

το δικό τους πλαίσιο ως αντίδραση στο δημοφιλές πλαίσιο Struts, καθώς και ελλείψεις σε άλλα διαθέσιμα πλαίσια. Ειδικότερα, αισθάνθηκαν ότι υπήρξε ανεπαρκής διαχωρισμός μεταξύ της παρουσίασης και του ελεγκτή, καθώς και μεταξύ του ελεγκτή και του μοντέλου.

Το πλαίσιο Spring MVC σήμερα γνωρίζει ευρεία αποδοχή, για αυτό και έχει επιλεγεί η υιοθέτησή του σε αυτήν την εργασία.

## **6.2 Πλεονεκτήματα από την υιοθέτηση τεχνολογιών αιχμής**

Οι τεχνολογίες που έχουν αναφερθεί στο προηγούμενο κεφάλαιο αποτελούν τεχνολογίες αιχμής. Τα πλεονεκτήματα που προσδίδουν στην εφαρμογή μας είναι τα ακόλουθα:

- Το σύνολο των εργαλείων και των εφαρμογών που έχουν χρησιμοποιηθεί, ανήκουν στο Ελεύθερο Λογισμικό / Λογισμικό Ανοικτού Κώδικα (ΕΛ/ΛΑΚ). Ως εκ τούτου, το κόστος υλοποίησης ενός τέτοιου συστήματος περιορίζεται πρακτικά στο κόστος εργασίας.
- Χρησιμοποιούνται γνωστές πλατφόρμες για τα τετριμμένα σημεία της πλατφόρμας (όπως σύνδεση της πλατφόρμας με τη βάση δεδομένων, ή το σύστημα αυθεντικοποίησης και πιστοποίησης χρηστών). Η διατήρηση του συστήματος με

ενημερωμένες εκδόσεις το καθιστά εξαιρετικά ασφαλές και μειώνει δραματικά τις πιθανότητες ύπαρξης ανασφαλών κώδικα

- Η υιοθέτηση της αρχιτεκτονικής MVC προσφέρει επεκτασιμότητα στην εφαρμογή. Νέες δυνατότητες μπορούν να προστεθούν χωρίς να επηρεαστεί η υπόλοιπη λειτουργικότητα.
- Η χρήση της αρχιτεκτονικής MVC προσφέρει ανεξαρτησία από την γραφική διεπαφή. Το γραφικό περιβάλλον μπορεί να τροποποιηθεί οποιαδήποτε στιγμή χωρίς να επιδράσει στη λειτουργικότητα της εφαρμογής.

## 6.3 Παρουσίαση της εφαρμογής

Μετά την εκκίνηση της εφαρμογής, κατά την πρώτη προσπάθεια περιήγησης, η εφαρμογή απαγορεύει την πρόσβαση και εμφανίζει την φόρμα για την πιστοποίηση του χρήστη (Εικόνα 6). Η εφαρμογή υποστηρίζει την εγγραφή νέων χρηστών, καθώς και την υπενθύμιση κωδικού.

**Medicine**

### Sign In

admin

.....

Remember Me

Login

Not a member? [Signup for an account.](#)

Forgot your password? Have your [password hint e-mailed to you.](#)

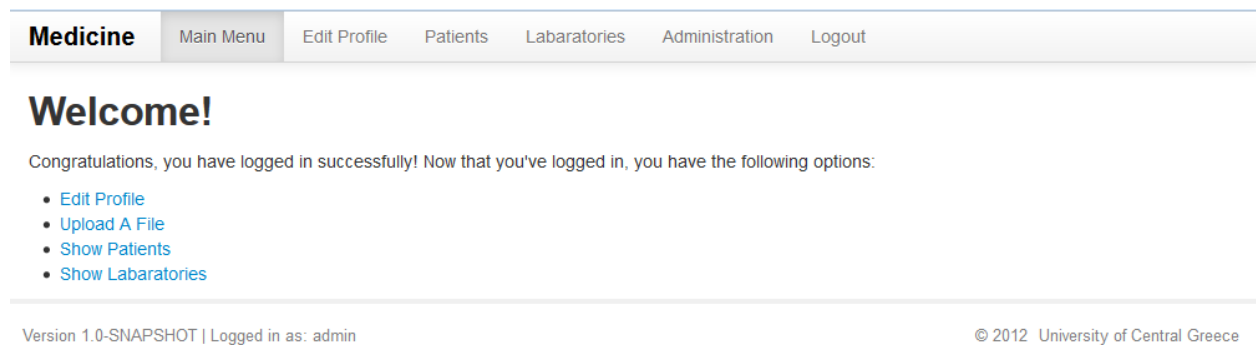
Version 1.0-SNAPSHOT © 2012 University of Central Greece

**Εικόνα 6: Γραφική Διεπαφή – Μονάδα πιστοποίησης χρηστών**

Στη συνέχεια ο χρήστης οδηγείται στο μενού βασικών επιλογών, από όπου μπορεί να επιλέξει την ενέργεια που επιθυμεί:

- Επεξεργασία του προσωπικού προφίλ
- Διαχείριση ασθενών
- Διαχείριση Εργαστηρίων/Ιατρείων
- Διαχείριση της εφαρμογής

Από τις παραπάνω επιλογές σε κάθε χρήστη είναι διαθέσιμες μόνο όσες επιτρέπονται από τον ρόλο του (Εικόνα 7).



**Εικόνα 7: Γραφική Διεπαφή – Κεντρικό μενού**

Στην επεξεργασία του προφίλ, ο χρήστης μπορεί να τροποποιήσει τα στοιχεία του, να αλλάξει το password και τα στοιχεία επικοινωνίας του (Εικόνα 8)

The screenshot shows a web application interface for editing a user profile. The top navigation bar includes 'Medicine', 'Main Menu', 'Edit Profile', 'Patients', 'Laboratories', 'Administration', and 'Logout'. The main content area is titled 'User Profile' and contains the following fields:

- Username \*: admin
- Password \*: [masked]
- Confirm Password \*: [masked]
- Password Hint \*: Not a female kitty.
- First Name \*: Matt
- Last Name \*: Raible
- E-Mail \*: matt@raibledesigns.com
- Phone Number: [empty]
- Website: http://raibledesigns.com

Below the fields is an 'Address' section with 'Current Roles: ROLE\_ADMIN'. At the bottom of the form are two buttons: 'Save' (with a checkmark icon) and 'Cancel' (with an 'X' icon).

Version 1.0-SNAPSHOT | Logged in as: admin © 2012 University of Central Greece

**Εικόνα 8: Επεξεργασία προφίλ χρήστη**

Στο μενού των ασθενών, ο χειριστής μπορεί να παρατηρήσει το σύνολο των καταγεγραμμένων ασθενών (Εικόνα 9), καθώς και να τους επεξεργαστεί Εικόνα 10.

## Persons

[+ Add](#) [✓ Done](#)

One People found.

Id	First Name	Last Name	Actions
1	Matt	Raible	<a href="#">Prescriptions</a>

Export options: [CSV](#) | [Excel](#) | [XML](#) | [PDF](#)

### Εικόνα 9: Λίστα ασθενών

## Person Information

First Name \*

Last Name \*

Birth Day

[✓ Save](#) [🗑 Delete](#) [✕ Cancel](#)

### Εικόνα 10: Προσθήκη / Επεξεργασία ασθενή



Στις εικόνες που ακολουθούν (Εικόνα 11-Εικόνα 14), παρουσιάζεται η μορφή της εφαρμογής για την λίστα των ιατρείων/εργαστηρίων, καθώς και των εξετάσεων ανά ιατρείο /εργαστήριο. Επίσης παρουσιάζονται οι φόρμες εισαγωγής και επεξεργασίας των νέων στοιχείων.

## Laboratories Information

[+ Add](#) [✓ Done](#)

2 Laboratories found, displaying all Laboratories.

ID	Laboratory Name	Laboratory Open Hour	Laboratory Close Hour	Actions
1	Laboratory 1	7	22	<a href="#">Show examinations</a>
2	Laboratory 2	5	23	<a href="#">Show examinations</a>

Export options: [CSV](#) | [Excel](#) | [XML](#) | [PDF](#)

### Εικόνα 11: Κατάλογος ιατρείων/εργαστηρίων

## Laboratory Details

Laboratory Name *	<input type="text" value="Laboratory 2"/>
Laboratory Open Hour *	<input type="text" value="5"/>
Laboratory Close Hour *	<input type="text" value="24"/>

### Εικόνα 12: Προσθήκη/επεξεργασία ιατρείου/εργαστηρίου

## Examination Details

Description	<input type="text" value="Introductory Examination"/>
Minutes Required for Examination	<input type="text" value="15"/>

### Εικόνα 13: Εισαγωγή/Επεξεργασία Εξέτασης

## Examination list

### Labaratory 1

[+ Add](#) [✓ Done](#)

One examinationList.examination found.

ID	Description	Minutes Required for Examination
1	Introductory Examination	15

Export options: [CSV](#) | [Excel](#) | [XML](#) | [PDF](#)

**Εικόνα 14: Κατάλογος διαθέσιμων εξετάσεων ανά εργαστήριο/ιατρείο.**

## 7 Επίλογος

Στην εργασία αυτή παρουσιάσαμε την χρησιμότητα των πληροφοριακών συστημάτων στο χώρο της υγείας. Έπειτα από την παρουσίαση των λειτουργιών που πρέπει να εκτελούν τα πληροφοριακά συστήματα στο χώρο της υγείας, παρουσιάζουμε ορισμένες ερευνητικές προσπάθειες σχετικά με την εφαρμογή τους, τόσο γενικότερα, όσο και με την μορφή των συστημάτων ERP, τα οποία είναι δυνατό να θεωρηθούν ως μια ολοκληρωμένη λύση για πληροφοριακό σύστημα, που μπορεί, με κατάλληλες προσαρμογές να ικανοποιήσει τις ανάγκες των μονάδων υγείας.

Στη συνέχεια, λάβαμε ένα μικρό υποπρόβλημα της συγκεκριμένης εργασίας, που αφορά την κινητικότητα των ασθενών εντός του ίδιου νοσοκομειακού οργανισμού. Προχωρήσαμε στην ανάλυση του προβλήματος, σχεδιάσαμε μια κατάλληλη βάση δεδομένων και την υλοποιήσαμε με χρήση τεχνολογιών αιχμής. Κατά την υλοποίηση έχουμε προνοήσει, ώστε το σύστημα να είναι εξελίξιμο και επεκτάσιμο ώστε να είναι σε θέση να υποστηρίξει πιθανές νέες απαιτήσεις και επεκτάσεις.

## Βιβλιογραφία

- [1] FORTH, «Integrated Health Information System and ICT Services for the 3rd Regional Health System of Attica,» FORTH, 2009. [Ηλεκτρονικό]. Available: [http://www.ics.forth.gr/ceha/index\\_main.php?l=e&c=494](http://www.ics.forth.gr/ceha/index_main.php?l=e&c=494). [Πρόσβαση 2012].
- [2] World Health Organization, Global Observatory for eHealth series - Volume 1, 2010.
- [3] H. Bidgoli, The Internet Encyclopedia, Vol 1, John Wiley & Sons, 2004.
- [4] L. SHAUL και D. TAUBER, «CSFs along ERP life-cycle in SMEs: a field study,» *Industrial Management & Data Systems*,, τόμ. 112, αρ. 3, pp. 360-384, 2012.
- [5] Thomson Course Technology, Thin Enterprise Resource Planning, Boston, 2006.
- [6] R. Gill, «The rise of two-tier ERP,» *Strategic Finance*, τόμ. 93, αρ. 5, pp. 35-40, 2011.
- [7] P. Bingi, M. K. S. Godla και J. K., «CRITICAL ISSUES AFFECTING AN ERP IMPLEMENTATION,» *Information Systems Management*, τόμ. 16, αρ. 3, pp. 7-15, 1999.
- [8] E. Monk και B. Wagner, «Concepts in Enterprise Resource

- Planning,» σε *Course Technology Cengage Learning.*, Boston, 2009.
- [9] M. Selchert, *Enhanced Project Success Through SAP Best Practices: International Benchmarking Study*, SAP PRESS, 2004.
- [10] N. Archer, U. Fevrier-Thomas, C. Lokker, K. A. McKibbin και S. E. Straus, «Personal health records: a scoping review,» *J Am Med Inform Assoc*, τόμ. 18, pp. 515-522, 2011.
- [11] C. EW, «Lessons from the NHS National Programme for IT.,» *Med J Aust.* , τόμ. 186, αρ. 1, pp. 3-4, 2007 .
- [12] C. J. Reynolds και J. C. Wyatt, «Open Source, Open Standards, and Health Care Information Systems,» *J Med Internet Res*, τόμ. 13, αρ. 1, p. e24, 2011 Jan-Mar.
- [13] J. AK, D. CM, C. EG, D. K, R. SR, F. TG, S. A, R. S και B. D, «Use of electronic health records in U.S. hospitals,» *N Engl J Med*, τόμ. 360, αρ. 16, pp. 1628-38, 2009.
- [14] OpenMRS, «OpenMRS: Open Medical Record System,» 2012. [Ηλεκτρονικό]. Available: <http://openmrs.org/>.
- [15] B. Escobar και T. Escobar, «ERP Systems In Hospitals: A Case Study,» *Global Journal of Management and Business Research*, τόμ. 10, αρ. 2, pp. 104-112, April 2010.
- [16] Z. Huq και T. N. Martin, «The recovery of BPR implementation through an ERP approach A hospital case

study,» *Business Process Management*, τόμ. 12, αρ. 5, pp. 576-587, 2006.

[17] M. A. Weisfeld, «The Object-Oriented Thought Process,» Sams Publishing, 2004, p. p. 239.

[18] «MySQL Community Database,» Oracle, 2012. [Ηλεκτρονικό]. Available: <http://www.mysql.com/>.

## 8 Παράρτημα- Source Code

### Περιεχόμενα

Controllers.....	74
ExaminationController .....	74
ExaminationFormController .....	75
LabaratoryController .....	77
LabaratoryFormController .....	78
PerscriptionFromController .....	80
PersonController .....	83
PersonFormController .....	83
PersonPrescriptionController .....	85
PersonPrescriptionFormController.....	86
PrescriptionController .....	89
UserController .....	91
UserFormController .....	92
Models .....	97
ExaminationModel .....	97
LaboratoryModel .....	99
PersonModel .....	101
PrescriptionModel .....	103
RootObjectModel.....	105
Managers.....	106
ExaminationManager .....	106



PrescriptionManager .....	106
ExaminationManagerImpl .....	106
PrescriptionManagerImpl .....	107
Views.....	108
ExaminationForm .....	108
Examinations .....	110
Laboratories .....	111
LaboratoryForm .....	112
MainMenu .....	113
PersonForm .....	114
PersonPrescription .....	116
Persons.....	117
PrescriptionForm .....	118
Prescriptions.....	119
Signup.....	120
UserForm .....	124

## 8.1 Controllers

### *ExaminationController*

```
package gr.edu.webapp.controller;

import java.util.Iterator;
import java.util.Map;
import java.util.Map.Entry;

import javax.servlet.http.HttpServletRequest;

import gr.edu.webapp.model.Examination;
import gr.edu.webapp.model.Laboratory;
import gr.edu.webapp.service.ExaminationManager;

import org.appfuse.service.GenericManager;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

@Controller
@RequestMapping("/examinations*")
public class ExaminationController {
    private ExaminationManager examinationManager;
    private GenericManager<Laboratory, Long> laboratoryManager = null;

    @Autowired
    public void setLaboratoryManager(@Qualifier("laboratoryManager")
GenericManager<Laboratory, Long> laboratoryManager) {
        this.laboratoryManager = laboratoryManager;
    }

    @Autowired
    public void setExaminationManager(@Qualifier("examinationManager")
ExaminationManager examinationManager) {
        this.examinationManager = examinationManager;
    }

    @RequestMapping(value="/{labid}",method = RequestMethod.GET)
    public ModelAndView handleRequest(HttpServletRequest
request,@PathVariable long labid)
    throws Exception {
        Laboratory lab = laboratoryManager.get(labid);
        if(lab == null)
        {
            throw new Exception("Lab not found");
        }
    }
}
```

```

    }
    ModelAndView mv = new ModelAndView("examinations");
mv.addObject(examinationManager.findByLaboratory(lab)).addObject(lab);
    return mv;
}
}

```

### *ExaminationFormController*

```

package gr.edu.webapp.controller;

import gr.edu.webapp.model.*;
import gr.edu.webapp.service.ExaminationManager;

import java.util.Locale;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.commons.lang.StringUtils;
import org.appfuse.service.GenericManager;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.validation.ObjectError;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.servlet.ModelAndView;

@Controller
@RequestMapping("/examinationform*")
public class ExaminationFormController extends BaseFormController {

    private ExaminationManager examinationManager = null;
    private GenericManager<Laboratory, Long> laboratoryManager = null;

    @Autowired
    public void setLaboratoryManager(@Qualifier("laboratoryManager")
GenericManager<Laboratory, Long> laboratoryManager) {
        this.laboratoryManager = laboratoryManager;
    }

    @Autowired
    public void setLaboratoryManager(@Qualifier("examinationManager")
ExaminationManager examinationManager) {
        this.examinationManager = examinationManager;
    }
}

```

```

}

public ExaminationFormController() {
}

@RequestMapping(value = "{labid}", method = RequestMethod.GET)
protected String showForm(HttpServletRequest request,
@PathVariable long labid,
@ModelAttribute("examination") Examination ex)
throws Exception {
    setCancelView("redirect:/examinations/" + labid);
    setSuccessView("redirect:/examinations/" + labid);
    String id = request.getParameter("id");

    if (!StringUtils.isBlank(id)) {
        ex = examinationManager.get(new Long(id));
    } else {
        Laboratory lab = laboratoryManager.get(new Long(labid));

        if (lab == null) {
            throw new Exception("Lab not found");
        }

        ex = new Examination();
        ex.setLab(lab);
    }
    return "examinationForm";
}

@RequestMapping(value = "{labid}/examinationform", method =
RequestMethod.POST)
public String onSubmit(Examination ex, BindingResult errors,
HttpServletRequest request,
    HttpServletResponse response, @PathVariable long labid)
throws Exception {
    setCancelView("redirect:/examinations/" + labid);
    setSuccessView("redirect:/examinations/" + labid);
    if (request.getParameter("cancel") != null) {
        return getCancelView();
    }
    if (ex.getLab() == null) {
        ex.setLab(laboratoryManager.get(new Long(labid)));
    }

    if (validator != null) { // validator is null during testing
        validator.validate(ex, errors);
        for (ObjectError v : errors.getAllErrors()) {
            System.out.println(v.getDefaultMessage());
        }
        if (errors.hasErrors() && request.getParameter("delete")
== null) { // don't validate when deleting
            return "examinationform";
        }
    }
}

```

```

    }

    log.debug("entering 'onSubmit' method...");

    boolean isNew = (ex.getId() == null);
    String success = getSuccessView();
    Locale locale = request.getLocale();

    if (request.getParameter("delete") != null) {
        examinationManager.remove(ex.getId());
        saveMessage(request, getText("examination.deleted",
locale));
    } else {
        examinationManager.save(ex);
        String key = (isNew) ? "examination.added" :
"examination.updated";
        saveMessage(request, getText(key, locale));

        if (!isNew) {
            success = "redirect:examinationform?/" + labid + "id="
+ ex.getId();
        }
    }

    return success;
}
}

```

### *LabaratoryController*

```

package gr.edu.webapp.controller;

import gr.edu.webapp.model.*;

import org.appfuse.service.GenericManager;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

@Controller
@RequestMapping("/labaratories*")
public class LabaratoryController {
    private GenericManager<Labaratory, Long> labaratoryManager;

    @Autowired
    public void setPersonManager(@Qualifier("labaratoryManager")
GenericManager<Labaratory, Long> labaratoryManager) {
        this.labaratoryManager = labaratoryManager;
    }
}

```

```

    @RequestMapping(method = RequestMethod.GET)
    public ModelAndView handleRequest()
    throws Exception {
        return
ModelAndView().addObject(labaratoryManager.getAll());
    }
}

```

new

## LabaratoryFormController

```

package gr.edu.webapp.controller;

import gr.edu.webapp.model.*;

import java.util.Locale;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.commons.lang.StringUtils;
import org.appfuse.service.GenericManager;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Controller;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

@Controller
@RequestMapping("/labaratoryform*")
public class LabaratoryFormController extends BaseFormController {
    private GenericManager<Laboratory, Long> labaratoryManager = null;

    @Autowired
    public void setLabaratoryManager(@Qualifier("labaratoryManager")
GenericManager<Laboratory, Long> labaratoryManager) {
        this.labaratoryManager = labaratoryManager;
    }

    public LabaratoryFormController() {
        setCancelView("redirect:labaratories");
        setSuccessView("redirect:labaratories");
    }

    @ModelAttribute
    @RequestMapping(method = RequestMethod.GET)
    protected Laboratory showForm(HttpServletRequest request)
    throws Exception {
        String id = request.getParameter("id");
}

```

```

        if (!StringUtils.isBlank(id)) {
            return laboratoryManager.get(new Long(id));
        }

        return new Laboratory();
    }

    @RequestMapping(method = RequestMethod.POST)
    public String onSubmit(Laboratory lab, BindingResult errors,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        if (request.getParameter("cancel") != null) {
            return getCancelView();
        }

        if (validator != null) { // validator is null during testing
            validator.validate(lab, errors);

            if (errors.hasErrors() && request.getParameter("delete")
                == null) { // don't validate when deleting
                return "personform";
            }
        }

        log.debug("entering 'onSubmit' method...");

        boolean isNew = (lab.getId() == null);
        String success = getSuccessView();
        Locale locale = request.getLocale();

        if (request.getParameter("delete") != null) {
            laboratoryManager.remove(lab.getId());
            saveMessage(request,
                locale,
                getText("laboratory.deleted",
                    locale));
        } else {
            laboratoryManager.save(lab);
            String key = (isNew) ? "laboratory.added" :
                "laboratory.updated";
            saveMessage(request, getText(key, locale));

            if (!isNew) {
                success = "redirect:laboratoryform?id=" + lab.getId();
            }
        }

        return success;
    }
}

```

## *PerscriptionFromController*

```
package gr.edu.webapp.controller;

import gr.edu.webapp.model.*;
import gr.edu.webapp.service.ExaminationManager;
import gr.edu.webapp.service.PrescriptionManager;
import java.util.Date;

import java.util.Locale;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.commons.lang.StringUtils;
import org.appfuse.service.GenericManager;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.validation.ObjectError;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.servlet.ModelAndView;

@Controller
@RequestMapping("/prescriptionform*")
public class PrescriptionFormController extends BaseFormController {

    private PrescriptionManager prescriptionManager = null;
    private ExaminationManager examinationManager = null;
    private GenericManager<Laboratory, Long> laboratoryManager = null;
    private GenericManager<Person, Long> personManager = null;

    @Autowired
    public void setPersonManager(@Qualifier("personManager")
GenericManager<Person, Long> personManager) {
        this.personManager = personManager;
    }

    @Autowired
    public void setLaboratoryManager(@Qualifier("laboratoryManager")
GenericManager<Laboratory, Long> laboratoryManager) {
        this.laboratoryManager = laboratoryManager;
    }

    @Autowired
    public void setExaminationManager(@Qualifier("examinationManager")
ExaminationManager examinationManager) {
```



```

        this.examinationManager = examinationManager;
    }

    @Autowired
    public void setLaboratoryManager(@Qualifier("prescriptionManager")
PrescriptionManager prescriptionManager) {
        this.prescriptionManager = prescriptionManager;
    }

    public PrescriptionFormController() {
    }

    @RequestMapping(value = "/{patientid}", method =
RequestMethod.GET)
    protected String showForm(HttpServletRequest request,
@PathVariable long patientid,
@ModelAttribute("prescription") Prescription pr, Model
model)
        throws Exception {
        setCancelView("redirect:/prescriptions/" + patientid);
        setSuccessView("redirect:/prescriptions/" + patientid);
        String id = request.getParameter("id");
        model.addAttribute("examinationList",
examinationManager.getAll());
        if (!StringUtils.isBlank(id)) {
            pr = prescriptionManager.get(new Long(id));
        } else {
            Person per = personManager.get(new Long(patientid));

            if (per == null) {
                throw new Exception("Patient not found");
            }

            pr = new Prescription();
            pr.setPatient(per);
        }
        return "prescriptionForm";
    }

    @RequestMapping(value = "/{patientid}/prescriptionform", method =
RequestMethod.POST)
    public String onSubmit(Prescription ex, BindingResult errors,
HttpServletRequest request,
HttpServletResponse response, @PathVariable long
patientid)
        throws Exception {
        setCancelView("redirect:/prescriptions/" + patientid);
        setSuccessView("redirect:/prescriptions/" + patientid);
        if (request.getParameter("cancel") != null) {
            return getCancelView();
        }
        if (ex.getPatient() == null) {
            ex.setPatient(personManager.get(new Long(patientid)));
        }
    }

```

```

    }
    if(ex.getExamination()==null)
    {
        try {
            Long lg =
Long.parseLong(request.getParameter("examination_id"));
            Examination exm = examinationManager.get(lg);
            ex.setExamination(exm);

        } catch (Exception e) {
            log.info("error",e);
        }
    }
    if (validator != null) { // validator is null during testing
        validator.validate(ex, errors);
        for (ObjectError v : errors.getAllErrors()) {
            System.out.println(v.getDefaultMessage());
        }
        if (errors.hasErrors() && request.getParameter("delete")
== null) { // don't validate when deleting
            return "prescriptionForm";
        }
    }

    log.debug("entering 'onSubmit' method...");
    Examination exam = ex.getExamination();
    exam = examinationManager.get(exam.getId());
    Date dt = exam.getNextAvailableRandevious();
    ex.setRandevious(dt);

    boolean isNew = (ex.getId() == null);
    String success = getSuccessView();
    Locale locale = request.getLocale();

    if (request.getParameter("delete") != null) {
        prescriptionManager.remove(ex.getId());
        saveMessage(request, getText("prescription.deleted",
locale));
    } else {
        prescriptionManager.save(ex);
        String key = (isNew) ? "prescription.added" :
"prescription.updated";
        saveMessage(request, getText(key, locale));

        if (!isNew) {
            success = "redirect:prescriptionform?/" + patientid +
"id=" + ex.getId();
        }
    }
    return success;
}
}

```

## *PersonController*

```
package gr.edu.webapp.controller;

import gr.edu.webapp.model.*;

import org.appfuse.service.GenericManager;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.*;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.servlet.ModelAndView;

@Controller
@RequestMapping("/persons*")
public class PersonController {
    private GenericManager<Person, Long> personManager;

    @Autowired
    public void setPersonManager(@Qualifier("personManager")
GenericManager<Person, Long> personManager) {
        this.personManager = personManager;
    }

    @RequestMapping(method = RequestMethod.GET)
    public ModelAndView handleRequest()
throws Exception {
        return new ModelAndView().addObject(personManager.getAll());
    }
}
```

## *PersonFormController*

```
package gr.edu.webapp.controller;

import java.util.Locale;

import javax.servlet.http.*;

import gr.edu.webapp.model.Person;

import org.appfuse.service.GenericManager;
import org.springframework.beans.factory.annotation.*;
import org.springframework.stereotype.*;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.*;
import org.apache.commons.lang.StringUtils;
```

```

@Controller
@RequestMapping("/personform*")
public class PersonFormController extends BaseFormController {
    private GenericManager<Person, Long> personManager = null;

    @Autowired
    public void setPersonManager(@Qualifier("personManager")
GenericManager<Person, Long> personManager) {
        this.personManager = personManager;
    }

    public PersonFormController() {
        setCancelView("redirect:persons");
        setSuccessView("redirect:persons");
    }

    @ModelAttribute
    @RequestMapping(method = RequestMethod.GET)
    protected Person showForm(HttpServletRequest request)
throws Exception {
        String id = request.getParameter("id");

        if (!StringUtils.isBlank(id)) {
            return personManager.get(new Long(id));
        }

        return new Person();
    }

    @RequestMapping(method = RequestMethod.POST)
    public String onSubmit(Person person, BindingResult errors,
HttpServletRequest request,
        HttpServletResponse response)
throws Exception {
        if (request.getParameter("cancel") != null) {
            return getCancelView();
        }

        if (validator != null) { // validator is null during testing
            validator.validate(person, errors);

            if (errors.hasErrors() && request.getParameter("delete")
== null) { // don't validate when deleting
                return "personform";
            }
        }

        log.debug("entering 'onSubmit' method...");

        boolean isNew = (person.getId() == null);
        String success = getSuccessView();
        Locale locale = request.getLocale();

```

```

        if (request.getParameter("delete") != null) {
            personManager.remove(person.getId());
            saveMessage(request, getText("person.deleted", locale));
        } else {
            personManager.save(person);
            String key = (isNew) ? "person.added" : "person.updated";
            saveMessage(request, getText(key, locale));

            if (!isNew) {
                success = "redirect:personform?id=" + person.getId();
            }
        }

        return success;
    }
}

```

### *PersonPrescriptionController*

```

package gr.edu.webapp.controller;

import java.util.ArrayList;
import java.util.List;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import gr.edu.webapp.model.Examination;
import gr.edu.webapp.model.Laboratory;
import gr.edu.webapp.model.Person;
import gr.edu.webapp.model.Prescription;
import gr.edu.webapp.service.ExaminationManager;
import gr.edu.webapp.service.PrescriptionManager;

import org.springframework.service.GenericManager;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

@Controller
@RequestMapping("/personprescription*")
public class PersonPrescriptionController {

    private ExaminationManager examinationManager;
    private PrescriptionManager prescriptionManager;
    private GenericManager<Person, Long> personManager = null;

```

```

    @Autowired
    public void setPrescriptionManager(@Qualifier("prescriptionManager")
    PrescriptionManager prescriptionManager) {
        this.prescriptionManager = prescriptionManager;
    }

    @Autowired
    public void setPersonManager(@Qualifier("personManager")
    GenericManager<Person, Long> personManager) {
        this.personManager = personManager;
    }

    @Autowired
    public void setExaminationManager(@Qualifier("examinationManager")
    ExaminationManager examinationManager) {
        this.examinationManager = examinationManager;
    }

    @RequestMapping(method = RequestMethod.POST)
    public String onSubmit(Prescription prescription, BindingResult
    errors, HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        if(prescription.getExamination()!=null)
        {
            throw new Exception("got examination");
        }
        throw new Exception("got nothing");
    }
}

```

### *PersonPrescriptionFormController*

```

package gr.edu.webapp.controller;

import gr.edu.webapp.model.Examination;
import java.util.Locale;

import javax.servlet.http.*;

import gr.edu.webapp.model.Person;
import gr.edu.webapp.model.Prescription;
import gr.edu.webapp.service.ExaminationManager;
import gr.edu.webapp.service.PrescriptionManager;
import java.util.List;

import org.appfuse.service.GenericManager;
import org.springframework.beans.factory.annotation.*;
import org.springframework.stereotype.*;

```

```

import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.*;
import org.apache.commons.lang.StringUtils;
import org.springframework.ui.Model;

@Controller
@RequestMapping("/personprescriptionform*")
public class PersonPrescriptionFormController extends
BaseFormController {

    private ExaminationManager examinationManager;
    private PrescriptionManager prescriptionManager;
    private GenericManager<Person, Long> personManager = null;

    @Autowired
    public void
setPrescriptionManager(@Qualifier("prescriptionManager")
PrescriptionManager prescriptionManager) {
    this.prescriptionManager = prescriptionManager;
}

    @Autowired
    public void
setPersonManager(@Qualifier("personManager")
GenericManager<Person, Long> personManager) {
    this.personManager = personManager;
}

    @Autowired
    public void
setExaminationManager(@Qualifier("examinationManager")
ExaminationManager examinationManager) {
    this.examinationManager = examinationManager;
}

    public PersonPrescriptionFormController() {
        setCancelView("redirect:prescriptions");
        setSuccessView("redirect:prescriptions");
    }

    @RequestMapping(value =("/{personid}", method =
RequestMethod.GET)
    public String showForm(HttpServletRequest request,@PathVariable
long personid, Model model) throws Exception
    {
        Person patient;
        List<Examination> exlist;
        patient = personManager.get(personid);
        if(patient == null)
            throw new Exception("Patient not found");
        exlist = examinationManager.getAll();
        if(exlist.size()==0)
            throw new Exception("No Examination found");
        model.addAttribute("patient",patient);
        model.addAttribute("person",patient);
        model.addAttribute("examinationList",exlist);
    }
}

```

```

        Prescription p = new Prescription();
        p.setPatient(patient);
        model.addAttribute("prescription", p);

        return "personPrescription";
    }

    @RequestMapping(value = "{pid}/examinationform", method =
RequestMethod.POST)
    public String onSubmit(Prescription prescr, BindingResult errors,
HttpServletRequest request,
        @PathVariable long pid,
        HttpServletResponse response)
        throws Exception {
        log.info("onSubmit entered");
        setCancelView("redirect:/persons/" + pid);
        setSuccessView("redirect:/persons/" + pid);

        if (request.getParameter("cancel") != null) {
            return getCancelView();
        }

        if (validator != null) { // validator is null during testing
            validator.validate(prescr, errors);

            if (errors.hasErrors() && request.getParameter("delete")
== null) { // don't validate when deleting
                return "personform";
            }
        }
        log.debug("entering 'onSubmit' method...");

        boolean isNew = (prescr.getId() == null);
        String success = getSuccessView();
        Locale locale = request.getLocale();

        if (request.getParameter("delete") != null) {
            prescriptionManager.remove(prescr.getId());
            saveMessage(request, getText("prescription.deleted",
locale));
        } else {
            prescriptionManager.save(prescr);
            String key = (isNew) ? "prescription.added" :
"prescription.updated";
            saveMessage(request, getText(key, locale));

            if (!isNew) {
                success = "redirect:prescriptionform?id=" +
prescr.getId();
            }
        }
        return success;
    }

```



```
}  
}
```

## *PrescriptionController*

```
package gr.edu.webapp.controller;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import gr.edu.webapp.model.Examination;  
import gr.edu.webapp.model.Labaratory;  
import gr.edu.webapp.model.Person;  
import gr.edu.webapp.model.Prescription;  
import gr.edu.webapp.service.ExaminationManager;  
import gr.edu.webapp.service.PrescriptionManager;  
  
import javax.servlet.http.HttpServletRequest;  
  
import org.appfuse.service.GenericManager;  
import org.hibernate.Hibernate;  
import org.hibernate.collection.internal.PersistentBag;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.beans.factory.annotation.Qualifier;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.servlet.ModelAndView;  
  
@Controller  
@RequestMapping("/prescriptions*")  
public class PrescriptionController {  
  
    private ExaminationManager examinationManager;  
    private PrescriptionManager prescriptionManager;  
    private GenericManager<Labaratory, Long> labaratoryManager = null;  
    private GenericManager<Person, Long> personManager = null;  
  
    @Autowired  
    public void  
    setPrescriptionManager(@Qualifier("prescriptionManager")  
    PrescriptionManager prescriptionManager) {  
        this.prescriptionManager = prescriptionManager;  
    }  
  
    @Autowired  
    public void  
    setPersonManager(@Qualifier("personManager")  
    GenericManager<Person, Long> personManager) {  
        this.personManager = personManager;  
    }  
}
```

```

    @Autowired
    public void setLaboratoryManager(@Qualifier("laboratoryManager")
GenericManager<Laboratory, Long> laboratoryManager) {
        this.laboratoryManager = laboratoryManager;
    }

    @Autowired
    public void setExaminationManager(@Qualifier("examinationManager")
ExaminationManager examinationManager) {
        this.examinationManager = examinationManager;
    }

    @RequestMapping(value="laboratory/{labid}",method =
RequestMethod.GET)
    public ModelAndView handleRequestLaboratory(HttpServletRequest request,
@PathVariable long labid)
    throws Exception {
        String type = "laboratory";
        Laboratory lab = laboratoryManager.get(labid);
        if(lab == null)
        {
            throw new Exception("Lab not found");
        }
        ModelAndView mv = new ModelAndView("prescriptions");

mv.addObject(getByLaboratory(lab)).addObject(lab).addObject("type",
type);
        return mv;
    }

    @RequestMapping(value="patient/{patientid}",method =
RequestMethod.GET)
    public ModelAndView handleRequestPatient(HttpServletRequest request,
@PathVariable long patientid)
    throws Exception {
        Person lab = personManager.get(patientid);
        String type = "person";
        if(lab == null)
        {
            throw new Exception("Lab not found");
        }
        ModelAndView mv = new ModelAndView("prescriptions");
        List<Prescription> prescriptions =
prescriptionManager.findByPatient(lab);
        mv.addObject(prescriptions).addObject(lab).addObject("type",
type);

        return mv;
    }

    private List<Prescription> getByLaboratory(Laboratory lab)
    {

```

```

        List<Prescription> listToReturn = new
ArrayList<Prescription>();
        List<Examination> exs =
examinationManager.findByLaboratory(lab);
        for(Examination x: exs)
        {
            List<Prescription> ps =
prescriptionManager.findByExamination(x);
            listToReturn.addAll(ps);
        }
        return listToReturn;
    }
}

```

## UserController

```

package gr.edu.webapp.controller;

import org.appfuse.Constants;
import org.appfuse.dao.SearchException;
import org.appfuse.service.UserManager;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ExtendedModelMap;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;

/**
 * Simple class to retrieve a list of users from the database.
 * <p/>
 * <p>
 * <a href="UserController.java.html"><i>View Source</i></a>
 * </p>
 *
 * @author <a href="mailto:matt@raibledesigns.com">Matt Raible</a>
 */
@Controller
@RequestMapping("/admin/users*")
public class UserController {
    private UserManager userManager = null;

    @Autowired
    public void setUserManager(UserManager userManager) {
        this.userManager = userManager;
    }

    @RequestMapping(method = RequestMethod.GET)

```

```

    public ModelAndView handleRequest(@RequestParam(required = false,
value = "q") String query) throws Exception {
        Model model = new ExtendedModelMap();
        try {
            model.addAttribute(Constants.USER_LIST,
userManager.search(query));
        } catch (SearchException se) {
            model.addAttribute("searchError", se.getMessage());
            model.addAttribute(userManager.getUsers());
        }
        return new ModelAndView("admin/userList", model.asMap());
    }
}

```

### *UserFormController*

```

package gr.edu.webapp.controller;

import org.apache.commons.lang.StringUtils;
import org.appfuse.Constants;
import org.appfuse.model.Role;
import org.appfuse.model.User;
import org.appfuse.service.RoleManager;
import org.appfuse.service.UserExistsException;
import org.appfuse.service.UserManager;
import gr.edu.webapp.util.RequestUtil;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.mail.MailException;
import org.springframework.security.access.AccessDeniedException;
import org.springframework.security.authentication.AuthenticationTrustResolve
r;
import org.springframework.security.authentication.AuthenticationTrustResolve
rImpl;
import org.springframework.security.core.Authentication;
import org.springframework.security.core.context.SecurityContext;
import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.stereotype.Controller;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.util.Locale;

/**

```

```

* Implementation of <strong>SimpleFormController</strong> that
interacts with
* the {@link UserManager} to retrieve/persist values to the database.
*
* <p><a href="UserFormController.java.html"><i>View Source</i></a>
*
* @author <a href="mailto:matt@raibledesigns.com">Matt Raible</a>
*/
@Controller
@RequestMapping("/userform*")
public class UserFormController extends BaseFormController {
    private RoleManager roleManager;

    @Autowired
    public void setRoleManager(RoleManager roleManager) {
        this.roleManager = roleManager;
    }

    public UserFormController() {
        setCancelView("redirect:/mainMenu");
        setSuccessView("redirect:/admin/users");
    }

    @RequestMapping(method = RequestMethod.POST)
    public String onSubmit(User user, BindingResult errors,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        if (request.getParameter("cancel") != null) {
            if (!StringUtils.equals(request.getParameter("from"),
"list")) {
                return getCancelView();
            } else {
                return getSuccessView();
            }
        }

        if (validator != null) { // validator is null during testing
            validator.validate(user, errors);

            if (errors.hasErrors() && request.getParameter("delete")
== null) { // don't validate when deleting
                return "userform";
            }
        }

        log.debug("entering 'onSubmit' method...");

        Locale locale = request.getLocale();

        if (request.getParameter("delete") != null) {
            getUserManager().removeUser(user.getId().toString());
        }
    }
}

```

```

        saveMessage(request,                getText("user.deleted",
user.getFullName(), locale));

        return getSuccessView();
    } else {

        // only attempt to change roles if user is admin for other
users,
        // showForm() method will handle populating
        if (request.isUserInRole(Constants.ADMIN_ROLE)) {
            String[]          userRoles          =
request.getParameterValues("userRoles");

            if (userRoles != null) {
                user.getRoles().clear();
                for (String roleName : userRoles) {
                    user.addRole(roleManager.getRole(roleName));
                }
            }
        } else {
            // if user is not an admin then load roles from the
database
            // (or any other user properties that should not be
editable
            // by users without admin role)
            User cleanUser = getUserManager().getUserByUsername(
                request.getRemoteUser());
            user.setRoles(cleanUser.getRoles());
        }

        Integer originalVersion = user.getVersion();

        try {
            getUserManager().saveUser(user);
        } catch (AccessDeniedException ade) {
            // thrown by UserSecurityAdvice configured in
aop:advisor userManagerSecurity
            log.warn(ade.getMessage());
            response.sendError(HttpServletResponse.SC_FORBIDDEN);
            return null;
        } catch (UserExistsException e) {
            errors.rejectValue("username", "errors.existing.user",
                new Object[]{user.getUsername(),
user.getEmail()}, "duplicate user");

            // redisplay the unencrypted passwords
            user.setPassword(user.getConfirmPassword());
            // reset the version # to what was passed in
            user.setVersion(originalVersion);

            return "userform";
        }
    }
}

```

```

        if (!StringUtils.equals(request.getParameter("from"),
"list")) {
            saveMessage(request,                getText("user.saved",
user.getFullName(), locale));

            // return to main Menu
            return getCancelView();
        } else {
            if
(StringUtils.isBlank(request.getParameter("version"))) {
                saveMessage(request,                getText("user.added",
user.getFullName(), locale));

                // Send an account information e-mail
                message.setSubject(getText("signup.email.subject",
locale));

                try {
                    sendUserMessage(user,
getText("newuser.email.message", user.getFullName(), locale),
RequestUtil.getAppURL(request));
                } catch (MailException me) {
                    saveError(request,
me.getCause().getLocalizedMessage());
                }

                return getSuccessView();
            } else {
                saveMessage(request,
getText("user.updated.byAdmin", user.getFullName(), locale));
            }
        }

        return "userform";
    }

    @ModelAttribute
    @RequestMapping(method = RequestMethod.GET)
    protected User showForm(HttpServletRequest request,
    HttpServletResponse response)
        throws Exception {
        // If not an administrator, make sure user is not trying to
add or edit another user
        if (!request.isUserInRole(Constants.ADMIN_ROLE) &&
!isFormSubmission(request)) {
            if (isAdd(request) || request.getParameter("id") != null)
{
                response.sendError(HttpServletResponse.SC_FORBIDDEN);
                log.warn("User '" + request.getRemoteUser() + "' is
trying to edit user with id '" +
                    request.getParameter("id") + "'");
            }
        }
    }

```

```

        throw new AccessDeniedException("You do not have
permission to modify other users.");
    }
}

if (!isFormSubmission(request)) {
    String userId = request.getParameter("id");

    // if user logged in with remember me, display a warning
that they can't change passwords
    log.debug("checking for remember me login...");

    AuthenticationTrustResolver resolver = new
AuthenticationTrustResolverImpl();
    SecurityContext ctx = SecurityContextHolder.getContext();

    if (ctx.getAuthentication() != null) {
        Authentication auth = ctx.getAuthentication();

        if (resolver.isRememberMe(auth)) {
            request.getSession().setAttribute("cookieLogin",
"true");

            // add warning message
            saveMessage(request,
getText("userProfile.cookieLogin", request.getLocale()));
        }
    }

    User user;
    if (userId == null && !isAdd(request)) {
        user =
getUserManager().getUserByUsername(request.getRemoteUser());
    } else if (!StringUtils.isBlank(userId) &&
!"".equals(request.getParameter("version"))) {
        user = getUserManager().getUser(userId);
    } else {
        user = new User();
        user.addRole(new Role(Constants.USER_ROLE));
    }

    user.setConfirmPassword(user.getPassword());

    return user;
} else {
    // populate user object from database, so all fields don't
need to be hidden fields in form
    return
getUserManager().getUser(request.getParameter("id"));
}

private boolean isFormSubmission(HttpServletRequest request) {

```



```

        return request.getMethod().equalsIgnoreCase("post");
    }

    protected boolean isAdd(HttpServletRequest request) {
        String method = request.getParameter("method");
        return (method != null && method.equalsIgnoreCase("add"));
    }
}

```

## 8.2 Models

### *ExaminationModel*

```

package gr.edu.webapp.model;

import java.util.ArrayList;
import java.util.Collection;
import java.util.Collections;
import java.util.Date;
import java.util.List;

import javax.persistence.*;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;

import org.appfuse.model.BaseObject;
import org.hibernate.Hibernate;
import org.hibernate.annotations.LazyCollection;
import org.hibernate.annotations.LazyCollectionOption;

@Entity
public class Examination extends BaseObject {

    private String description;
    protected Long id;
    protected Laboratory lab;
    protected int requiredMinutes;
    protected List<Prescription> prescriptions;

    @LazyCollection(LazyCollectionOption.FALSE)
    @OneToMany(mappedBy = "examination")
    public List<Prescription> getPrescriptions() {
        return prescriptions;
    }

    public void setPrescriptions(List<Prescription> prescriptions) {
        this.prescriptions = prescriptions;
    }

    public int getRequiredMinutes() {
        return requiredMinutes;
    }
}

```

```

public void setRequiredMinutes(int requiredMinutes) {
    this.requiredMinutes = requiredMinutes;
}

@ManyToOne(fetch = FetchType.EAGER)
@JoinColumn(name = "LAB_ID", nullable = false)
public Laboratory getLab() {
    return lab;
}

public void setLab(Laboratory lab) {
    this.lab = lab;
}

public void setId(Long id) {
    this.id = id;
}

public String getDescription() {
    return description;
}

public void setDescription(String description) {
    this.description = description;
}

@Transient
public Date getNextAvailableRandevious() {
    Log logger = LogFactory.getLog(getClass());
    Hibernate.initialize(this);
    Date toReturn = new Date(), now = new Date();
    ArrayList<Date> list = new ArrayList<Date>();
    ArrayList<Prescription> lst = new ArrayList<Prescription>(getPrescriptions());
    if (this.getPrescriptions() != null) {
        int r = this.getPrescriptions().size();
        for (Prescription pr : this.getPrescriptions()) {
            logger.info("Randevious:
"+pr.getRandevious().getTime());
            logger.info("Randevious-
end:"+pr.getRandevious().getTime()+60000*pr.getExamination().getRequire
dMinutes());
            if
(pr.getRandevious().getTime()+60000*pr.getExamination().getRequiredMinu
tes() >toReturn.getTime()) {
                list.add(pr.getRandevious());
            }
        }
    }
    if (list.size() > 0) {
        Collections.sort(list);
    }
}

```

```

        toReturn = list.get(list.size() - 1);
        toReturn.setTime(requiredMinutes * 60000 +
toReturn.getTime());
    } else {
        if (toReturn.getHours() < getLab().getStartHour() ||
toReturn.getHours() >= getLab().getStopHour()) {
            toReturn.setHours(getLab().getStartHour());
            toReturn.setMinutes(0);
            toReturn.setSeconds(0);
        }
        if (toReturn.before(now)) {
            toReturn.setDate(toReturn.getDate() + 1);
        }
    }
    return toReturn;
}

@Id
@GeneratedValue(strategy = GenerationType.AUTO)
public Long getId() {
    return id;
}

public boolean equals(Object a) {
    if (a == null) {
        return false;
    }
    if (!a.getClass().equals(getClass())) {
        return false;
    }
    Examination obj = (Examination) a;
    if (!description.equals(obj.description)) {
        return false;
    }
    return this.lab.equals(obj.lab);
}

public int hashCode() {
    return (int) (lab.getId() % Integer.MAX_VALUE) + 31
        * description.hashCode();
}

public String toString() {
    return description;
}
}

```

### LaboratoryModel

```
package gr.edu.webapp.model;
```

```

import java.util.List;

import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToOne;

import org.appfuse.model.BaseObject;
import org.hibernate.annotations.LazyCollection;
import org.hibernate.annotations.LazyCollectionOption;

@Entity
public class Labaratory extends BaseObject {

    private String description;
    private int startHour;
    private int stopHour;
    protected Long id;
    protected List<Examination> examinations;

    public void setExaminations(List<Examination> examinations) {
        this.examinations = examinations;
    }

    @LazyCollection(LazyCollectionOption.FALSE)
    @OneToMany(mappedBy = "lab")
    public List<Examination> getExaminations() {
        return this.examinations;
    }

    public void setId(Long id) {
        this.id = id;
    }

    @Id @GeneratedValue(strategy = GenerationType.AUTO)
    public Long getId() {
        return id;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public int getStartHour() {

```

```

        return startHour;
    }

    public void setStartHour(int startHour) {
        this.startHour = startHour;
    }

    public int getStopHour() {
        return stopHour;
    }

    public void setStopHour(int stopHour) {
        this.stopHour = stopHour;
    }

    public boolean equals(Object a) {
        if(a == null)
            return false;
        if(!a.getClass().equals(getClass()))
            return false;
        Laboratory obj = (Laboratory)a;
        if(!description.equals(obj.description))
            return false;
        return obj.startHour == startHour && obj.stopHour ==
stopHour;
    }

    public int hashCode() {
        return startHour + 125*stopHour + 31*description.hashCode();
    }

    public String toString() {
        return description;
    }
}

```

### *PersonModel*

```

package gr.edu.webapp.model;

import java.util.List;

import javax.persistence.*;

import org.appfuse.model.BaseObject;
import org.hibernate.annotations.LazyCollection;
import org.hibernate.annotations.LazyCollectionOption;

```

```

@Entity
public class Person extends BaseObject {

    protected List<Prescription> prescriptions;

    @LazyCollection(LazyCollectionOption.FALSE)
    @OneToMany(mappedBy="patient")
    public List<Prescription> getPrescriptions() {
        return prescriptions;
    }

    public void setPrescriptions(List<Prescription> prescriptions) {
        this.prescriptions = prescriptions;
    }

    protected java.util.Date birthDay;

    public java.util.Date getBirthDay() {
        return birthDay;
    }

    public void setBirthDay(java.util.Date birthDay) {
        this.birthDay = birthDay;
    }

    protected Long id;

    public void setId(Long id) {
        this.id = id;
    }

    public boolean bequels(Object a) {
        if(a == null)
            return false;
        if(!a.getClass().equals(getClass()))
            return false;
        return true;
    }
    private String firstName;
    private String lastName;

    @Id @GeneratedValue(strategy = GenerationType.AUTO)
    public Long getId() {
        return id;
    }

    @Column(length=50)
    public String getFirstName() {

```

```

        return firstName;
    }
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
    public String getLastName() {
        return lastName;
    }
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
    @Override
    public boolean equals(Object a) {
        if(a == null)
            return false;
        if(!a.getClass().equals(getClass()))
            return false;
        Person p = (Person)a;
        return      firstName.equals(p.firstName)      &&
lastName.equals(p.lastName);
    }
    @Override
    public int hashCode() {
        return firstName.hashCode() + lastName.hashCode()*12;
    }
    @Override
    public String toString() {
        // TODO Auto-generated method stub
        return firstName + " " + lastName;
    }
}

```

### *PrescriptionModel*

```

package gr.edu.webapp.model;

import java.util.Calendar;
import java.util.Date;

import javax.persistence.*;

```

```

import org.appfuse.model.BaseObject;

@Entity
public class Prescription extends BaseObject {
    Long id;
    String notes;
    Date randevous;
    Person patient;
    Examination examination;

    public void setExamination(Examination examination)
    {
        this.examination = examination;
    }

    @ManyToOne(fetch = FetchType.EAGER)
    @JoinColumn(name = "EXAMINATION_ID", nullable = false)
    public Examination getExamination()
    {
        return examination;
    }

    public void setPatient(Person patient)
    {
        this.patient = patient;
    }

    @ManyToOne(fetch = FetchType.EAGER)
    @JoinColumn(name = "PERSON_ID", nullable = false)
    public Person getPatient()
    {
        return patient;
    }

    @Id @GeneratedValue(strategy = GenerationType.AUTO)
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }

    public String getNotes() {
        return notes;
    }
    public void setNotes(String notes) {
        this.notes = notes;
    }
}
@Temporal(javax.persistence.TemporalType.TIMESTAMP)
public Date getRandevous() {

```



```

        return randevous;
    }
    public void setRandevous(Date randevous) {
        this.randevous = randevous;
    }
    @Override
    public boolean equals(Object a) {
        if(a == null)
            return false;
        if(!a.getClass().equals(getClass()))
            return false;
        Prescription p = (Prescription)a;
        if(!p.examination.equals(examination)) return false;
        if(!p.notes.equals(notes)) return false;
        return randevous.getTime() == p.randevous.getTime();
    }
    @Override
    public int hashCode() {
        return examination.hashCode() * 7*notes.hashCode() *
37*((int)randevous.getTime()%Integer.MAX_VALUE);
    }
    @Override
    public String toString() {
        // TODO Auto-generated method stub
        return null;
    }
}
}

```

### *RootObjectModel*

```

package gr.edu.webapp.model;

import org.appfuse.model.BaseObject;

public abstract class RootObject extends BaseObject {
    protected Long id;

    public void setId(Long id) {
        this.id = id;
    }

    @Override

```

```

    public boolean equals(Object a) {
        if(a == null)
            return false;
        if(!a.getClass().equals(getClass()))
            return false;
        return true;
    }
}

```

## 8.3 Manager Classes

### *ExaminationManager*

```

package gr.edu.webapp.service;

import java.util.List;

import gr.edu.webapp.model.Examination;
import gr.edu.webapp.model.Labaratory;

import org.appfuse.service.GenericManager;

public interface ExaminationManager extends
GenericManager<Examination, Long> {
    List<Examination> findByLabaratory(Labaratory lab);
}

```

### *PrescriptionManager*

```

package gr.edu.webapp.service;

import gr.edu.webapp.model.*;

import java.util.List;

import org.appfuse.service.GenericManager;

public interface PrescriptionManager extends
GenericManager<Prescription, Long> {
    List<Prescription> findByExamination(Examination ex);
    List<Prescription> findByPatient(Person patient);
}

```

### *ExaminationManagerImpl*

```

package gr.edu.webapp.service.impl;

```

```

import java.util.List;

import gr.edu.webapp.dao.ExaminationDao;
import gr.edu.webapp.model.Examination;
import gr.edu.webapp.model.Laboratory;
import gr.edu.webapp.service.ExaminationManager;

import org.appfuse.service.impl.GenericManagerImpl;

public class ExaminationManagerImpl extends
GenericManagerImpl<Examination, Long> implements
    ExaminationManager {

    protected ExaminationDao examinationDao;
    public ExaminationManagerImpl(ExaminationDao dao)
    {
        super(dao);
        examinationDao =dao;
    }

    @Override
    public List<Examination> findByLaboratory(Laboratory lab) {
        return examinationDao.getByLaboratory(lab);
    }

}

```

### *PrescriptionManagerImpl*

```

package gr.edu.webapp.service.impl;

import gr.edu.webapp.dao.ExaminationDao;
import gr.edu.webapp.dao.PrescriptionDao;
import gr.edu.webapp.model.Examination;
import gr.edu.webapp.model.Laboratory;
import gr.edu.webapp.model.Person;
import gr.edu.webapp.model.Prescription;
import gr.edu.webapp.service.ExaminationManager;
import gr.edu.webapp.service.PrescriptionManager;

import java.util.List;

import org.appfuse.service.impl.GenericManagerImpl;

public class PrescriptionManagerImpl extends
    GenericManagerImpl<Prescription, Long> implements
PrescriptionManager {

    protected PrescriptionDao prescriptionDao;

```

```

    public PrescriptionManagerImpl(PrescriptionDao dao) {
        super(dao);
        prescriptionDao = dao;
    }

    @Override
    public List<Prescription> findByExamination(Examination ex) {
        return prescriptionDao.getByExamination(ex);
    }

    @Override
    public List<Prescription> findByPatient(Person patient) {
        return prescriptionDao.getByPerson(patient);
    }
}

```

## 8.4 Views

### *ExaminationForm*

```

<%@ include file="/common/taglibs.jsp"%>
<head>
    <title><fmt:message key="examinationDetail.title"/></title>
    <meta name="menu" content="examinationMenu"/>
</head>

<c:set          var="delObject"                scope="request"><fmt:message
key="userList.user"/></c:set>
<script type="text/javascript">var msgDelConfirm =
    <fmt:message          key="delete.confirm"><fmt:param
value="\${delObject}"/></fmt:message>;
</script>

<div class="span2">
    <h2><fmt:message key='examinationDetail.heading'/></h2>
</div>
<div class="span7">
    <form:errors path="*" cssClass="alert alert-error fade in"
element="div"/>
    <form:form          commandName="examination"          method="post"
action="examinationform" id="examinationForm"
cssClass="well form-horizontal" onsubmit="return
validateexamination(this)">
        <form:hidden path="id"/>
        <spring:bind path="examination.description">
            <div class="control-group${(not empty status.errorMessage) ? '
error' : ''}">
                </spring:bind>

```

```

        <appfuse:label                                styleClass="control-label"
key="examination.description"/>
        <div class="controls">
            <form:input path="description" id="description"/>
            <form:errors path="description" cssClass="help-inline"/>
        </div>
    </div>
    <spring:bind path="examination.requiredMinutes">
    <div class="control-group${(not empty status.errorMessage) ? '
error' : ''}">
        </spring:bind>
        <appfuse:label                                styleClass="control-label"
key="examination.requiredMinutes"/>
        <div class="controls">
            <form:input path="requiredMinutes" id="requiredMinutes"
/>
            <form:errors path="requiredMinutes" cssClass="help-
inline"/>
        </div>
    </div>

    <div class="form-actions">
        <button type="submit" class="btn btn-primary" name="save"
onclick="bCancel=false">
            <i class="icon-ok icon-white"></i> <fmt:message
key="button.save"/>
        </button>
        <c:if test="${not empty examination.id}">
            <button type="submit" class="btn" name="delete"
onclick="bCancel=true;return confirmMessage(msgDelConfirm)">
                <i class="icon-trash"></i> <fmt:message
key="button.delete"/>
            </button>
        </c:if>
        <button type="submit" class="btn" name="cancel"
onclick="bCancel=true">
            <i class="icon-remove"></i> <fmt:message
key="button.cancel"/>
        </button>
    </div>
</form:form>
</div>

<v:javascript formName="examination" cdata="false"
dynamicJavascript="true" staticJavascript="false"/>
<script type="text/javascript" src="<c:url
value='/scripts/validator.jsp'/>"></script>

<script type="text/javascript">
    $(document).ready(function() {
        $("input[type='text']:visible:enabled:first",
document.forms['examinationForm']).focus();
    });

```

```
</script>
```

## Examinations

```
<%@ include file="/common/taglibs.jsp"%>
<head>
  <title><fmt:message key="examinationsList.title"/></title>
  <meta name="menu" content="LaboratoryMenu"/>
</head>
<div class="span10">
  <h2><fmt:message key='examinationList.heading'/></h2>
  <h3>${laboratory.description}</h3>
  <div id="actions" class="form-actions">
    <a class="btn btn-primary" href="<c:url
value='/examinationform/${laboratory.id}/' />">
      <i class="icon-plus icon-white"></i> <fmt:message
key="button.add"/></a>
    <a class="btn" href="<c:url value='/mainMenu' />">
      <i class="icon-ok"></i> <fmt:message
key="button.done"/></a>
  </div>
  <display:table name="examinationList" class="table table-condensed
table-striped table-hover" requestURI="" id="examinationList"
export="true" pagesize="25">
    <display:column property="id" sortable="true"
href="examinationform/${laboratory.id}/" media="html"
paramId="id" paramProperty="id"
titleKey="examination.id"/>
    <display:column property="id" media="csv excel xml pdf"
titleKey="examination.id"/>
    <display:column property="description" sortable="true"
titleKey="examination.description"/>
    <display:column property="requiredMinutes" sortable="true"
titleKey="examination.requiredMinutes"/>

    <display:setProperty
name="paging.banner.item_name"><fmt:message
key="examinationList.examination"/></display:setProperty>
    <display:setProperty
name="paging.banner.items_name"><fmt:message
key="examinationList.examinations"/></display:setProperty>
    <display:setProperty name="export.excel.filename"><fmt:message
key="examinationList.title"/>.xls</display:setProperty>
    <display:setProperty name="export.csv.filename"><fmt:message
key="examinationList.title"/>.csv</display:setProperty>
    <display:setProperty name="export.pdf.filename"><fmt:message
key="examinationList.title"/>.pdf</display:setProperty>
  </display:table>
</div>
```

## Laboratories

```
<%@ include file="/common/taglibs.jsp"%>
<head>
  <title><fmt:message key="laboratoriesList.title"/></title>
  <meta name="menu" content="LaboratoryMenu"/>
</head>
<div class="span10">
  <h2><fmt:message key='laboratoryList.heading'/></h2>
  <div id="actions" class="form-actions">
    <a class="btn btn-primary" href="<c:url
value='/laboratoryform'/>">
      <i class="icon-plus icon-white"></i> <fmt:message
key="button.add"/></a>
    <a class="btn" href="<c:url value='/mainMenu'/>">
      <i class="icon-ok"></i> <fmt:message
key="button.done"/></a>
  </div>
  <display:table name="laboratoryList" class="table table-condensed
table-striped table-hover" requestURI="" id="laboratoryList"
export="true" pagesize="25">
    <display:column property="id" sortable="true"
href="laboratoryform" media="html"
paramId="id" paramProperty="id" titleKey="laboratory.id"/>
    <display:column property="id" media="csv excel xml pdf"
titleKey="laboratory.id"/>
    <display:column property="description" sortable="true"
titleKey="laboratory.description"/>
    <display:column property="startHour" sortable="true"
titleKey="laboratory.startHour"/>
    <display:column property="stopHour" sortable="true"
titleKey="laboratory.stopHour"/>
    <display:column media="html"
titleKey="laboratory.actions"><a
href="examinations/${laboratoryList.id}"><fmt:message>laboratory.exami
nations</fmt:message></a>
      <a
href="prescriptions/laboratory/${laboratoryList.id}"><fmt:message>laba
ratory.prescriptions</fmt:message></a></display:column>
    <display:setProperty
name="paging.banner.item_name"><fmt:message
key="laboratoryList.laboratory"/></display:setProperty>
    <display:setProperty
name="paging.banner.items_name"><fmt:message
key="laboratoryList.laboratories"/></display:setProperty>
    <display:setProperty name="export.excel.filename"><fmt:message
key="laboratoryList.title"/>.xls</display:setProperty>
    <display:setProperty name="export.csv.filename"><fmt:message
key="laboratoryList.title"/>.csv</display:setProperty>
    <display:setProperty name="export.pdf.filename"><fmt:message
key="laboratoryList.title"/>.pdf</display:setProperty>
  </display:table>
</div>
```

## LaboratoryForm

```
<%@ include file="/common/taglibs.jsp"%>
<head>
  <title><fmt:message key="laboratoryDetail.title"/></title>
  <meta name="menu" content="LaboratoryMenu"/>
</head>

<c:set          var="delObject"                scope="request"><fmt:message
key="userList.user"/></c:set>
<script type="text/javascript">var msgDelConfirm =
  "<fmt:message          key="delete.confirm"><fmt:param
value="\${delObject}"/></fmt:message>";
</script>

<div class="span2">
  <h2><fmt:message key='laboratoryDetail.heading'/></h2>
</div>
<div class="span7">
  <form:errors path="*" cssClass="alert alert-error fade in"
element="div"/>
  <form:form          commandName="laboratory"          method="post"
action="laboratoryform" id="laboratoryForm"
          cssClass="well          form-horizontal"          onsubmit="return
validateLaboratory(this)">
    <form:hidden path="id"/>
    <spring:bind path="laboratory.description">
      <div class="control-group${(not empty status.errorMessage) ? '
error' : ''}">
        </spring:bind>
        <appfuse:label          styleClass="control-label"
key="laboratory.description"/>
        <div class="controls">
          <form:input path="description" id="description"/>
          <form:errors path="description" cssClass="help-inline"/>
        </div>
      </div>
      <spring:bind path="laboratory.startHour">
        <div class="control-group${(not empty status.errorMessage) ? '
error' : ''}">
          </spring:bind>
          <appfuse:label          styleClass="control-label"
key="laboratory.startHour"/>
          <div class="controls">
            <form:input path="startHour" id="startHour" />
            <form:errors path="startHour" cssClass="help-inline"/>
          </div>
        </div>
      </div>
      <spring:bind path="laboratory.stopHour">
```



```

    <div class="control-group${(not empty status.errorMessage) ? '
error' : ''}">
      </spring:bind>
      <appfuse:label                                styleClass="control-label"
key="laboratory.stopHour"/>
      <div class="controls">
        <form:input path="stopHour" id="stopHour" />
        <form:errors path="stopHour" cssClass="help-inline"/>
      </div>
    </div>
    <div class="form-actions">
      <button type="submit" class="btn btn-primary" name="save"
onclick="bCancel=false">
        <i class="icon-ok icon-white"></i> <fmt:message
key="button.save"/>
      </button>
      <c:if test="${not empty laboratory.id}">
        <button type="submit" class="btn" name="delete"
onclick="bCancel=true;return confirmMessage(msgDelConfirm)">
          <i class="icon-trash"></i> <fmt:message
key="button.delete"/>
        </button>
      </c:if>
      <button type="submit" class="btn" name="cancel"
onclick="bCancel=true">
        <i class="icon-remove"></i> <fmt:message
key="button.cancel"/>
      </button>
    </div>
  </form:form>
</div>

<v:javascript formName="laboratory" cdata="false"
dynamicJavascript="true" staticJavascript="false"/>
<script type="text/javascript" src="<c:url
value='/scripts/validator.jsp'/"></script>

<script type="text/javascript">
  $(document).ready(function() {
    $("input[type='text']:visible:enabled:first",
document.forms['laboratoryForm']).focus();
  });
</script>

```

## MainMenu

```

<%@ include file="/common/taglibs.jsp"%>

<head>
  <title><fmt:message key="mainMenu.title"/></title>

```

```

    <meta name="menu" content="MainMenu"/>
</head>
<body class="home">

<h2><fmt:message key="mainMenu.heading"/></h2>
<p><fmt:message key="mainMenu.message"/></p>

<ul class="glassList">
  <li>
    <a href="<c:url value='<u>userform</u>' />"><fmt:message
key="menu.user"/></a>
  </li>
  <li>
    <a href="<c:url value='<u>fileupload</u>' />"><fmt:message
key="menu.selectFile"/></a>
  </li>
  <li>
    <a href="<c:url value="/persons"/>"><fmt:message
key="menu.viewPeople"/></a>
  </li>
  <li>
    <a href="<c:url value="/labaratories"/>"><fmt:message
key="menu.viewLabs"/></a>
  </li>
</ul>
</body>

```

## *PersonForm*

```

<%@ include file="/common/taglibs.jsp"%>
<head>
  <title><fmt:message key="personDetail.title"/></title>
  <meta name="menu" content="PersonMenu"/>
</head>

<c:set var="delObject" scope="request"><fmt:message
key="userList.user"/></c:set>
<script type="text/javascript">var msgDelConfirm =
  "<fmt:message key="delete.confirm"><fmt:param
value="{delObject}"/></fmt:message>";
</script>

<div class="span2">
  <h2><fmt:message key='personDetail.heading' /></h2>
</div>
<div class="span7">
  <form:errors path="*" cssClass="alert alert-error fade in"
element="div"/>
  <form:form commandName="person" method="post" action="<u>personform</u>"
id="personForm"

```

```

                cssClass="well        form-horizontal"        onsubmit="return
validatePerson(this)">
        <form:hidden path="id"/>
        <spring:bind path="person.firstName">
        <div class="control-group${(not empty status.errorMessage) ? '
error' : ''}">
                <spring:bind>
                <appfuse:label                                styleClass="control-label"
key="person.firstName"/>
                <div class="controls">
                        <form:input                path="firstName"                id="firstName"
maxlength="50"/>
                        <form:errors path="firstName" cssClass="help-inline"/>
                </div>
                </div>
                <spring:bind path="person.lastName">
                <div class="control-group${(not empty status.errorMessage) ? '
error' : ''}">
                        <spring:bind>
                                <appfuse:label                                styleClass="control-label"
key="person.lastName"/>
                                <div class="controls">
                                        <form:input path="lastName" id="lastName" maxlength="50"/>
                                        <form:errors path="lastName" cssClass="help-inline"/>
                                </div>
                                </div>
                                <spring:bind path="person.birthDay">
                                <div class="control-group${(not empty status.errorMessage) ? '
error' : ''}">
                                        <spring:bind>
                                                <appfuse:label                                styleClass="control-label"
key="person.birthDay"/>
                                                <div class="controls">
                                                        <form:input path="birthDay" id="lastName" />
                                                        <form:errors path="birthDay" cssClass="help-inline"/>
                                                </div>
                                                </div>
                                                <div class="form-actions">
                                                        <button type="submit" class="btn btn-primary" name="save"
onclick="bCancel=false">
                                                                <i class="icon-ok icon-white"></i> <fmt:message
key="button.save"/>
                                                                </button>
                                                                <c:if test="${not empty person.id}">
                                                                <button type="submit" class="btn" name="delete"
onclick="bCancel=true;return confirmMessage(msgDelConfirm)">
                                                                <i class="icon-trash"></i> <fmt:message
key="button.delete"/>
                                                                </button>
                                                                </c:if>
                                                                <button type="submit" class="btn" name="cancel"
onclick="bCancel=true">

```

```

        <i class="icon-remove"></i> <fmt:message
key="button.cancel"/>
    </button>
</div>
</form:form>
</div>

<v:javascript formName="person" cdata="false" dynamicJavascript="true"
staticJavascript="false"/>
<script type="text/javascript" src="<c:url
value='/scripts/validator.jsp'/>"></script>

<script type="text/javascript">
    $(document).ready(function() {
        $("input[type='text']:visible:enabled:first",
document.forms['personForm']).focus();
    });
</script>

```

## PersonPrescription

```

<%@ include file="/common/taglibs.jsp"%>
<head>
<title><fmt:message key="personList.title" /></title>
<meta name="menu" content="PersonMenu" />
</head>
<div class="span10">
    <h2>${patient.firstName}&nbsp;${patient.lastName}</h2>
    <p>
        <fmt:message>examination.select</fmt:message>
    </p>
    <form:form commandName="prescription" method="post"
        action="../../personprescriptionform/${patient.id}/examinationfor
m" id="personPrescription"
        cssClass="well form-horizontal"
modelAttribute="prescription">
        <form:hidden path="id"/>
        <form:select path="examination" size="1">
            <form:options items="${examinationList}"
itemValue="id" />
        </form:select>
        <div class="form-actions">
            <button type="submit" class="btn btn-primary"
name="save"
                onclick="bCancel=false">
                <i class="icon-ok icon-white"></i>
                <fmt:message key="button.save" />
            </button>
            <button type="submit" class="btn" name="cancel"

```

```

        onclick="bCancel=true">
        <i class="icon-remove"></i>
        <fmt:message key="button.cancel" />
    </button>
</div>
</form:form>
</div>

```

## Persons

```

<%@ include file="/common/taglibs.jsp"%>
<head>
    <title><fmt:message key="personList.title"/></title>
    <meta name="menu" content="PersonMenu"/>
</head>
<div class="span10">
    <h2><fmt:message key='personList.heading'/></h2>
    <div id="actions" class="form-actions">
        <a class="btn btn-primary" href="<c:url
value='/personform'/>">
            <i class="icon-plus icon-white"></i> <fmt:message
key="button.add"/></a>
        <a class="btn" href="<c:url value='/mainMenu'/>">
            <i class="icon-ok"></i> <fmt:message
key="button.done"/></a>
    </div>
    <display:table name="personList" class="table table-condensed
table-striped table-hover" requestURI="" id="personList" export="true"
pagesize="25">
        <display:column property="id" sortable="true"
href="personform" media="html"
paramId="id" paramProperty="id" titleKey="person.id"/>
        <display:column property="id" media="csv excel xml pdf"
titleKey="person.id"/>
        <display:column property="firstName" sortable="true"
titleKey="person.firstName"/>
        <display:column property="lastName" sortable="true"
titleKey="person.lastName"/>

        <display:column media="html"
titleKey="persons.actions"><a
href="prescriptions/patient/{personList.id}"><fmt:message>person.pres
cription</fmt:message></a></display:column>

        <display:setProperty
name="paging.banner.item_name"><fmt:message
key="personList.person"/></display:setProperty>
        <display:setProperty
name="paging.banner.items_name"><fmt:message
key="personList.persons"/></display:setProperty>

```

```

        <display:setProperty name="export.excel.filename"><fmt:message
key="personList.title"/>.xls</display:setProperty>
        <display:setProperty name="export.csv.filename"><fmt:message
key="personList.title"/>.csv</display:setProperty>
        <display:setProperty name="export.pdf.filename"><fmt:message
key="personList.title"/>.pdf</display:setProperty>
    </display:table>
</div>

```

## PrescriptionForm

```

<%@ include file="/common/taglibs.jsp"%>
<head>
    <title><fmt:message key="prescriptionDetail.title"/></title>
    <meta name="menu" content="prescriptionMenu"/>
</head>

<c:set          var="delObject"                scope="request"><fmt:message
key="userList.user"/></c:set>
<script type="text/javascript">var msgDelConfirm =
    "<fmt:message          key="delete.confirm"><fmt:param
value="\${delObject}"/></fmt:message>";
</script>

<div class="span2">
    <h2><fmt:message key='prescriptionDetail.heading'/></h2>
</div>
<div class="span7">
    <form:errors path="*" cssClass="alert alert-error fade in"
element="div"/>
    <form:form          commandName="prescription"          method="post"
action="prescriptionform" id="prescriptionform"
          cssClass="well          form-horizontal"          onsubmit="return
validatePrescription(this)">
        <form:hidden path="id"/>
        <form:select path="examination.id" size="1">
            <form:options items="\${examinationList}" itemValue="id" />
        </form:select>
        <div class="form-actions">
            <button type="submit" class="btn btn-primary" name="save"
onclick="bCancel=false">
                <i class="icon-ok icon-white"></i> <fmt:message
key="button.save"/>
            </button>
            <c:if test="\${not empty prescription.id}">
                <button type="submit" class="btn" name="delete"
onclick="bCancel=true;return confirmMessage(msgDelConfirm)">
                    <i class="icon-trash"></i> <fmt:message
key="button.delete"/>
                </button>
            </c:if>
        </div>
    </form:form>
</div>

```

```

        <button type="submit" class="btn" name="cancel"
onclick="bCancel=true">
        <i class="icon-remove"></i> <fmt:message
key="button.cancel"/>
        </button>
    </div>
</form:form>
</div>

```

```

<:javascript formName="prescription" cdata="false"
dynamicJavascript="true" staticJavascript="false"/>
<script type="text/javascript" src="<c:url
value='/scripts/validator.jsp'/>"></script>

<script type="text/javascript">
    $(document).ready(function() {
        $("input[type='text']:visible:enabled:first",
document.forms['prescriptionForm']).focus();
    });
</script>

```

## Prescriptions

```

<%@ include file="/common/taglibs.jsp"%>
<head>
    <title><fmt:message key="prescriptionsList.title"/></title>
    <meta name="menu" content="PrescriptionMenu"/>
</head>
<div class="span10">
    <h2><fmt:message key='prescriptionList.heading'/></h2>
    <h3>${laboratory.description}</h3>
    <h3>${person.firstName}&nbsp;${person.lastName}</h3>

    <c:if test="${type eq 'laboratory'}">
    </c:if>
    <c:if test="${type eq 'person'}">
    <div id="actions" class="form-actions">
        <a class="btn btn-primary" href="<c:url
value='/prescriptionform/${person.id}"/>">
            <i class="icon-plus icon-white"></i> <fmt:message
key="button.add"/></a>
        <a class="btn" href="<c:url value='/mainMenu'/>">
            <i class="icon-ok"></i> <fmt:message
key="button.done"/></a>
    </div>
    </c:if>
    <display:table name="prescriptionList" class="table table-
condensed table-striped table-hover" requestURI=""
id="prescriptionList" export="true" pagesize="25">
        <display:column property="id" sortable="true"
titleKey="prescription.id"/>

```

```

        <display:column        property="randevous"        sortable="true"
titleKey="prescription.randevous-date" format="{0,date,short}"/>
        <display:column        property="randevous"        sortable="true"
titleKey="prescription.randevous-time" format="{0,time}"/>
        <display:column        property="patient"        sortable="true"
titleKey="prescription.patient"/>
        <display:column        property="examination"        sortable="true"
titleKey="prescription.examination"/>

        <display:setProperty
name="paging.banner.item_name"><fmt:message
key="prescriptionList.prescription"/></display:setProperty>
        <display:setProperty
name="paging.banner.items_name"><fmt:message
key="prescriptionList.prescription"/></display:setProperty>
        <display:setProperty name="export.excel.filename"><fmt:message
key="prescriptionList.title"/>.xls</display:setProperty>
        <display:setProperty name="export.csv.filename"><fmt:message
key="prescriptionList.title"/>.csv</display:setProperty>
        <display:setProperty name="export.pdf.filename"><fmt:message
key="prescriptionList.title"/>.pdf</display:setProperty>
        </display:table>
</div>

```

## Signup

```

<%@ include file="/common/taglibs.jsp" %>

<head>
    <title><fmt:message key="signup.title"/></title>
</head>

<body class="signup"/>

<div class="span2">
    <h2><fmt:message key="signup.heading"/></h2>
    <p><fmt:message key="signup.message"/></p>
</div>
<div class="span7">
    <spring:bind path="user.*">
        <c:if test="${not empty status.errorMessages}">
            <div class="alert alert-error fade in"
class="close">&times;</a>
                <a href="#" data-dismiss="alert"
                <c:forEach var="error"
items="${status.errorMessages}">
                    <c:out value="${error}" escapeXml="false"/><br/>
                </c:forEach>
            </div>
        </c:if>
    </spring:bind>

```



```

    <form:form    commandName="user"    method="post"    action="signup"
id="signupForm" autocomplete="off"
                cssClass="well    form-horizontal"    onsubmit="return
validateUser(this)">
    <spring:bind path="user.username">
    <fieldset    class="control-group${(not    empty
status.errorMessage) ? ' error' : ''}">
    </spring:bind>
    <appfuse:label    styleClass="control-label"
key="user.username"/>
    <div class="controls">
    <form:input path="username" id="username"/>
    <form:errors path="username" cssClass="help-inline"/>
    </div>
    </fieldset>
    <spring:bind path="user.password">
    <fieldset    class="control-group${(not    empty
status.errorMessage) ? ' error' : ''}">
    </spring:bind>
    <appfuse:label    styleClass="control-label"
key="user.password"/>
    <div class="controls">
    <form:password    path="password"    id="password"
showPassword="true"/>
    <form:errors path="password" cssClass="help-inline"/>
    </div>
    </fieldset>
    <spring:bind path="user.confirmPassword">
    <fieldset    class="control-group${(not    empty
status.errorMessage) ? ' error' : ''}">
    </spring:bind>
    <appfuse:label    styleClass="control-label"
key="user.confirmPassword"/>
    <div class="controls">
    <form:password    path="confirmPassword"
id="confirmPassword" showPassword="true"/>
    <form:errors path="confirmPassword" cssClass="help-
inline"/>
    </div>
    </fieldset>
    <spring:bind path="user.passwordHint">
    <fieldset    class="control-group${(not    empty
status.errorMessage) ? ' error' : ''}">
    </spring:bind>
    <appfuse:label    styleClass="control-label"
key="user.passwordHint"/>
    <div class="controls">
    <form:input path="passwordHint" id="passwordHint"/>
    <form:errors path="passwordHint" cssClass="help-
inline"/>
    </div>
    </fieldset>
    <spring:bind path="user.firstName">

```

```

        <fieldset class="control-group${(not
status.errorMessage) ? ' error' : ''}">
            </spring:bind>
            <appfuse:label styleClass="control-label"
key="user.firstName"/>
            <div class="controls">
                <form:input path="firstName" id="firstName"
maxlength="50"/>
                <form:errors path="firstName" cssClass="help-inline"/>
            </div>
        </fieldset>
        <spring:bind path="user.lastName">
            <fieldset class="control-group${(not
status.errorMessage) ? ' error' : ''}">
                </spring:bind>
                <appfuse:label styleClass="control-label"
key="user.lastName"/>
                <div class="controls">
                    <form:input path="lastName" id="lastName"
maxlength="50"/>
                    <form:errors path="lastName" cssClass="help-inline"/>
                </div>
            </fieldset>
            <spring:bind path="user.email">
                <fieldset class="control-group${(not
status.errorMessage) ? ' error' : ''}">
                    </spring:bind>
                    <appfuse:label styleClass="control-label"
key="user.email"/>
                    <div class="controls">
                        <form:input path="email" id="email"/>
                        <form:errors path="email" cssClass="help-inline"/>
                    </div>
                </fieldset>
                <fieldset class="control-group">
                    <appfuse:label styleClass="control-label"
key="user.phoneNumber"/>
                    <div class="controls">
                        <form:input path="phoneNumber" id="phoneNumber"/>
                    </div>
                </fieldset>
                <fieldset class="control-group">
                    <appfuse:label styleClass="control-label"
key="user.website"/>
                    <div class="controls">
                        <form:input path="website" id="website"/>
                    </div>
                </fieldset>
                <fieldset>
                    <legend class="accordion-heading">
                        <a data-toggle="collapse" href="#collapse-
address"><fmt:message key="user.address.address"/></a>
                    </legend>

```

```

        <div id="collapse-address" class="accordion-body
collapse">
            <fieldset class="control-group">
                <appfuse:label styleClass="control-label"
key="user.address.address"/>
                <div class="controls">
                    <form:input path="address.address"
id="address.address"/>
                </div>
            </fieldset>
            <fieldset class="control-group">
                <appfuse:label styleClass="control-label"
key="user.address.city"/>
                <div class="controls">
                    <form:input path="address.city"
id="address.city"/>
                </div>
            </fieldset>
            <fieldset class="control-group">
                <appfuse:label styleClass="control-label"
key="user.address.province"/>
                <div class="controls">
                    <form:input path="address.province"
id="address.province"/>
                </div>
            </fieldset>
            <fieldset class="control-group">
                <appfuse:label styleClass="control-label"
key="user.address.postalCode"/>
                <div class="controls">
                    <form:input path="address.postalCode"
id="address.postalCode"/>
                </div>
            </fieldset>
            <fieldset class="control-group">
                <appfuse:label styleClass="control-label"
key="user.address.country"/>
                <div class="controls">
                    <appfuse:country name="address.country"
prompt="" default="{user.address.country}"/>
                </div>
            </fieldset>
        </div>
        <fieldset class="form-actions">
            <button type="submit" class="btn btn-primary" name="save"
onclick="bCancel=false">
                <i class="icon-ok icon-white"></i> <fmt:message
key="button.register"/>
            </button>
            <button type="submit" class="btn" name="cancel"
onclick="bCancel=true">

```

```

                <i class="icon-remove"></i> <fmt:message
key="button.cancel"/>
            </button>
        </fieldset>
    </form:form>
</div>

```

```

<c:set var="scripts" scope="request">
<v:javascript formName="user" staticJavascript="false"/>
<script type="text/javascript" src="<c:url
value="/scripts/validator.jsp"/>"></script>
<script type="text/javascript">
    $(document).ready(function() {
        $("input[type='text']:visible:enabled:first",
document.forms['signupForm']).focus();
    });
</script>
</c:set>

```

## UserForm

```

<%@ include file="/common/taglibs.jsp"%>

<head>
    <title><fmt:message key="userProfile.title"/></title>
    <meta name="menu" content="UserMenu"/>
</head>

<c:set var="delObject" scope="request"><fmt:message
key="userList.user"/></c:set>
<script type="text/javascript">var msgDelConfirm =
    "<fmt:message key="delete.confirm"><fmt:param
value="${delObject}"/></fmt:message>";
</script>

<div class="span2">
    <h2><fmt:message key="userProfile.heading"/></h2>
    <c:choose>
        <c:when test="${param.from == 'list'}">
            <p><fmt:message key="userProfile.admin.message"/></p>
        </c:when>
        <c:otherwise>
            <p><fmt:message key="userProfile.message"/></p>
        </c:otherwise>
    </c:choose>
</div>
<div class="span7">
    <spring:bind path="user.*">
        <c:if test="${not empty status.errorMessages}">
            <div class="alert alert-error fade in">
                <a href="#" data-dismiss="alert"
class="close">&times;</a>

```

```

                <c:forEach                                var="error"
items="${status.errorMessages}">
                    <c:out value="${error}" escapeXml="false"/><br/>
                </c:forEach>
            </div>
        </c:if>
    </spring:bind>

    <form:form    commandName="user"    method="post"    action="userform"
id="userForm"  autoComplete="off"
                cssClass="well    form-horizontal"    onSubmit="return
validateUser(this)">
        <form:hidden path="id"/>
        <form:hidden path="version"/>
        <input    type="hidden"    name="from"    value="<c:out
value="${param.from}"/>"/>

        <c:if test="${cookieLogin == 'true'}">
            <form:hidden path="password"/>
            <form:hidden path="confirmPassword"/>
        </c:if>

        <c:if test="${empty user.version}">
            <input type="hidden" name="encryptPass" value="true"/>
        </c:if>

        <spring:bind path="user.username">
            <fieldset    class="control-group${(not    empty
status.errorMessage) ? ' error' : ''}">
                </spring:bind>
                <appfuse:label    styleClass="control-label"
key="user.username"/>
                <div class="controls">
                    <form:input path="username" id="username"/>
                    <form:errors path="username" cssClass="help-inline"/>
                </div>
            </fieldset>
            <c:if test="${cookieLogin != 'true'}">
                <spring:bind path="user.password">
                    <fieldset    class="control-group${(not    empty
status.errorMessage) ? ' error' : ''}">
                        </spring:bind>
                        <appfuse:label    styleClass="control-label"
key="user.password"/>
                        <div class="controls">
                            <form:password    path="password"    id="password"
showPassword="true"/>
                            <form:errors path="password" cssClass="help-inline"/>
                        </div>
                    </fieldset>
                    <spring:bind path="user.confirmPassword">
                        <fieldset    class="control-group${(not    empty
status.errorMessage) ? ' error' : ''}">

```

```

        </spring:bind>
        <appfuse:label                                styleClass="control-label"
key="user.confirmPassword"/>
        <div class="controls">
            <form:password                            path="confirmPassword"
id="confirmPassword" showPassword="true"/>
            <form:errors    path="confirmPassword"    cssClass="help-
inline"/>
        </div>
    </fieldset>
</c:if>
    <spring:bind path="user.passwordHint">
    <fieldset                                class="control-group${(not        empty
status.errorMessage) ? ' error' : ''}">
        </spring:bind>
        <appfuse:label                                styleClass="control-label"
key="user.passwordHint"/>
        <div class="controls">
            <form:input path="passwordHint" id="passwordHint"/>
            <form:errors    path="passwordHint"    cssClass="help-
inline"/>
        </div>
    </fieldset>
    <spring:bind path="user.firstName">
    <fieldset                                class="control-group${(not        empty
status.errorMessage) ? ' error' : ''}">
        </spring:bind>
        <appfuse:label                                styleClass="control-label"
key="user.firstName"/>
        <div class="controls">
            <form:input            path="firstName"            id="firstName"
maxlength="50"/>
            <form:errors path="firstName" cssClass="help-inline"/>
        </div>
    </fieldset>
    <spring:bind path="user.lastName">
    <fieldset                                class="control-group${(not        empty
status.errorMessage) ? ' error' : ''}">
        </spring:bind>
        <appfuse:label                                styleClass="control-label"
key="user.lastName"/>
        <div class="controls">
            <form:input            path="lastName"            id="lastName"
maxlength="50"/>
            <form:errors path="lastName" cssClass="help-inline"/>
        </div>
    </fieldset>
    <spring:bind path="user.email">
    <fieldset                                class="control-group${(not        empty
status.errorMessage) ? ' error' : ''}">
        </spring:bind>
        <appfuse:label                                styleClass="control-label"
key="user.email"/>

```

```

        <div class="controls">
            <form:input path="email" id="email"/>
            <form:errors path="email" cssClass="help-inline"/>
        </div>
    </fieldset>
    <fieldset class="control-group">
        <appfuse:label                                styleClass="control-label"
key="user.phoneNumber"/>
        <div class="controls">
            <form:input path="phoneNumber" id="phoneNumber"/>
        </div>
    </fieldset>
    <fieldset class="control-group">
        <appfuse:label                                styleClass="control-label"
key="user.website"/>
        <div class="controls">
            <form:input path="website" id="website"/>
        </div>
    </fieldset>
    <fieldset>
        <legend class="accordion-heading">
            <a          data-toggle="collapse"          href="#collapse-
address"><fmt:message key="user.address.address"/></a>
        </legend>
        <div          id="collapse-address"          class="accordion-body
collapse">
            <fieldset class="control-group">
                <appfuse:label                    styleClass="control-label"
key="user.address.address"/>
                <div class="controls">
                    <form:input                    path="address.address"
id="address.address"/>
                </div>
            </fieldset>
            <fieldset class="control-group">
                <appfuse:label                    styleClass="control-label"
key="user.address.city"/>
                <div class="controls">
                    <form:input                    path="address.city"
id="address.city"/>
                </div>
            </fieldset>
            <fieldset class="control-group">
                <appfuse:label                    styleClass="control-label"
key="user.address.province"/>
                <div class="controls">
                    <form:input                    path="address.province"
id="address.province"/>
                </div>
            </fieldset>
            <fieldset class="control-group">
                <appfuse:label                    styleClass="control-label"
key="user.address.postalCode"/>

```

```

        <div class="controls">
            <form:input path="address.postalCode"
id="address.postalCode"/>
        </div>
    </fieldset>
    <fieldset class="control-group">
        <appfuse:label styleClass="control-label"
key="user.address.country"/>
        <div class="controls">
            <appfuse:country name="address.country"
prompt="" default="{user.address.country}"/>
        </div>
    </fieldset>
</div>
</fieldset>
<c:choose>
    <c:when test="{param.from == 'list' or param.method == 'Add'}">
        <fieldset class="control-group">
            <label class="control-label"><fmt:message
key="userProfile.accountSettings"/></label>
            <div class="controls">
                <label class="checkbox inline">
                    <form:checkbox path="enabled" id="enabled"/>
                    <fmt:message key="user.enabled"/>
                </label>

                <label class="checkbox inline">
                    <form:checkbox path="accountExpired"
id="accountExpired"/>
                    <fmt:message key="user.accountExpired"/>
                </label>

                <label class="checkbox inline">
                    <form:checkbox path="accountLocked"
id="accountLocked"/>
                    <fmt:message key="user.accountLocked"/>
                </label>

                <label class="checkbox inline">
                    <form:checkbox path="credentialsExpired"
id="credentialsExpired"/>
                    <fmt:message key="user.credentialsExpired"/>
                </label>
            </div>
        </fieldset>
        <fieldset class="control-group">
            <label for="userRoles" class="control-label"><fmt:message
key="userProfile.assignRoles"/></label>
            <div class="controls">
                <select id="userRoles" name="userRoles"
multiple="true">
                    <c:forEach items="{availableRoles}" var="role">

```



```

                <option
    ${fn:contains(user.roles, role.label)} ? 'selected' :
    ''>${role.label}</option>
            </c:forEach>
        </select>
    </div>
</fieldset>
</c:when>
<c:when test="{not empty user.username}">
    <fieldset class="control-group">
        <label class="control-label"><fmt:message
key="user.roles"/></label>
        <div class="controls readonly">
            <c:forEach var="role" items="{user.roleList}"
varStatus="status">
                <c:out value="{role.label}"/><c:if
test="{!status.last}">,</c:if>
                <input type="hidden" name="userRoles"
value="<c:out value="{role.label}"/>"/>
            </c:forEach>
        </div>
        <form:hidden path="enabled"/>
        <form:hidden path="accountExpired"/>
        <form:hidden path="accountLocked"/>
        <form:hidden path="credentialsExpired"/>
    </fieldset>
</c:when>
</c:choose>
    <fieldset class="form-actions">
        <button type="submit" class="btn btn-primary" name="save"
onclick="bCancel=false">
            <i class="icon-ok icon-white"></i> <fmt:message
key="button.save"/>
        </button>

        <c:if test="{param.from == 'list' and param.method !=
'Add'}">
            <button type="submit" class="btn" name="delete"
onclick="bCancel=true;return confirmMessage(msgDelConfirm)">
                <i class="icon-trash"></i> <fmt:message
key="button.delete"/>
            </button>
        </c:if>

        <button type="submit" class="btn" name="cancel"
onclick="bCancel=true">
            <i class="icon-remove"></i> <fmt:message
key="button.cancel"/>
        </button>
    </fieldset>
</form:form>
</div>

```

```

<c:set var="scripts" scope="request">
<script type="text/javascript">
    function passwordChanged(passwordField) {
        if (passwordField.id == "password") {
            var origPassword = "${user.password}";
        } else if (passwordField.id == "confirmPassword") {
            var origPassword = "${user.confirmPassword}";
        }

        if (passwordField.value != origPassword) {
            createFormElement("input", "hidden", "encryptPass",
"encryptPass",
                                "true", passwordField.form);
        }
    }

<!-- This is here so we can exclude the selectAll call when roles is
hidden -->
function onFormSubmit(theForm) {
    return validateUser(theForm);
}
</script>
</c:set>

<v:javascript formName="user" staticJavascript="false"/>
<script type="text/javascript" src="<c:url
value="/scripts/validator.jsp"/>"></script>

```