

HERMES

Functional Requirements & High Level Design

Version: 1.2

Status: Released

Ιστορικό Αναθεωρήσεων

Date	Version	Description	Authors
<10/12/2020>	1.0	Basic Functional Requirements & HLD Documented	Nick Dimitrakopoulos Paraskevi-Theofania Gourgioti Ioannis Christou
<19/2/2021>	1.1	Document Updated To Match Current Specs	Nick Dimitrakopoulos Paraskevi-Theofania Gourgioti Ioannis Christou
<5/7/2021>	1.2	Minor Document Updates	Nick Dimitrakopoulos

Πίνακας Περιεχομένων

Ιστορικό Αναθεωρήσεων	2
Πίνακας Περιεχομένων.....	3
Εισαγωγή	5
Σκοπός αυτού του εγγράφου	5
Λίγα λόγια για το Πληροφοριακό Σύστημα ΕΡΜΗΣ.....	5
Κύριοι στόχοι του ΠΣ ΕΡΜΗΣ.....	5
Λειτουργικές Απαιτήσεις.....	6
Τύποι χρηστών και λειτουργίες.....	6
Εσωτερικό σύστημα.....	6
Εξωτερικό σύστημα	8
Μοντέλο Δεδομένων.....	8
Εισαγωγή	8
Σχεσιακό Μοντέλο Του Keycloak	8
Χρήστες (Users).....	8
Γκρουπ (Groups).....	9
Ρόλοι (Roles)	9
Χρήστες	9
Γνωρίσματα του χρήστη	10
Χρήστες και Groups	10
Οργανισμοί	11
Ιεραρχία των Group για τους οργανισμούς.....	11
Γνωρίσματα του οργανισμού	11
Αιτήσεις.....	11
Γνωρίσματα της αίτησης.....	11
Καταστάσεις της αίτησης.....	12
High Level Design.....	13
Υποσυστήματα του ΠΣ.....	13
Συνοπτική Περιγραφή Αρχιτεκτονικής	14
Internal Front-End	14
External Front-End	14
Back-End	14
Keycloak.....	15
Παράρτημα	15
Use Case διαγράμματα.....	15
Εσωτερικό σύστημα.....	15
UC 1 - Διαχειριστής συστήματος.....	15
UC 2 - Διαχειριστής/στρια οργανισμών	17
UC 3 - Προϊστάμενος/η οργανισμού.....	18
UC 4 - Υπάλληλος οργανισμού.....	19

Εξωτερικό Σύστημα.....	20
UC 5 - Πολίτης	20
ER Διαγράμματα.....	21
ER 1 - ER διάγραμμα του Keycloak	21
State Machine Διαγράμματα	22
SM 1 - Καταστάσεις Αιτήσεων.....	22
Class Διαγράμματα	23
CD 1 – Αφηρημένο Class Diagram του συστήματος.....	23

Εισαγωγή

Σκοπός αυτού του εγγράφου

Σκοπός αυτού του εγγράφου είναι να τεκμηριωθούν οι λειτουργικές προδιαγραφές (Functional Requirements) και ο σχεδιασμός υψηλού επιπέδου (High Level Design) του πληροφοριακού συστήματος ΕΡΜΗΣ.

Ορισμός, Ακρωνύμια και Συνοτομογραφίες

Όρος	Σημασία
Keycloak	Λογισμικό ανοιχτού κώδικα, το οποίο παρέχει Single-Sign-On (SSO) και Identity and Access Management (IAM) δυνατότητες σε μοντέρνες εφαρμογές και υπηρεσίες.
Vaadin	Πλατφόρμα ανοιχτού κώδικα, η οποία παρέχει δυνατότητα κατασκευής ποιοτικών Web User Interfaces (Front-End), χρησιμοποιώντας Java.
Εσωτερικό Σύστημα	Αναφέρεται στο συνδυασμό των Internal Front-End και Back-End υποσυστημάτων. Για περισσότερες λεπτομέρειες, ανατρέξτε στην ενότητα Συνοπτική Περιγραφή Αρχιτεκτονικής .
Εξωτερικό Σύστημα	Αναφέρεται στο External Front-End υποσύστημα. Για περισσότερες λεπτομέρειες, ανατρέξτε στην ενότητα Συνοπτική Περιγραφή Αρχιτεκτονικής .

Συνοπτική Περιγραφή

Λίγα λόγια για το Πληροφοριακό Σύστημα ΕΡΜΗΣ

Το πληροφοριακό σύστημα ΕΡΜΗΣ έχει ως πρωταρχικό στόχο την καλύτερη οργάνωση των ραντεβού που επιθυμούν να κλείσουν οι πολίτες σε διάφορους δημόσιους οργανισμούς, οι οποίοι είναι ενταγμένοι στο σύστημα.

Κύριοι στόχοι του ΠΣ ΕΡΜΗΣ

- Να παρέχει στους πολίτες έναν εύκολο τρόπο να διαχειρίζονται τα ραντεβού τους σε διάφορους οργανισμούς του δημοσίου.
- Να απλοποιεί την διαχείριση των καταλόγων χρηστών των οργανισμών.
- Να απλοποιεί τις επιχειρηματικές διαδικασίες των οργανισμών ως προς την διαχείριση των αιτήσεων ραντεβού με τους πολίτες.

Λειτουργικές Απαιτήσεις

Τύποι χρηστών και λειτουργίες

Οι παρακάτω πίνακες διαχωρίζουν τους τύπους των End Users που πρόκειται να χρησιμοποιήσουν το πληροφοριακό σύστημα ΕΡΜΗΣ, καθώς και τι λειτουργίες δικαιούται να πραγματοποιεί ο καθένας.

Εσωτερικό σύστημα

Ονομασία:	Διαχειριστής/στρια Συστήματος (System Administrator)
Σύντομη Περιγραφή:	Ο Διαχειριστής Συστήματος είναι η οντότητα με την υψηλότερη εξουσία γιατί μπορεί να διαχειρίζεται εξολοκλήρου τον Authorization Server (Keycloak). Οι κύριες αρμοδιότητες του είναι η διαχείριση καταλόγου χρηστών των Διαχειριστών Οργανισμών
Use Case Diagram:	UC 1
Λειτουργικές Προδιαγραφές	
Διαχείριση Καταλόγου Χρηστών Διαχειριστών Οργανισμών	Πλήρη CRUD δικαιώματα, μέσω του Admin Panel του Keycloak (δηλαδή δεν απαιτείται υλοποίηση εκ μέρους μας).

Ονομασία:	Διαχειριστής/στρια οργανισμών (Organizations Administrator)
Σύντομη Περιγραφή:	Ο/Η Διαχειριστής/στρια οργανισμού διαχειρίζεται τους οργανισμούς του συστήματος, καθώς και τους προϊσταμένους του κάθε οργανισμού. Επίσης, διαχειρίζεται τον κατάλογο χρηστών των πολιτών.
Use Case Diagram:	UC 2
Λειτουργικές Προδιαγραφές	
Διαχείριση Οργανισμών	Ο/Η Διαχειριστής/στρια οργανισμών μπορεί: <ul style="list-style-type: none">• Να δημιουργεί οργανισμούς και να αναθέτει σε αυτούς έναν/μία ή περισσότερους/ες προϊσταμένους/ες.• Να βλέπει όλη τη λίστα των οργανισμών που έχουν δημιουργηθεί στο σύστημα.• Να ενημερώνει στοιχεία του οργανισμού (για παράδειγμα μετονομασία του οργανισμού ή ενημέρωση των προϊσταμένων).
Διαχείριση Καταλόγου Προϊσταμένων των Οργανισμών	Ο/Η Διαχειριστής/στρια οργανισμών μπορεί: <ul style="list-style-type: none">• Να δημιουργεί λογαριασμούς προϊσταμένων οργανισμού.• Να βλέπει τη λίστα των προϊσταμένων που έχουν δημιουργηθεί για κάθε οργανισμό.• Να ενημερώνει τα στοιχεία ενός προϊσταμένου.• Να απενεργοποιεί τον λογαριασμό ενός προϊσταμένου.
Διαχείριση Καταλόγου Πολιτών	Ο/Η Διαχειριστής/στρια οργανισμών μπορεί: <ul style="list-style-type: none">• Να δημιουργεί λογαριασμούς πολιτών.

	<ul style="list-style-type: none"> • Να βλέπει τη λίστα των πολιτών του συστήματος. • Να ενημερώνει τα στοιχεία των πολιτών. • Να απενεργοποιεί τον λογαριασμό ενός πολίτη.
Σημειώσεις	
1. Η διαγραφή ενός/μια προϋσταμένου/ής από έναν/μία διαχειριστή/στρια οργανισμών είναι μία επικίνδυνη ενέργεια διότι πιθανώς συνδέεται με δραστηριότητα του οργανισμού. Συνεπώς δεν θα επιτρέπεται η διαγραφή του λογαριασμού του/της, αλλά η απενεργοποίησή του. 2. Το παραπάνω ισχύει και στην περίπτωση των πολιτών.	

Ονομασία:	Προϊστάμενος/η Οργανισμού (Organization Supervisor)
Σύντομη Περιγραφή:	Ο/Η Προϊστάμενος/η Οργανισμού διαχειρίζεται τον κατάλογο χρηστών των υπαλλήλων του οργανισμού στον οποίο εργάζεται.
Use Case Diagram	UC 3
Λειτουργικές Προδιαγραφές	
Διαχείριση καταλόγου χρηστών των υπαλλήλων του οργανισμού στον οποίο εργάζεται ο/η προϊστάμενος/η.	Ο/Η προϊστάμενος/η ενός οργανισμού μπορεί: <ul style="list-style-type: none"> ▪ Να δημιουργεί λογαριασμούς χρηστών για νέους υπαλλήλους του οργανισμού. ▪ Να έχει δικαίωμα να βλέπει όλους τους λογαριασμούς χρηστών που αντιστοιχούν σε υπαλλήλους του οργανισμού που εργάζεται. ▪ Να ενημερώνει στοιχεία λογαριασμών που αντιστοιχούν σε υπαλλήλους του οργανισμού. ▪ Να απενεργοποιεί λογαριασμούς υπαλλήλων του οργανισμού.
Σημειώσεις	
1. Η διαγραφή λογαριασμών γενικώς σε επίπεδο οργανισμού είναι μία επικίνδυνη ενέργεια που μπορεί να συντελέσει σε απώλεια δεδομένων σχετικά με τη γενικότερη δραστηριότητα του οργανισμού που συσχετιζόταν με αυτούς τους χρήστες. Οπότε θα προτιμηθεί να απενεργοποιείται απλώς ο λογαριασμός.	

Ονομασία:	Υπάλληλος Οργανισμού (Organization Employee)
Σύντομη Περιγραφή:	Ο/Η Υπάλληλος Οργανισμού ελέγχει και εγκρίνει/ζητάει αλλαγές/απορρίπτει τις αιτήσεις για ραντεβού που δημιουργούν οι πολίτες.
Use Case Diagram:	UC 4
Λειτουργικές Προδιαγραφές	
Διαχείριση καταλόγου αιτήσεων ραντεβού των πολιτών για τον οργανισμό που εργάζεται ο/η υπάλληλος.	Ο/Η υπάλληλος ενός οργανισμού μπορεί: <ul style="list-style-type: none"> • Να βλέπει όλες τις αιτήσεις των πολιτών για τον οργανισμό στον οποίο εργάζεται. • Να εγκρίνει αιτήσεις, αν αυτές πληρούν τα κριτήρια που θέτει ο οργανισμός. • Να ζητάει αλλαγές στην αίτηση ώστε να μπορεί μετέπειτα να την εγκρίνει, όπου είναι εφικτό. • Να ενημερώνει την κατάσταση μιας αίτησης ως ολοκληρωμένη, αν το ραντεβού τελικά πραγματοποιήθηκε επιτυχώς. • Να απορρίπτει οριστικά αιτήσεις αν αυτές δεν ικανοποιούν τα κριτήρια του οργανισμού και δεν είναι εφικτή κάποια διορθωτική ενέργεια.

Εξωτερικό σύστημα

Ονομασία:	Πολίτης (Citizen)
Σύντομη Περιγραφή:	Ο/Η πολίτης μπορεί να στείλει αιτήσεις για να κλείσει ραντεβού με κάποιον οργανισμό.
Use Case Diagram:	UC 5
Περιγραφή Επιχειρησιακής Λειτουργίας Της Οντότητας	
Διαχείριση ραντεβού	<p>Ο/Η πολίτης μπορεί:</p> <ul style="list-style-type: none">• Να βλέπει το ιστορικό των αιτήσεών του.• Να δημιουργεί καινούριες αιτήσεις προς έναν από τους οργανισμούς που είναι καταχωρημένοι στο σύστημα.• Να τροποποιεί αιτήσεις όταν αυτό ζητείται από τον υπάλληλο του οργανισμού.• Να ακυρώνει ραντεβού, αν τελικά δεν μπορεί να παραστεί ενώ έχει εγκριθεί από τον οργανισμό.

Μοντέλο Δεδομένων

Εισαγωγή

Το ΠΣ ΕΡΜΗΣ έχει 3 κύριους τύπους οντοτήτων:

- Τους χρήστες (Users)
- Τους οργανισμούς (Organizations)
- Τις αιτήσεις (Applications)

Για τους χρήστες και τους οργανισμούς δεν απαιτείται η δημιουργία σχεσιακού μοντέλου, καθώς οι απαιτήσεις καλύπτονται από το Keycloak. Ένα τμήμα του σχεσιακού μοντέλου του Keycloak περιγράφεται αποσπασματικά σε παρακάτω ενότητες. Το ER διάγραμμα που δείχνει αυτό το μέρος του σχεσιακού μοντέλου του Keycloak βρίσκεται στο [παράρτημα, στην ενότητα των ERs](#).

Σχεσιακό Μοντέλο Του Keycloak

Χρήστες (Users)

Το Keycloak έχει έναν βασικό πίνακα **USER_ENTITY**, ο οποίος περιλαμβάνει κάποια βασικά γνωρίσματα για τον χρήστη όπως:

- **ID (PK)**
- **E-mail**
- **Όνομα**
- **Επώνυμο**
- **Username (μοναδικό)**
- **Enabled** (Boolean τιμή που υποδηλώνει αν ο λογαριασμός είναι ενεργοποιημένος)

Μέσω του πίνακα **CREDENTIAL** το Keycloak αντιστοιχεί χρήστες με στοιχεία πρόσβασης (password κτλ.).

Μέσω του πίνακα **USER_ATTRIBUTE**, το Keycloak επιτρέπει την καταχώρηση οποιουδήποτε επιπλέον γνωρίσματος επιθυμούμε για έναν χρήστη.

Γκρουπ (Groups)

Τα Groups μας επιτρέπουν να ομαδοποιούμε χρήστες και να τους αναθέτουμε αυτόματα κάποια χαρακτηριστικά όπως ρόλους ή και γνωρίσματα. Το ιεραρχικό στυλ των Groups, το καθιστούν ιδανικό για τη μοντελοποίηση των καταλόγων χρηστών.

Το Keycloak έχει τον πίνακα **KEYCLOAK_GROUP**, ο οποίος περιέχει μεταξύ των γνωρισμάτων του:

- **ID (PK)**
- **Όνομα**
- **ID γονέα** (αν είναι subgroup κάποιου άλλου group)

Μέσω του πίνακα **GROUP_ATTRIBUTE**, το Keycloak επιτρέπει την καταχώρηση οποιουδήποτε επιπλέον γνωρίσματος επιθυμούμε για ένα Group.

Μέσω του πίνακα **USER_GROUP_MEMBERSHIP**, το Keycloak αντιστοιχεί τους χρήστες με τα Groups στα οποία ανήκουν.

Ρόλοι (Roles)

Η αντιστοίχιση των χρηστών/group με ρόλους γίνεται μέσω του Keycloak.

Το Keycloak έχει τον πίνακα **KEYCLOAK_ROLE**, ο οποίος περιέχει μεταξύ των γνωρισμάτων του:

- **ID (PK)**
- **Όνομα**

Μέσω του πίνακα **ROLE_ATTRIBUTE**, μπορούν να καταχωρηθούν γνωρίσματα σε έναν ρόλο, τα οποία κληρονομεί οποιαδήποτε οντότητα στην οποία ανατίθεται αυτός ο ρόλος.

Η αντιστοίχιση των ρόλων με τους χρήστες γίνεται μέσω του πίνακα **USER_ROLE_MAPPING**.

Αντίστοιχα, η αντιστοίχιση των ρόλων με τα Group γίνεται μέσω του πίνακα **USER_GROUP_MEMBERSHIP**.

Χρήστες

Οι οντότητες των χρηστών είναι διαχειρίσιμες από το Keycloak. Αυτό είναι υπεύθυνο για τη διαχείριση όλων των καταλόγων χρηστών του πληροφοριακού συστήματος. Τα services (Back-Ends, Front-Ends κτλ.) που θα υλοποιηθούν από την ομάδα ανάπτυξης του συστήματος μπορούν να δημιουργούν/ζητούν/ενημερώνουν/απενεργοποιούν/διαγράφουν χρήστες, διατάζοντας τον Keycloak μέσω του [Admin REST API](#) που προσφέρει.

Γνωρίσματα του χρήστη

Κάθε τύπος χρήστη θα περιέχει:

- **ID (PK)**
- **E-mail**
- **Όνομα**
- **Επώνυμο**
- **Username (μοναδικό)**
- **Enabled** (Boolean τιμή που υποδηλώνει αν ο λογαριασμός είναι ενεργοποιημένος)
- Τη **διεύθυνση** (όπως ορίζεται από το address claim του OpenID Connect):
 - **Street**
 - **Locality**
 - **Region**
 - **Postal Code**
 - **Country**
 - **Formatted** (Πως να εμφανίζονται μαζί ωραία οι πληροφορίες)
- Το **φύλο** (Male/Female/Other)
- Την **ημερομηνία γέννησης**
- Έναν **αριθμό τηλεφώνου**

Χρήστες και Groups

Όλοι οι τύποι χρηστών του πληροφοριακού συστήματος (εκτός των Διαχειριστών Συστήματος) θα τοποθετηθούν σε κάποιο Group. Ο τρόπος τοποθέτησης των προϋσταμένων/υπάλληλων του κάθε οργανισμού περιγράφεται στην ενότητα [Ιεραρχία των Group για τους οργανισμούς](#). Εδώ τεκμηριώνονται οι υπόλοιποι τύποι χρηστών.

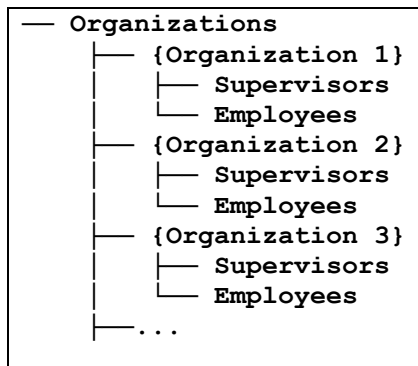
- Οι **Διαχειριστές/στριες Οργανισμών** θα τοποθετηθούν σε ένα Group **ORGANIZATIONS_ADMINS**. Σε αυτό το Group θα έχει ανατεθεί ο ρόλος **ROLE_ORGS_ADMIN**.
- Οι πολίτες θα τοποθετηθούν σε ένα Group **CITIZENS**. Σε αυτό το Group θα έχει ανατεθεί ο ρόλος **ROLE_CITIZEN**.

Οργανισμοί

Οι οργανισμοί μοντελοποιούνται απλώς ως [Groups](#) του Keycloak.

Ιεραρχία των Group για τους οργανισμούς

Η ιεραρχία των Group ως προς τους οργανισμούς είναι η εξής:



- Σε κάθε group Supervisors ενός οργανισμού θα έχει ανατεθεί ο ρόλος **ROLE_ORG_SUPERVISOR**. Σε αυτά τα group θα τοποθετούνται οι προϊστάμενοι/ες κάθε οργανισμού.
- Σε κάθε group Employees ενός οργανισμού θα έχει ανατεθεί ο ρόλος **ROLE_ORG_EMPLOYEE**. Σε αυτά τα group θα τοποθετούνται οι υπάλληλοι κάθε οργανισμού.

Γνωρίσματα του οργανισμού

Οι οργανισμοί αποτελούνται από τα εξής γνωρίσματα:

- **ID** (απόλυτη αντιστοιχία με το ID του Group που αναπαριστά τον οργανισμό)
- **Όνομα** (απόλυτη αντιστοιχία με το όνομα του Group που αναπαριστά τον οργανισμό)

Αιτήσεις

Η οντότητα της αίτησης πρέπει να υλοποιηθεί στο σχεσιακό μοντέλο του Back-End.

Γνωρίσματα της αίτησης

- **ID** (PK)
- **ID πολίτη** που κάνει την αίτηση
- **ID οργανισμού** στον οποίο καταθέτεται η αίτηση
- **Κατάσταση της αίτησης** (περιγράφεται παρακάτω)
- **Λεπτομέρειες σχετικά με την αίτηση** (σχόλια πολίτη – απλά ένα String)
- **Timestamp δημιουργίας αίτησης**
- **Ημερομηνία και ώρα που επιθυμεί ο πολίτης να γίνει η συνάντηση**

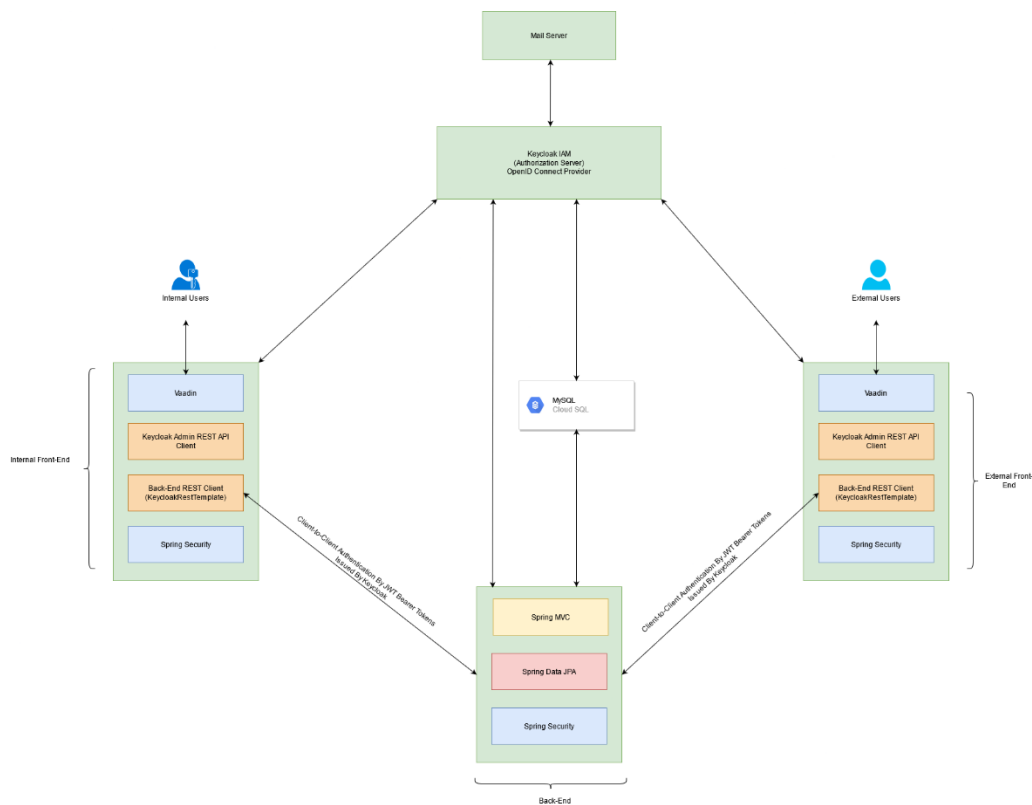
Καταστάσεις της αίτησης

Μια αίτηση έχει 6 καταστάσεις στην οποία μπορεί να βρίσκεται:

- **SUBMITTED:** Η αίτηση έχει καταχωρηθεί επιτυχώς από τον πολίτη και αναμένεται η απάντηση του υπαλλήλου.
- **RESUBMISSION_REQUIRED:** Ο υπάλληλος του οργανισμού διαπιστώνει ότι λεπτομέρειες της αίτησης πρέπει να τροποποιηθούν ώστε να προχωρήσει η πορεία της.
- **APPROVED:** Ο υπάλληλος εγκρίνει την αίτηση και το ραντεβού κλείνεται για τη συμφωνημένη ώρα και ημερομηνία.
- **COMPLETED:** Το ραντεβού ολοκληρώθηκε επιτυχώς, οπότε ο υπάλληλος επισημαίνει την αίτηση ως ολοκληρωμένη.
- **CANCELED:** Ο πολίτης ακυρώνει το ραντεβού, παρόλο που έχει εγκριθεί από τον οργανισμό (πριν την ημερομηνία που έχει οριστεί το ραντεβού).
- **REJECTED:** Ο υπάλληλος απορρίπτει οριστικά την αίτηση.

Ένα State Machine Diagram που περιγράφει τις μεταβάσεις καταστάσεων μιας αίτησης υπάρχει στο [Παράρτημα](#).

High Level Design



Υποσυστήματα του ΠΣ

Το πληροφοριακό σύστημα ΕΡΜΗΣ αποτελείται από:

- Τον Authorization Server **Keycloak**
 - Έκδοση: 14.0.0
- Το RDBMS **MySQL**
 - Έκδοση 5.7
- Ένα **Back-End Service**, το οποίο αναλαμβάνει τη διαχείριση των αιτήσεων των οργανισμών.
 - **Συνοπτικό Tech Stack:**
 - **Spring Boot 2.4.5**
 - **Spring Data JPA**
 - **Spring Web**
 - **Spring Security**
- Ένα **Front-End Service**, το οποίο αναλαμβάνει να εξυπηρετεί τους εσωτερικούς χρήστες του πληροφοριακού συστήματος
 - **Συνοπτικό Tech Stack:**
 - **Spring Boot 2.4.5**
 - **Spring Security**
 - **Keycloak Admin Client**
 - **Vaadin 14.4.10 (LTS)**

- Ένα **Front-End Service**, το οποίο αναλαμβάνει να εξυπηρετεί τους εξωτερικούς χρήστες του πληροφοριακού συστήματος
 - **Συνοπτικό Tech Stack:**
 - **Spring Boot 2.4.5**
 - **Spring Security**
 - **Vaadin 14.4.10 (LTS)**
- Ένας **Mail Server**, ο οποίος αναλαμβάνει να στέλνει μηνύματα ηλεκτρονικού ταχυδρομείου στους χρήστες της εφαρμογής. Προς το παρόν, η εφαρμογή θα έχει Mail Server μόνο για σκοπούς δοκιμών.
 - **Συνοπτικό Tech Stack:**
 - **Mailhog (Mock Mail Server)**

Συνοπτική Περιγραφή Αρχιτεκτονικής

Internal Front-End

Το Internal Front-End θα εξυπηρετεί τους εξής τύπους χρηστών:

- Διαχειριστές/στριες Οργανισμών
- Προϊσταμένους/ες Οργανισμού
- Υπαλλήλους Οργανισμού

Ως Front-End τεχνολογία θα χρησιμοποιηθεί το Vaadin 14 (LTS) Framework. Μέσω της επίσημης [Admin Client](#) βιβλιοθήκης του Keycloak (με κάποια customizations), η server-side μεριά της Vaadin εφαρμογής θα επικοινωνεί κατευθείαν με τον Authorization Server έτσι ώστε να είναι εφικτή η διαχείριση των καταλόγων χρηστών/οργανισμών για λογαριασμό της αυθεντικοποιημένης οντότητας. Η επικοινωνία με το Back-End για να έχουν οι υπάλληλοι οργανισμού πρόσβαση στις αιτήσεις του οργανισμού τους θα γίνεται μέσω του [KeycloakRestTemplate](#).

External Front-End

Το External Front-End θα εξυπηρετεί τους εξής τύπους χρηστών:

- Πολίτες

Ως Front-End τεχνολογία θα χρησιμοποιηθεί το Vaadin 14 (LTS) Framework. Η επικοινωνία με το Back-End για να μπορούν οι πολίτες να καταχωρήσουν τις αιτήσεις τους θα γίνεται μέσω του [KeycloakRestTemplate](#). Η εφαρμογή θα επικοινωνεί επίσης με το Keycloak για να λαμβάνει τους διαθέσιμους οργανισμούς στους οποίους ο πολίτης μπορεί να κάνει την αίτηση.

Back-End

Το Back-End θα χρησιμοποιείται για τη διαχείριση των αιτήσεων του κάθε οργανισμού. Το REST API του θα υποστηρίζει pagination μηχανισμούς που θα δένουν κατάλληλα με τους DataProviders των Vaadin εφαρμογών. Θα χρησιμοποιηθεί Spring Data JPA ως DAO layer και θα επικοινωνεί με την MySQL βάση έχοντας δικό του σχήμα σε αυτήν.

Keycloak

Μέσω των SPIs (Service Provider Interfaces), είναι δυνατή η ανακάλυψη και η ενσωμάτωση custom προσθέτων, τα οποία προσδίδουν έξτρα λειτουργικότητα στον server. Συγκεκριμένα, αυτή η δυνατότητα θα χρησιμοποιηθεί για την ανάπτυξη [Custom REST Endpoints](#) ώστε να στηριχθούν τα use cases της εφαρμογής και θα υλοποιηθούν και οι απαραίτητοι Mappers, ώστε να ενσωματωθούν στο Access Token οι απαιτούμενες έξτρα πληροφορίες.

Για το deployment του Keycloak θα χρησιμοποιηθεί ένα Dockerfile το οποίο θα χτίζει μία εικόνα που θα περιέχει όλες τις ειδικές τροποποιήσεις που χρειάζεται το Keycloak για να ανταποκριθεί στις ανάγκες αυτού του πληροφοριακού συστήματος. Θα χρησιμοποιεί την MySQL βάση για την αποθήκευση των δεδομένων του.

Mail Server (Mailhog)

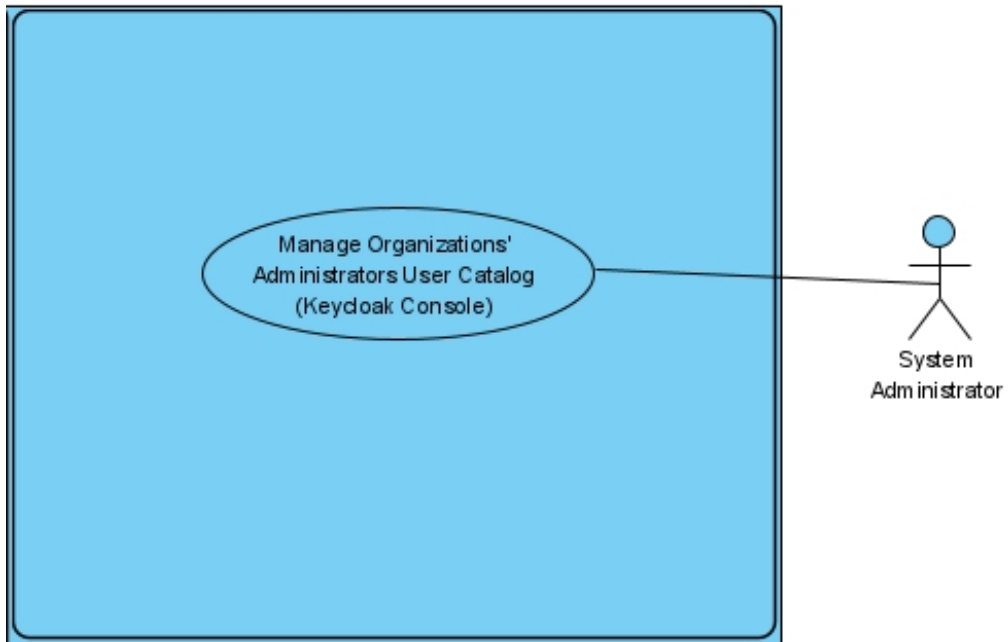
Το Keycloak θα χρησιμοποιεί το [Mailhog](#) ως SMTP Server και θα στέλνει μηνύματα ηλεκτρονικού ταχυδρομείου σχετικά με αιτήματα αλλαγής κωδικού, επιβεβαιώσεις λογαριασμών κ.α. Ο χρήστης που δοκιμάζει την εφαρμογή θα μπορεί μέσω του UI του Mailhog να βλέπει πως θα έμοιαζε το μήνυμα αν στελνόταν πραγματικά στον χρήστη.

Παράρτημα

Use Case διαγράμματα

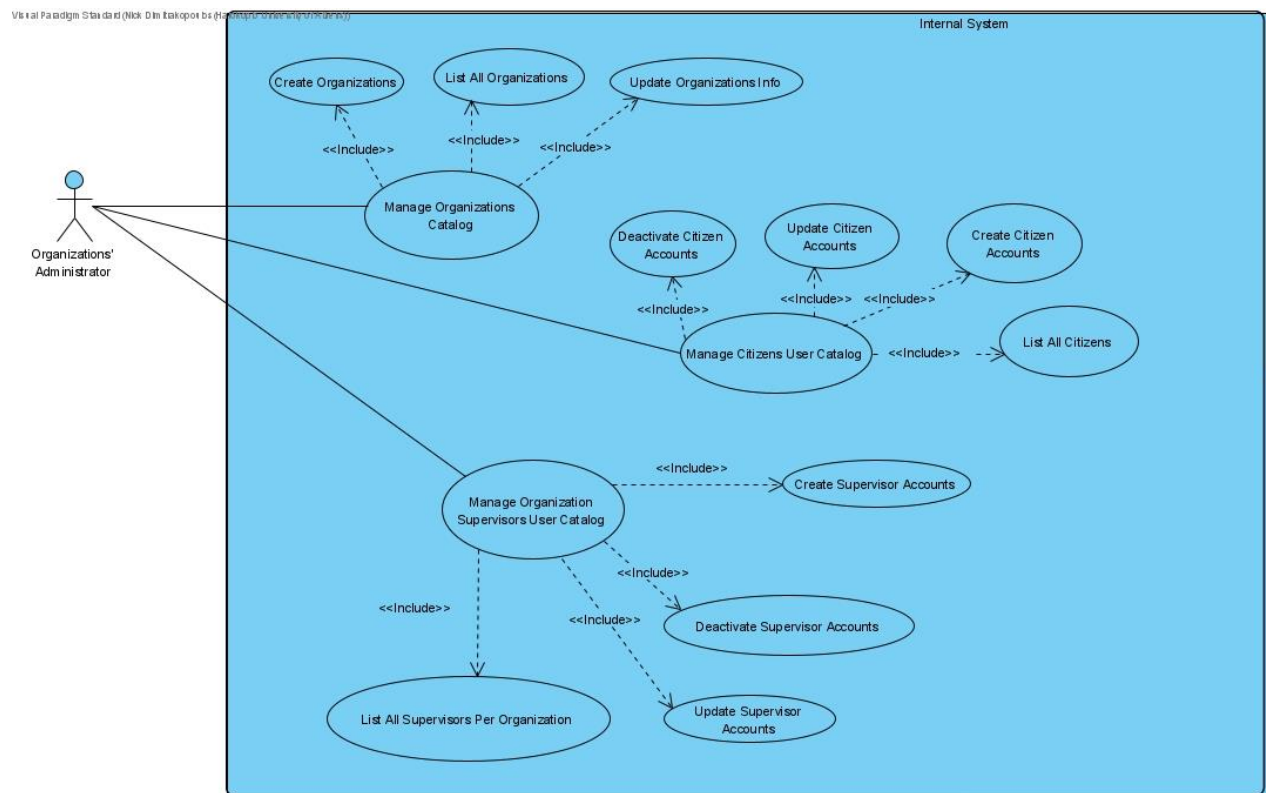
Εσωτερικό σύστημα

UC 1 - Διαχειριστής συστήματος



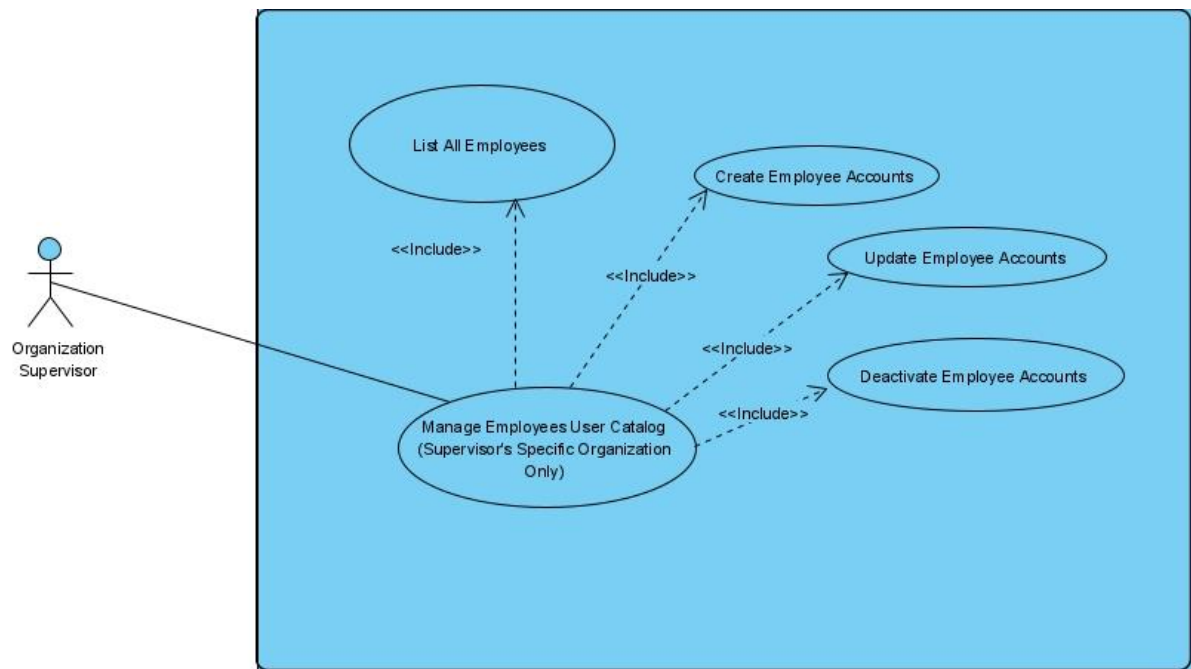
UC 1 Μέρος του Use Case διαγράμματος του εσωτερικού συστήματος που δείχνει τα Use Cases του διαχειριστή συστήματος

UC 2 - Διαχειριστές/στρια οργανισμών



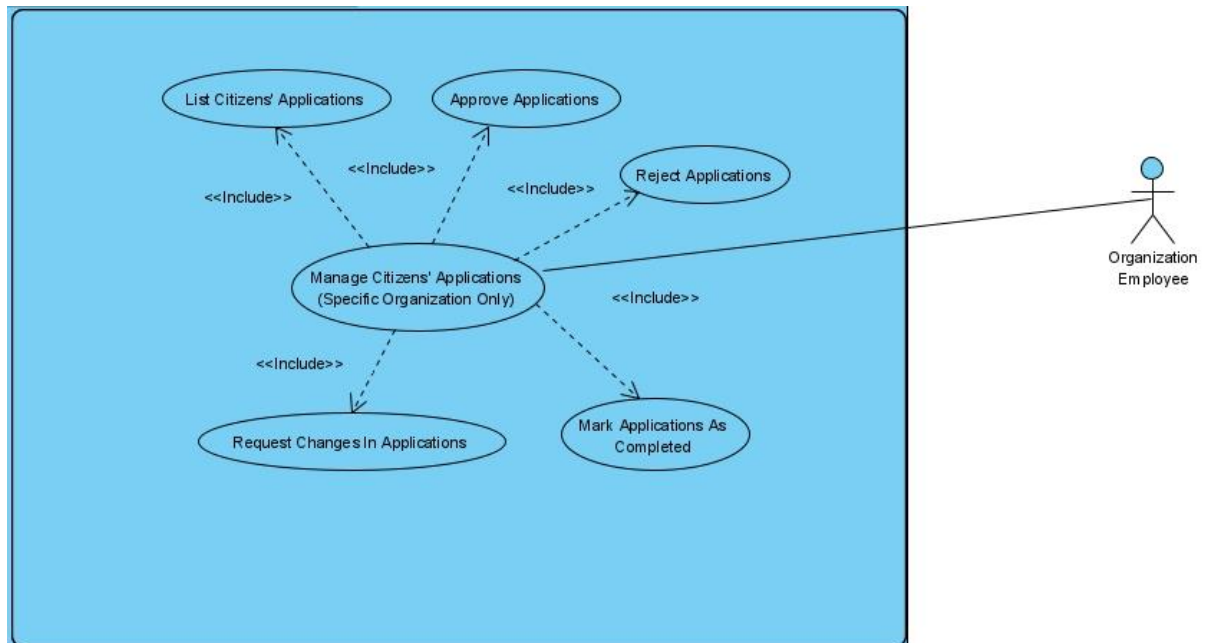
UC 2 Μέρος του Use Case διαγράμματος του εσωτερικού συστήματος που δείχνει τα Use Cases των διαχειριστών/στριων οργανισμών

UC 3 - Προϊστάμενος/η οργανισμού



UC 3 Μέρος του Use Case διαγράμματος του εσωτερικού συστήματος που δείχνει τα Use Cases των προϊσταμένων οργανισμών

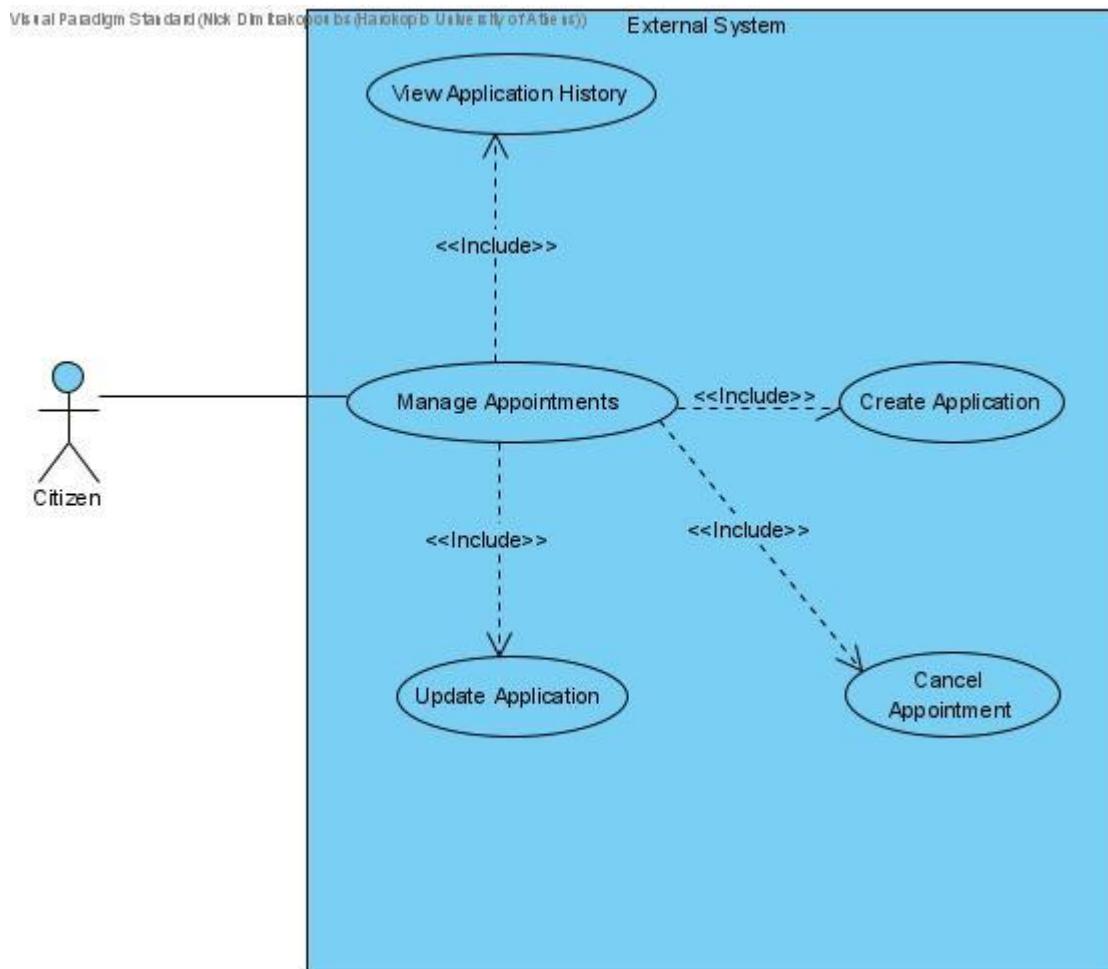
UC 4 - Υπάλληλος οργανισμού



UC 4 Μέρος του Use Case διαγράμματος του εσωτερικού συστήματος που δείχνει τα Use Cases των υπαλλήλων οργανισμών

Εξωτερικό Σύστημα

UC 5 - Πολίτης



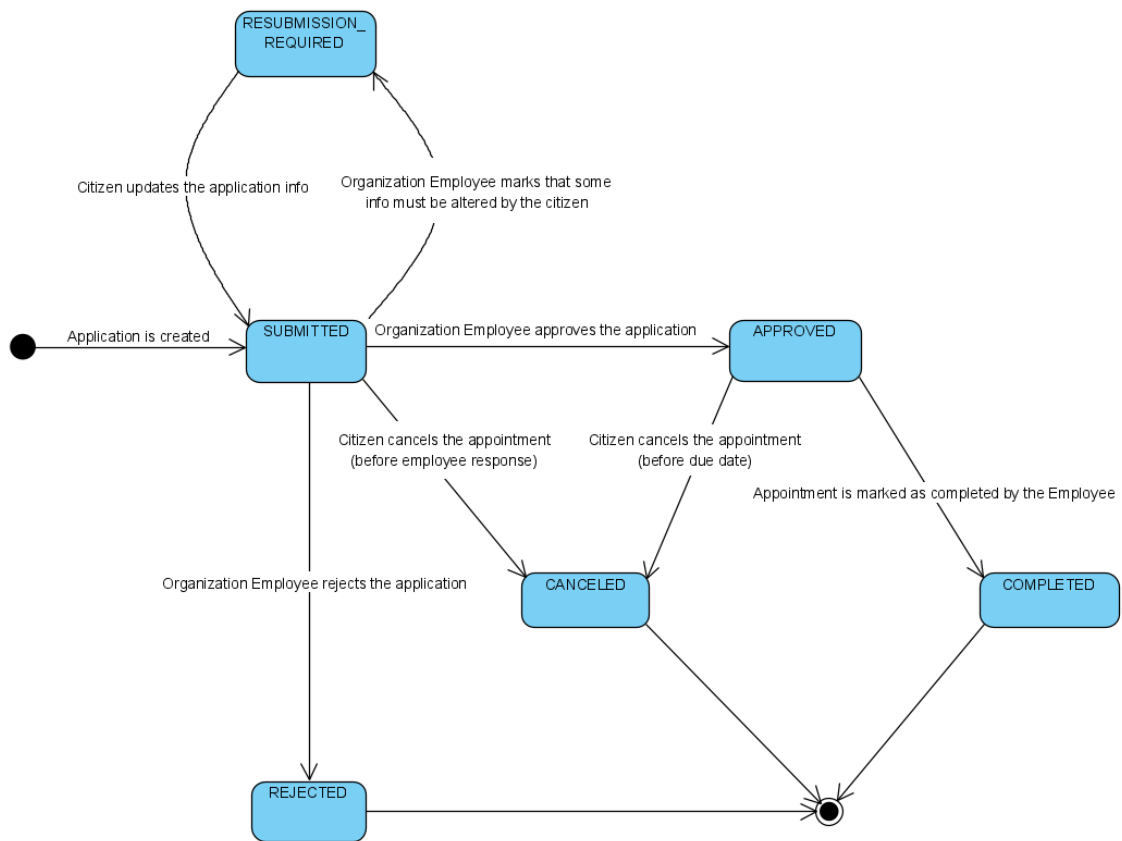
UC 5 Το πλήρες Use Case διάγραμμα του εξωτερικού συστήματος

ER 1 - ER διάγραμμα του Keycloak



State Machine Διαγράμματα

SM 1 - Καταστάσεις Αιτήσεων

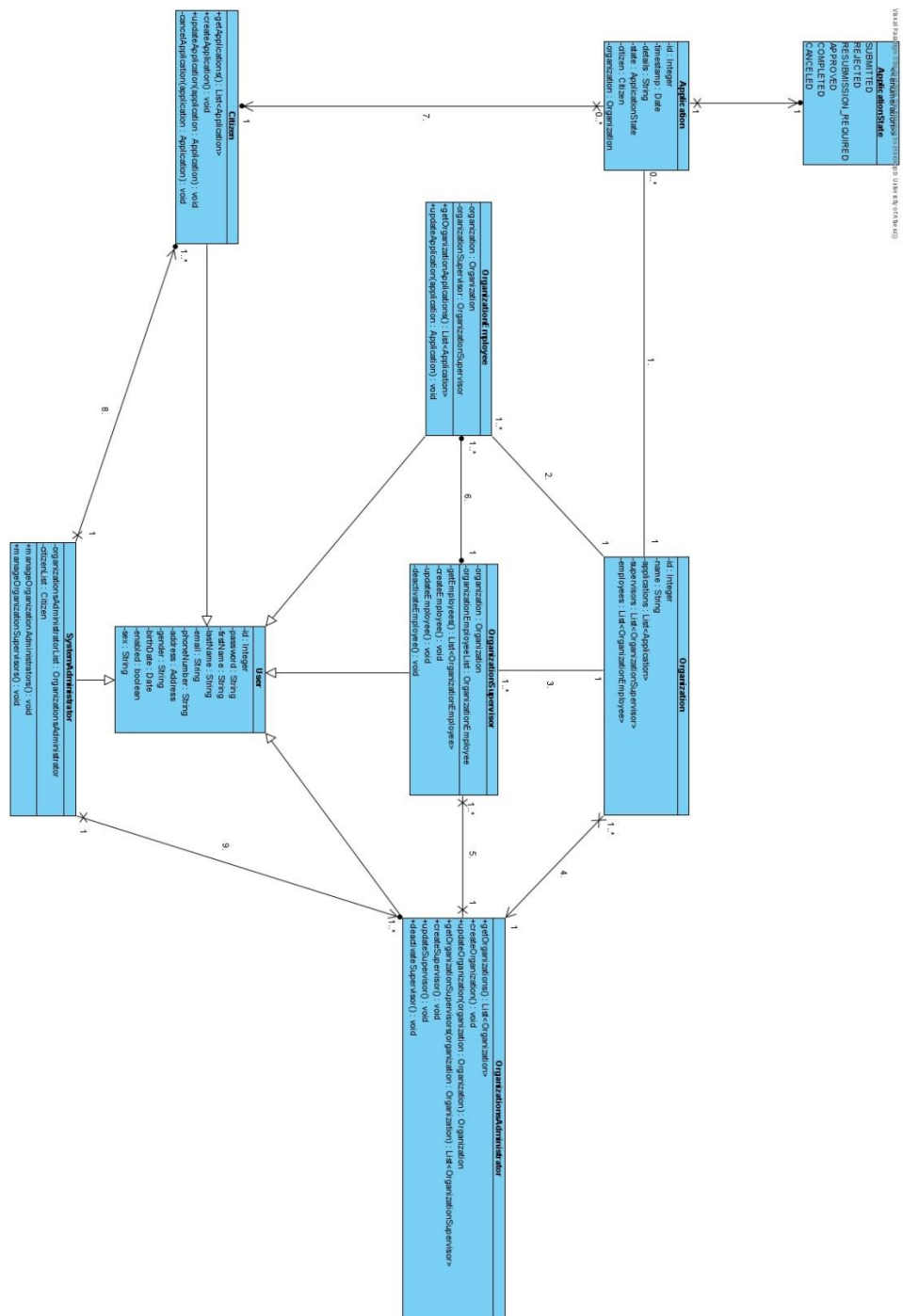


SM 1 Διάγραμμα πεπερασμένων καταστάσεων μηχανής για τις πιθανές καταστάσεις στις οποίες μπορεί να βρίσκεται μία αίτηση

Class Διαγράμματα

CD 1 – Αφηρημένο Class Diagram του συστήματος

ΠΡΟΣΟΧΗ: Το παρακάτω διάγραμμα αναπαριστά μια αφηρημένη προσέγγιση στον τρόπο που αλληλοεπιδρούν οι οντότητες του συστήματος και δεν πρέπει να ληφθεί σοβαρά υπόψιν για την πραγματική κατανόηση του συστήματος.



CD 1 Class διάγραμμα του συστήματος