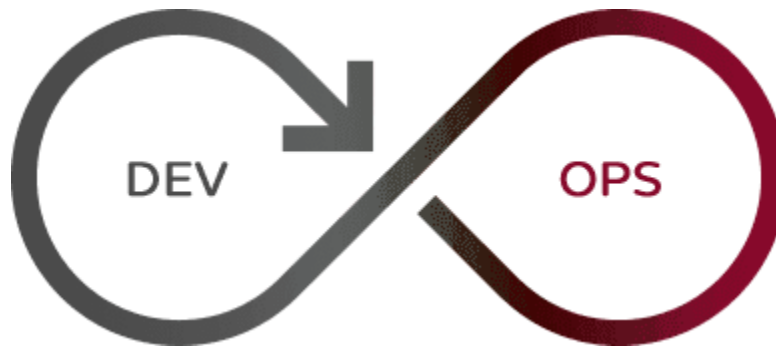




ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΜΑΤΙΚΗΣ



## Ομάδα 35

Άθως Εξάρχου - 2022134

Χρήστος Καλαματιανός - 22150

Σύστημα Διαχείρισης Ενοικίασης Ακινήτων

Εργασία στο Μάθημα «Βασικές έννοιες και εργαλεία DevOps»

Περιεχόμενα	
Ανάλυση και Σχεδίαση	3
Παραδοχές	3
Ανάλυση	4
Σχεδίαση	7
Τεχνολογίες / Εργαλεία	8
Αρχιτεκτονική Εφαρμογής	9
Σενάρια χρήσης	10
Deployment	15
Ansible	15
Ansible – Docker	18
Kubernetes	18
CI/CD	19
Γενικά Σχόλια/Παρατηρήσεις	19
Μας δυσκόλεψε / δεν υλοποιήσαμε	20
Κώδικας	20
Αποθετήρια κώδικα	20
Δοκιμαστικά accounts και urls	20
Url δοκιμαστικού περιβάλλοντος	21
Οδηγίες Χρήσης / Εγκατάστασης	21

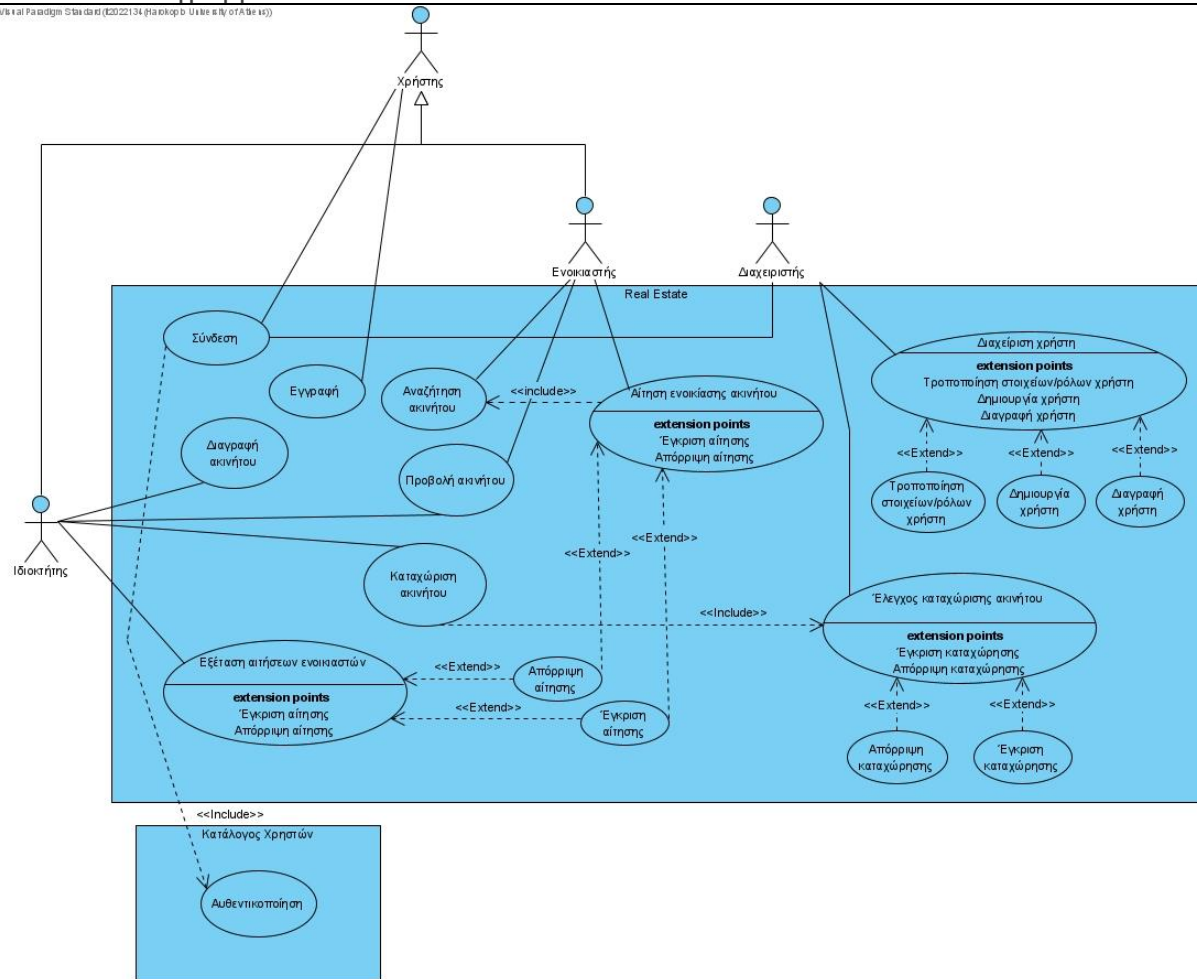
# Ανάλυση και Σχεδίαση

## Παραδοχές

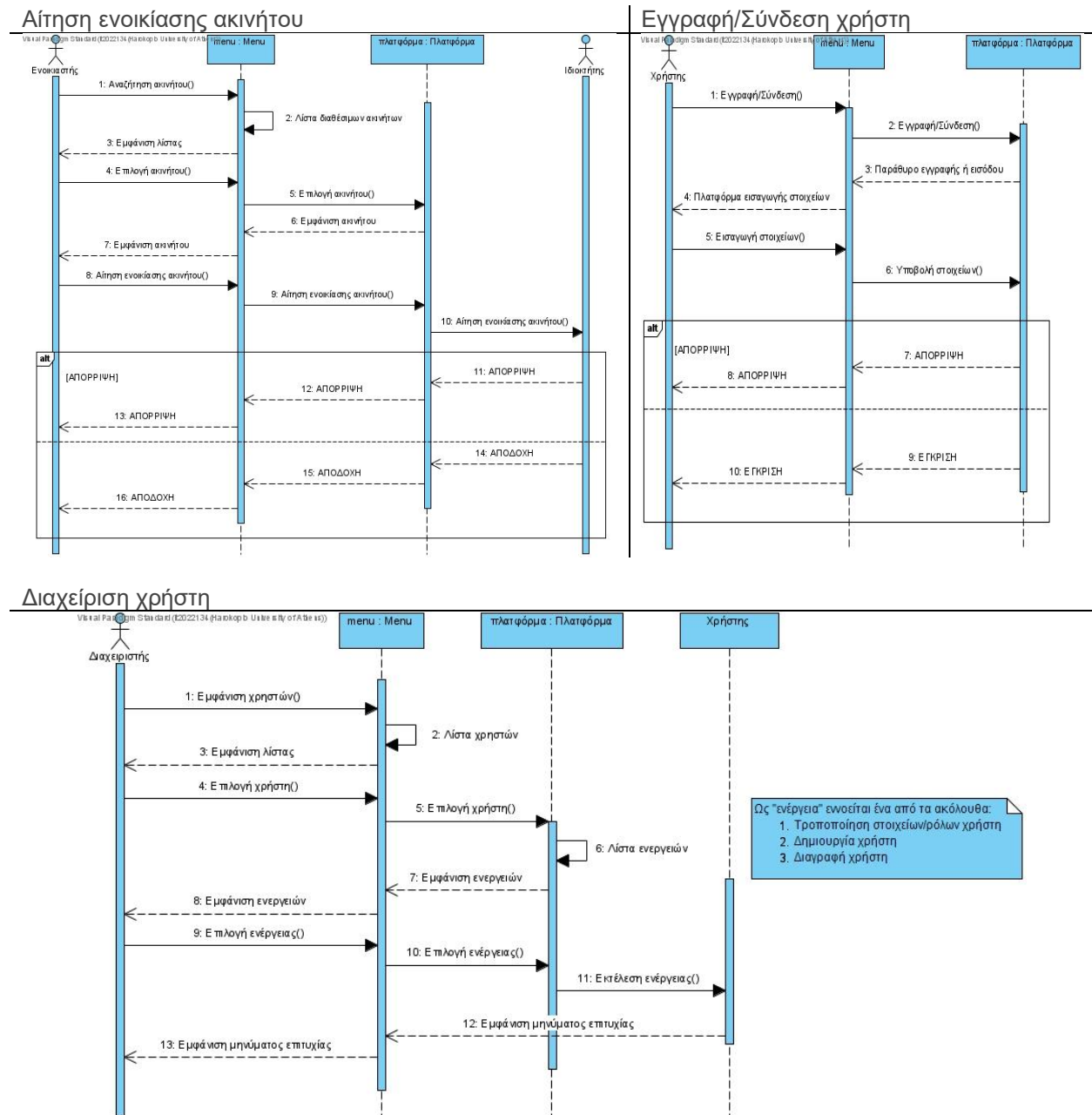
- Ως ακίνητο εννοείται ένα διαμέρισμα.
- Ως χρήστης εννοείται ο Ιδιοκτήτης ή ο Ενοικιαστής, όχι ο Διαχειριστής.
- Θεωρείται ότι για την αξιοποίηση των λειτουργιών της εφαρμογής, ο χρήστης είναι ήδη συνδεδεμένος στο σύστημα.
- Όταν ο ενοικιαστής κάνει “Αίτηση ενοικίασης ακινήτου” και ο Ιδιοκτήτης του ακινήτου αποδεχτεί την ενοικίαση αυτή, θεωρείται ότι η πληρωμή πραγματοποιείται σε δεύτερο χρόνο και όχι in-app.
- Ένα ακίνητο έχει ένα μόνο ιδιοκτήτη.
- Ο Ιδιοκτήτης έχει τη δυνατότητα να διαγράψει ακίνητα που έχει καταχωρήσει, με εξαίρεση τα ακίνητα που βρίσκονται υπό ενοικίαση.
- Ένας χρήστης για να εγγραφεί στο σύστημα πρέπει να εισάγει username, email και password.
- Ένας χρήστης λαμβάνει το ρόλο του “Ενοικιαστή” όταν ολοκληρώσει για πρώτη φορά τη διαδικασία ενοικίασης ενός ακινήτου.
- Ένας χρήστης λαμβάνει το ρόλο του “Ιδιοκτήτη” όταν καταχωρήσει για πρώτη φορά ένα ακίνητο για ενοικίαση.
- Ένας ενοικιαστής μπορεί να νοικιάζει ένα μόνο διαμέρισμα.
- Ο Διαχειριστής υπάρχει από τη δημιουργία της εφαρμογής, συνεπώς δεν έχει δυνατότητα εγγραφής παρά μόνο σύνδεσης.
- Όσο ένα διαμέρισμα είναι υπό ενοικίαση, η επιλογή ενοικίασης ακινήτου δεν υφίσταται.

## Use Case Διάγραμμα

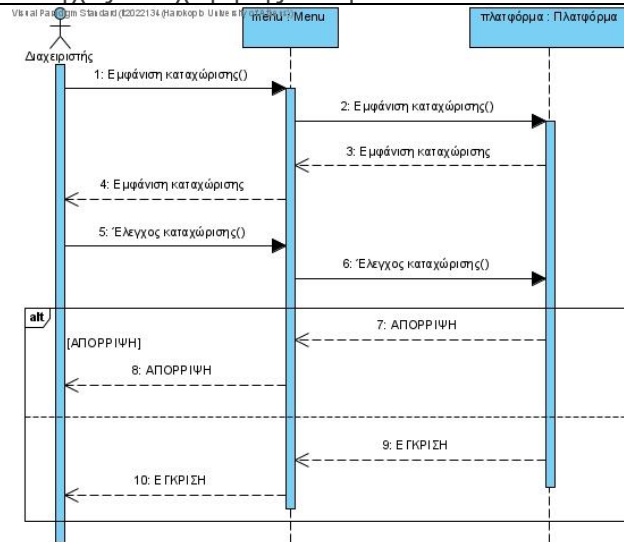
Visual Paradigm Standard (E2022134 (Hankook University of Arts))



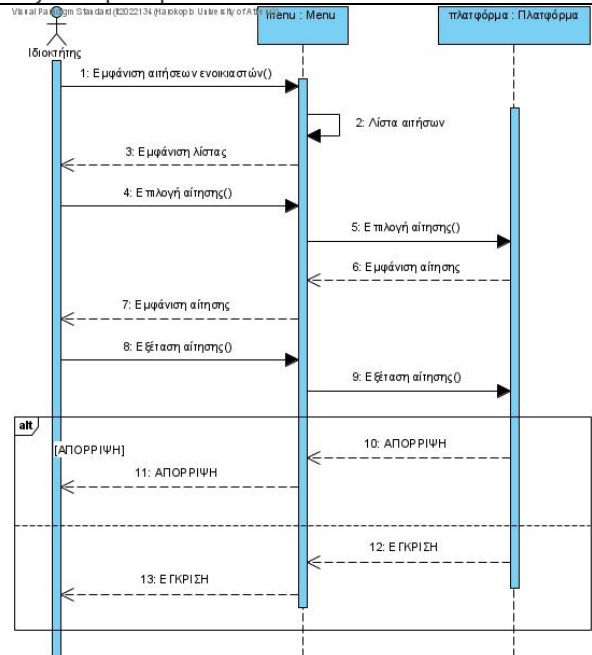
## Sequence διαγράμματα



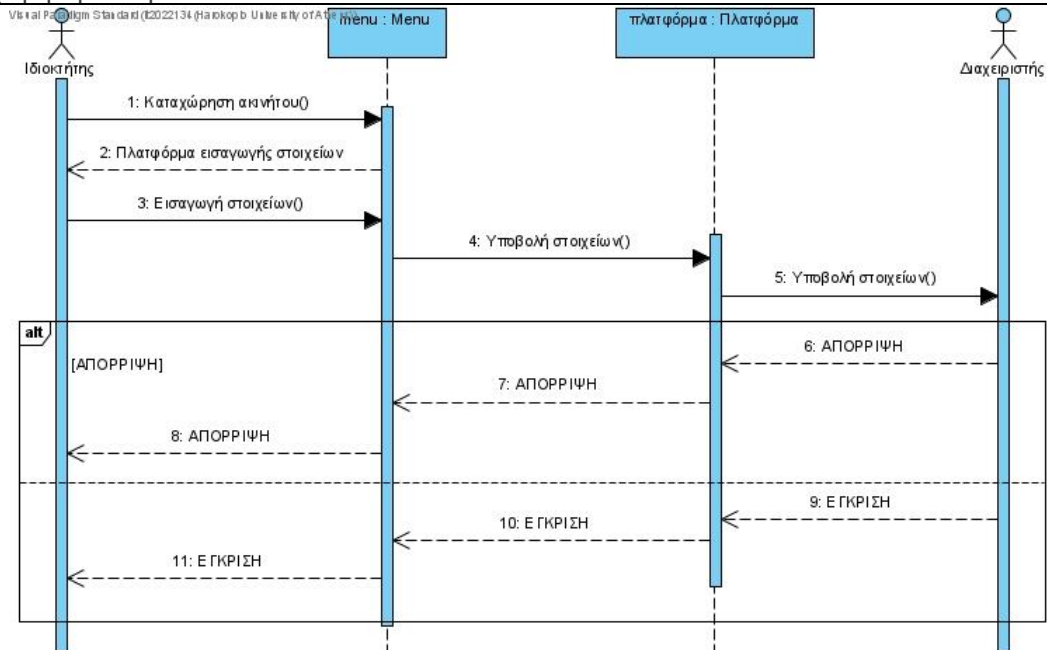
## Έλεγχος καταχώρησης ακινήτου



## Εξέταση αιτήσεων ενοικιαστών



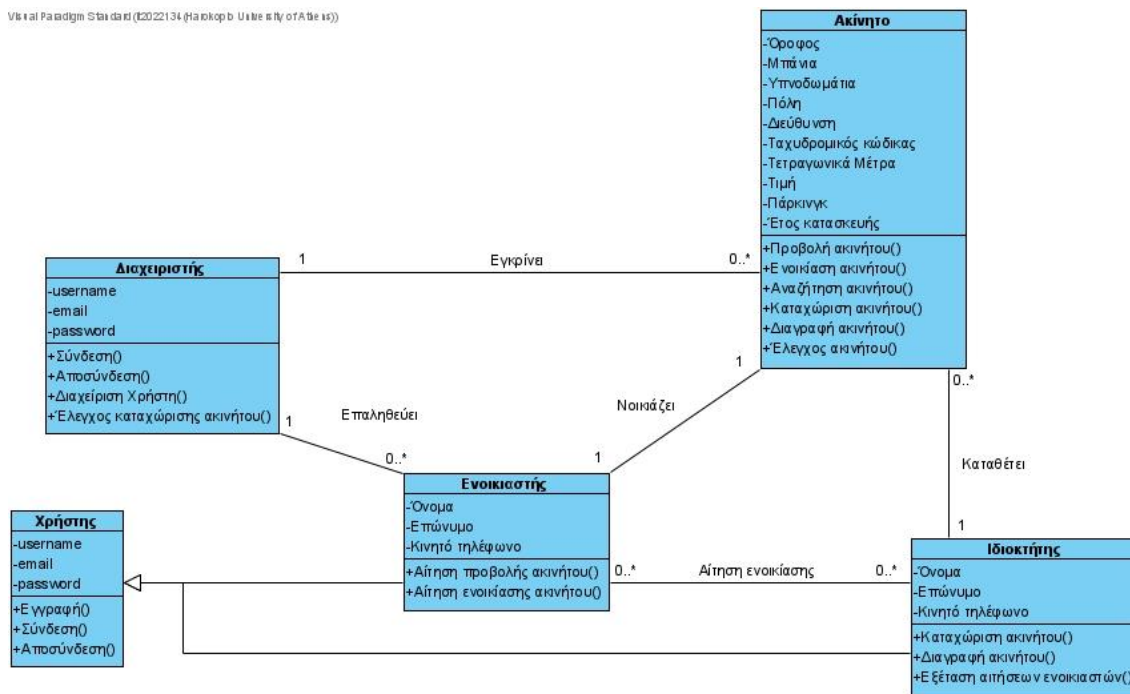
## Καταχώρηση ακινήτου



## Σχεδίαση

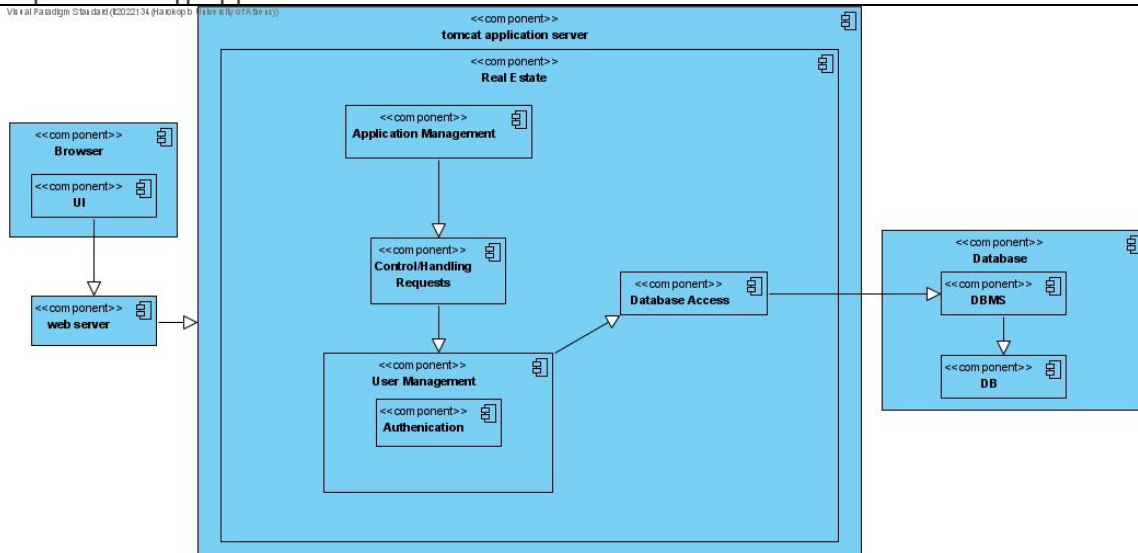
### Class διάγραμμα

Visual Paradigm Standard (©2022) 34 (Hellenic University of Athens)



### Component διάγραμμα

Visual Paradigm Standard (©2022) 34 (Hellenic University of Athens)



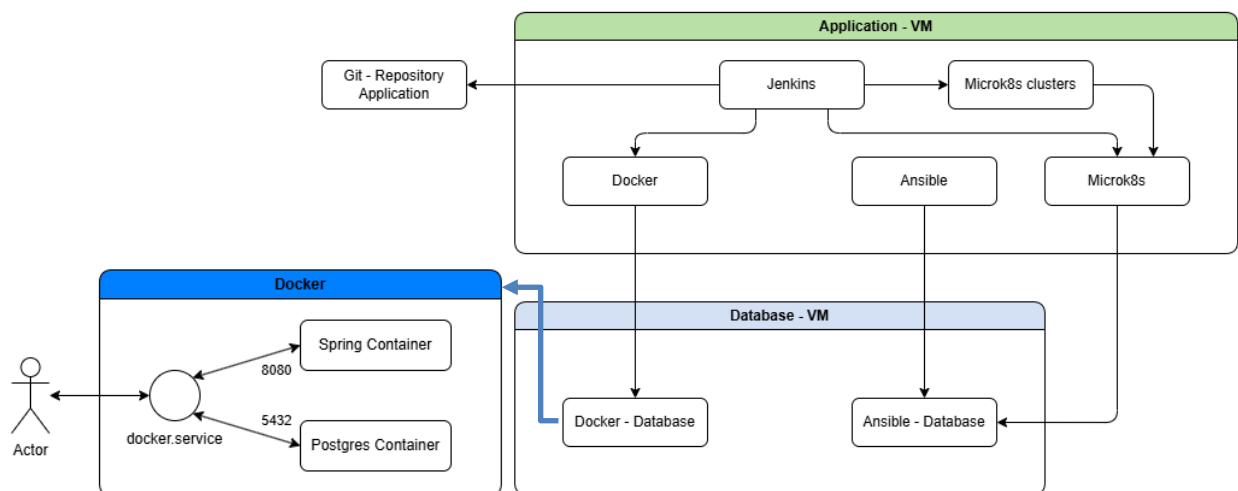
Τεχνολογίες / Εργαλεία

Εργαλείο / Τεχνολογία	Περιγραφή / Component
Spring boot	Βασική εφαρμογή backend (ds-exc-2024)
PostgreSQL	Βάση δεδομένων της εφαρμογής
Docker	Containerization της Spring εφαρμογής
Docker compose	Τοπική ανάπτυξη πολλαπλών services (app + db)
Ansible	Αυτοματοποίηση εγκατάστασης (VMs, Docker, Kubernetes)
Kubernetes	Ανάπτυξη της εφαρμογής και της βάσης σε pods
Nginx	Reverse proxy για την εφαρμογή (μέσω template)
Jenkins	CI/CD pipeline για τη δημιουργία image, test & deploy
Git	Έλεγχος έκδοσης: DevOps repository
Systemd	Εκκίνηση εφαρμογής σε VM μέσω Ansible (spring.service.j2)



# Αρχιτεκτονική Εφαρμογής

- Client → NGINX Ingress  
Ο χρήστης (browser) στέλνει HTTP/HTTPS αιτήματα στο NGINX, που λειτουργεί ως reverse – proxy / Ingress controller στο Kubernetes cluster.
- Routing προς Application Service  
Το Ingress δρομολογεί τα αιτήματα στο κύριο microservice μας (Spring Boot), το οποίο τρέχει σε container (Docker) και ως pod στο Kubernetes μέσω Deployment + Service τύπου ClusterIP.
- Σύστημα Διαχείρισης Δεδομένων  
Το Application Service επικοινωνεί με την PostgreSQL database, τρέχοντας σε ξεχωριστό pod με Persistent VolumeChain για μόνιμη αποθήκευση δεδομένων.
- Email Subsystem μέσω SMTP  
Η εφαρμογή αποστέλλει εξερχόμενα email απευθείας μέσω ενός εξωτερικού SMTP server της Google (smtp.gmail.com), χρησιμοποιώντας αυθεντικοποίηση με App Password και ασφαλή σύνδεση μέσω STARTTLS. Δεν χρησιμοποιείται τοπικός MailHog container ή άλλος ενδιάμεσος MTA, καθώς βασιζόμαστε πλήρως σε εξωτερική υπηρεσία (SaaS) για την αποστολή email.
- CI/CD Pipeline με Jenkins pipeline:
  1. Build: Δημιουργία Docker images
  2. Push: Ανέβασμα εικόνων σε Docker registry
  3. Deploy: Jenkins εφαρμόζει manifests και ενημερώνει (apply) το Kubernetes cluster.



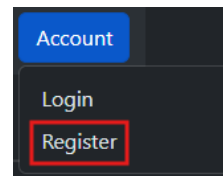
# Σενάρια χρήσης

## Σενάριο 1: Εγγραφή Χρήστη

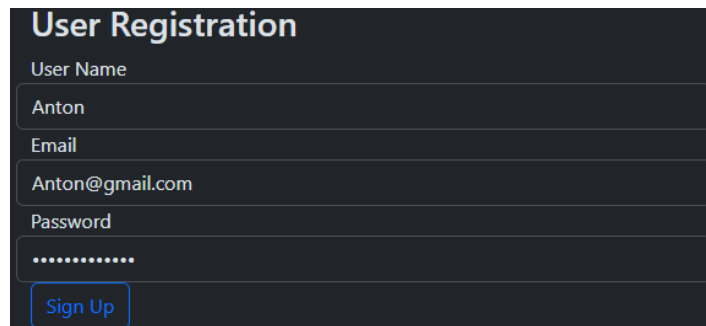
**Στόχος:** Ο χρήστης δημιουργεί νέο λογαριασμό για να αποκτήσει πρόσβαση στις λειτουργίες της εφαρμογής.

### Βήματα:

1. Ο χρήστης μεταβαίνει στη σελίδα <http://realestate.ip-ddns.com:8080/> και πατάει το κουμπί **Register**.
2. Συμπληρώνει:
  - Username (Όνομα χρήστη)
  - Email
  - Password (Κωδικό πρόσβασης)
3. Επιβεβαιώνει πατώντας **Sign up**.



### Screenshot:

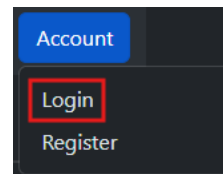
A dark-themed form titled 'User Registration'. It contains three input fields: 'User Name' with the value 'Anton', 'Email' with the value 'Anton@gmail.com', and 'Password' with masked characters '.....'. A blue 'Sign Up' button is at the bottom.

## Σενάριο 2: Σύνδεση Χρήστη

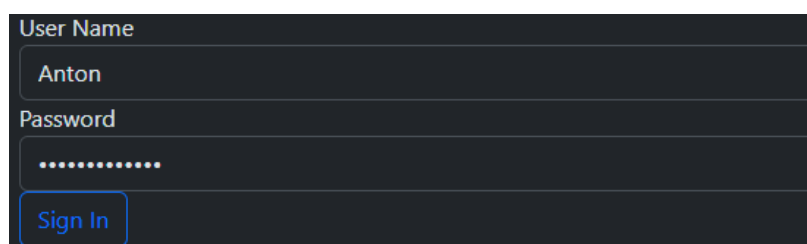
**Στόχος:** Ο χρήστης συνδέεται στο σύστημα με τον λογαριασμό του.

### Βήματα:

1. Πατάει το κουμπί **Login**
2. Εισάγει Username και Password.
3. Πατάει **Sign In**.



### Screenshot:

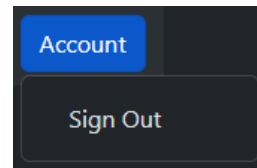
A dark-themed form titled 'User Login'. It contains two input fields: 'User Name' with the value 'Anton' and 'Password' with masked characters '.....'. A blue 'Sign In' button is at the bottom.

### Σενάριο 3: Αποσύνδεση Χρήστη

**Στόχος:** Ο χρήστης τερματίζει τη συνεδρία του με ασφάλεια.

**Βήματα:**

- Πατάει το κουμπί **Sign Out**



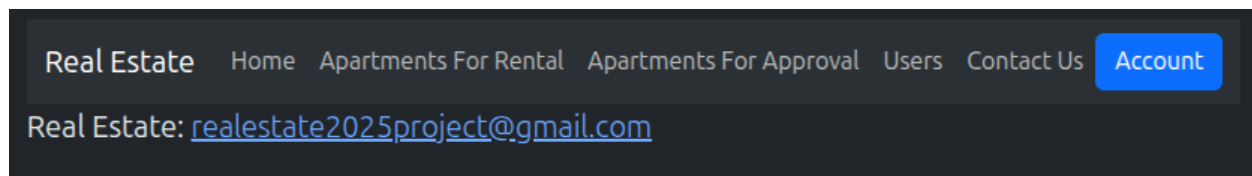
### Σενάριο 4: Λήψη Υποστήριξης

**Στόχος:** Ο χρήστης επικοινωνεί για βοήθεια.

**Βήματα:**

1. Πατάει στο tab **Contact**.
2. Συμπληρώνει φόρμα ή διαβάζει οδηγίες επικοινωνίας.

**Screenshot:** Σελίδα υποστήριξης



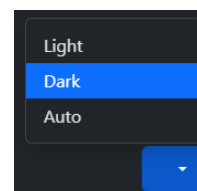
### Σενάριο 5: Αλλαγή Θέματος Εμφάνισης

**Στόχος:** Ο χρήστης αλλάζει την εμφάνιση της εφαρμογής.

**Βήματα:**

1. Πατάει το εικονίδιο εμφάνισης (κάτω δεξιά).
2. Επιλέγει:
  - Φωτεινή λειτουργία (Light Mode): Χρησιμοποιεί ένα ανοιχτόχρωμο θέμα.
  - Σκοτεινή λειτουργία (Dark Mode): Χρησιμοποιεί ένα σκοτεινό θέμα.
  - Αυτόματη λειτουργία (Auto Mode): Αλλάζει δυναμικά μεταξύ φωτεινής και σκοτεινής λειτουργίας, βάση των ρυθμίσεων του συστήματος.

**Screenshot:** Επιλογές εμφάνισης UI



## Σενάριο 6: Αναζήτηση και Αίτηση Ενοικίασης Ακινήτου

**Στόχος:** Ο χρήστης φιλτράρει ακίνητα και υποβάλει αίτηση ενοικίασης.

**Βήματα:**

1. Κάνει login.
2. Επιλέγει το tab **Apartments For Rental**.
3. Ορίζει φίλτρα τιμής (Min / Max Rent).
4. Πατάει **Apply Filter**.

Filter apartments according to your preferred Monthly Rent:

Min Rent (€):

900

Max Rent (€):

1100

Apply Filter

5. Επιλέγει ένα ακίνητο και πατάει **Apply for Rental**.

Floor	Bathrooms	Bedrooms	City	Street	PC	Square Meters	Price (p.m.)	Parking	YearBuilt	
2	1	3	Athens	Kolonaki	10676	120	1000 €	false	1968	Apply for Rental
2	2	1	Ampelokipoi - Pentagon	Vrana 21	11525	75	1160 €	true	2004	Apply for Rental
1	1	2	Pagрати	Ippomedontos 5	11635	90	1200 €	true	1980	Apply for Rental
3	1	2	Koukaki	Mpeles 6	11741	71	920 €	false	1965	Apply for Rental
1	1	2	Likavittos	Tsimiski 18	11471	45	840 €	true	1961	Apply for Rental

6. Αν είναι η πρώτη φορά:

- Συμπληρώνει: First Name, Last Name, Phone Number

7. Πατάει **Submit**.

### Tenant Details

First Name

George

Last Name

Papadimitriou

Phone Number

6977721651

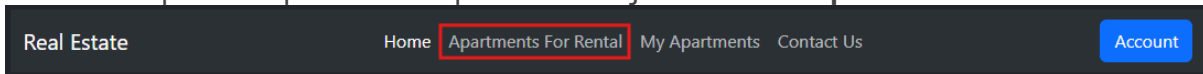
Submit

## Σενάριο 7: Καταχώρηση Νέου Ακινήτου από Ιδιοκτήτη

**Στόχος:** Ο ιδιοκτήτης προσθέτει νέο ακίνητο στην πλατφόρμα.

**Βήματα:**

1. Συνδέεται στο λογαριασμό του.
2. Μεταβαίνει στη σελίδα ακινήτων κάνοντας κλικ στο tab **Apartments For Rental**.



3. Επιλέγει **Create New Apartment**.

4. Συμπληρώνει:
  - Floor (όροφος)
  - Bathrooms (αριθμός μπάνιων)
  - Bedrooms (πλήθος δωματίων με κρεβάτι)
  - City (πόλη)
  - Street (Οδός ακινήτου)
  - Postal Code (Ταχυδρομικός Κώδικας)
  - Square Meters (Τετραγωνικά Μέτρα διαμερίσματος)
  - Price (Μηνιαία Τιμή Ενοικίασης)
  - Parking
  - Year Built (Χρονολογία Κατασκευής)

### Apartment Details

Floor

3

Bathrooms

1

Bedrooms

2

City

Koukaki

Street

Mpeles 6

Postal Code

11741

Square Meters

71

Price (Per Month)

920

Parking

☐ Yes

☒ No

Year built

1965

Submit Apartment

5. Αν είναι η πρώτη φορά:  
Συμπληρώνει: First Name (Όνομα), Last Name (Επώνυμο), Τηλέφωνο
6. Πατάει **Submit Apartment**.

### Owner Details

## Σενάριο 8: Προβολή Ακινήτων Ιδιοκτήτη και Διαγραφή Ακινήτου

**Στόχος:** Ο ιδιοκτήτης βλέπει τα καταχωρημένα του ακίνητα και αφαιρεί ένα από αυτά.

**Βήματα:**

1. Επιλέγει **My Apartments**.

Real Estate

[Home](#)
[Apartments For Rental](#)
[My Apartments](#)
[Contact Us](#)

Account

- Εμφανίζεται λίστα με τα δικά του ακίνητα.
- Εντοπίζει το ακίνητο.
- Πατάει **Delete**.
- Επιβεβαιώνει στο popup που εμφανίζεται.

1	1	2	Pagrat	Ippomedontos 5	11635	90	1200 €	true	1980	true	<div>Applications</div> <div>Delete</div>
---	---	---	--------	----------------	-------	----	--------	------	------	------	---

2. Εμφανίζεται λίστα με τα δικά του ακίνητα.
3. Εντοπίζει το ακίνητο.
4. Πατάει **Delete**.
5. Επιβεβαιώνει στο popup που εμφανίζεται.

## Σενάριο 9: Διαχείριση Αιτήσεων Ενοικίασης

**Στόχος:** Ο ιδιοκτήτης εξετάζει και εγκρίνει αιτήσεις.

**Βήματα:**

1. Επιλέγει **My Apartments**.
2. Επιλέγει ένα ακίνητο.
3. Πατάει **Applications**.

Floor	Bathrooms	Bedrooms	City	Street	PC	Square Meters	Price (p.m.)	Parking	YearBuilt	Approved	Actions
2	1	3	Athens	Kolonaki	10676	120	1000 €	false	1968	true	<div>Applications</div> <div>Delete</div>

4. Βλέπει λίστα αιτήσεων.
5. Πατάει **Approve** για αποδοχή.

### Applications for Apartment

First Name	Last Name	Email	Phone Number	Actions
John	Doe	JohnD@gmail.gr	6975721655	Approve

# Deployment

Το σύστημα έχει τη δυνατότητα να αναπτυχθεί σε διαφορετικά περιβάλλοντα:

- **Virtual Machines:**

Ένα VM για την εφαρμογή (Spring Boot)

Ένα VM για την βάση δεδομένων (PostgreSQL)

- **Dockerized περιβάλλον:**

Docker Compose με containers για την εφαρμογή και τη βάση δεδομένων

- **Kubernetes:**

Υλοποιημένα manifests για παραγωγή pods, services, ingress

Οι μεταβλητές της Ansible επιτρέπουν δυναμική παραμετροποίηση ανά περιβάλλον, υποστηρίζοντας scalable και modular ανάπτυξη.

## Ansible

Η Ansible υποδομή περιλαμβάνει:

- Playbooks/ :

postgres.yaml	Εγκατάσταση PostgreSQL service
spring.yaml	Ανάπτυξη της Spring Boot εφαρμογής
docker.yaml	Εγκατάσταση docker engine
deploy-all.yaml	Ολική εγκατάσταση
kubernetes.yaml	Εγκατάσταση πόρων στο K8s

- group\_vars/ :

- files/ :

spring.service.j2	systemd μονάδα για εκκίνηση της εφαρμογής ως service
nginx.http.j2	template για ρύθμιση αντίστροφου proxy

## Αυτοματοποίηση Ansible μέσω Terminal

```
krathws@krathws-X570-GAMING-K:~/DevOps/DevOps/realstate-monorepo/ansible/ansible-playgrounds
ansible-playbook playbooks/deploy-all.yaml --skip-tags dropdb

PLAY [Install and configure PostgreSQL] *****
*****

TASK [Gathering Facts] *****
*****
ok: [devops-vm-2]

TASK [Install PostgreSQL and psycopg2] *****
*****
ok: [devops-vm-2]

TASK [Edit postgres.conf to listen on all interfaces] *****
*****
ok: [devops-vm-2]

TASK [Edit pg_hba.conf to allow external connections] *****
*****
ok: [devops-vm-2]

TASK [Create PostgreSQL user] *****
*****
[WARNING]: Using world-readable permissions for temporary files Ansible needs to create when
becoming an unprivileged user. This may be insecure. For information on securing this, see
https://docs.ansible.com/ansible-core/2.18/playbook_guide/playbooks_privilege_escalation.html
#risks-of-becoming-an-unprivileged-user
ok: [devops-vm-2]

TASK [Create PostgreSQL database] *****
*****
ok: [devops-vm-2]

PLAY [appservers] *****
*****

TASK [Gathering Facts] *****
*****
ok: [devops-vm-1]

TASK [Install Java] *****
*****
ok: [devops-vm-1]

TASK [Install PostgreSQL client tools (psql) with retries] *****
*****
ok: [devops-vm-1]

TASK [Debug appdir] *****
*****
ok: [devops-vm-1] => {
  "msg": "App directory is /home/azureuser/spring-app/ds-exc-2024"
}

TASK [Clone the Spring repository] *****
*****
changed: [devops-vm-1]

TASK [Ensure mvnw is executable] *****
*****
changed: [devops-vm-1]

TASK [Build the Spring application] *****
*****
changed: [devops-vm-1]

TASK [Fetch SHA-512 checksum for Maven 3.9.9] *****
*****
skipping: [devops-vm-1]

TASK [Read the SHA-512 checksum from the file] *****
*****
skipping: [devops-vm-1]

TASK [Fix maven-wrapper.properties with correct SHA] *****
*****
skipping: [devops-vm-1]

TASK [Retry build after SHA fix] *****
*****
skipping: [devops-vm-1]

TASK [Install Spring systemd service] *****
*****
ok: [devops-vm-1]
```



```

TASK [Reload systemd] *****
*****
changed: [devops-vm-1]

TASK [Restart spring] *****
*****
changed: [devops-vm-1]

TASK [Enable and start Spring service] *****
*****
ok: [devops-vm-1]

TASK [Install Nginx] *****
*****
ok: [devops-vm-1]

TASK [Copy Nginx config] *****
*****
ok: [devops-vm-1]

TASK [Enable Spring Nginx site] *****
*****
ok: [devops-vm-1]

TASK [Disable default Nginx site] *****
*****
ok: [devops-vm-1]

PLAY RECAP *****
devops-vm-1      : ok=15  changed=5  unreachable=0  failed=0  skipped=4  r
escued=0  ignored=0
devops-vm-2      : ok=6    changed=0  unreachable=0  failed=0  skipped=0  r
escued=0  ignored=0

```

## Αυτοματοποίηση Docker μέσω Terminal

```

xrathws@xrathws-X570-GAMING-X:~/DevOps/DevOps/realstate-monorepo/ansible/ansible-playground$
ansible-playbook playbooks/docker.yaml

PLAY [Deploy application using Docker Compose] *****

TASK [Gathering Facts] *****
ok: [devops-vm-1]

TASK [Install Docker] *****
ok: [devops-vm-1]

TASK [Start and enable Docker] *****
ok: [devops-vm-1]

TASK [Add user to docker group] *****
ok: [devops-vm-1]

TASK [Install Docker Compose (v2 binary)] *****
ok: [devops-vm-1]

TASK [Create app directory] *****
ok: [devops-vm-1]

TASK [Copy docker-compose.yml] *****
ok: [devops-vm-1]

TASK [Run docker-compose (force rebuild)] *****
changed: [devops-vm-1]

PLAY RECAP *****
devops-vm-1      : ok=8    changed=1  unreachable=0  failed=0  skipped=0  r
escued=0  ignored=0

```

## Ansible – Docker

- Περιλαμβάνει docker-compose.yml που σηκώνει:
  - app: Spring Boot container με build από τοπικό ή public image
  - db: PostgreSQL με ρυθμίσεις για αποθήκευση δεδομένων (volumes)
- Η εγκατάσταση γίνεται με Ansible:
  - Αντιγραφή αρχείων
  - Εκτέλεση docker compose up -d

Το setup είναι έτοιμο για γρήγορη δοκιμή και local ανάπτυξη.

## Kubernetes

Το directory k8s/ περιλαμβάνει:

- **App (k8s/app/):**
  - app-deployment.yaml Ορίζει το deployment με image της εφαρμογής.
  - app-service.yaml Εκθέτει την εφαρμογή εσωτερικά.
  - app-ingress.yaml Ingress controller για HTTPS routing.
  - app-config.yaml Περιβαλλοντικές μεταβλητές.
  - app-secret.yaml Κρυφές μεταβλητές π.χ. DB credentials.
- **DB (k8s/db/):**
  - db-deployment.yaml PostgreSQL deployment
  - db-service.yaml ClusterIP service
  - db-pvc.yaml Persistent Volume Claim

Μέσω Microk8s, τα manifests είναι έτοιμα για χρήση με kubectl apply.

# CI/CD

Η εργασία υποστηρίζει pipelines Jenkins. Παρότι τα Jenkinsfiles δεν περιλαμβάνονται στον φάκελο, το μοντέλο υλοποίησης μπορεί να περιλαμβάνει τα εξής stages:

## Pipeline για Spring Boot App

1. **Checkout Code** (από το git)
2. **Build Application** (mvn clean install)
3. **Run Tests**
4. **Build Docker Image** (docker build)
5. **Push Image** (σε Docker Hub / registry)
6. **Deploy:**
  - a. Είτε μέσω Ansible playbook
  - b. Είτε με kubectl apply στο microk8s

## Γενικά Σχόλια/Παρατηρήσεις

- Η εργασία ακολουθεί αρχές modular DevOps αρχιτεκτονικής, με καθαρό διαχωρισμό εφαρμογής και υποδομής.
- Η χρήση Ansible παρέχει ευκολία στην αναπαραγωγή και επέκταση του συστήματος.
- Η υποστήριξη πολλαπλών περιβαλλόντων (VM, Docker, Kubernetes) αυξάνει την ευελιξία και την αξία της εργασίας.
- Για πλήρη παραγωγή, μπορεί να προστεθεί πλήρως ρυθμισμένο HTTPS + FQDN στο Ingress και real mail service (MailHog / min.io).

# Μας δυσκόλεψε / δεν υλοποιήσαμε

Η πλήρης λειτουργικότητα του Kubernetes αποδείχτηκε ιδιαίτερα απαιτητική: η εγκατάσταση και διαχείριση ενός πλήρως λειτουργικού cluster απαιτεί διαρκή συντονισμό μεταξύ πολλαπλών υποσυστημάτων, καθιστώντας κάθε βήμα ευαίσθητο σε εσφαλμένες ρυθμίσεις ή ασυμβατότητες.

Η ρύθμιση των ingress controllers σε συνδυασμό με την εφαρμογή κανόνων ασφάλειας στο επίπεδο pod-to-pod αποκάλυψαν ότι ακόμη και μικροαλλαγές μπορούν να πυροδοτήσουν απρόβλεπτα σφάλματα, καθιστώντας αναγκαία τη σχολαστική αυτοματοποίηση και συνεχόμενη παρακολούθηση για τη διασφάλιση της σταθερότητας.

## Κώδικας

### Αποθετήρια κώδικα

- https link: <https://github.com/AthosExarchou/DevOps.git>
- ssh link: `git@github.com:AthosExarchou/DevOps.git`

### Δοκιμαστικά accounts και urls

Docker «8080»	• Username: admin Password: admin – Role: ROLE_ADMIN
	• Username: owner Password: owner – Role: ROLE_USER, ROLE_OWNER
	• Username: tenant Password: tenant – Role: ROLE_USER, ROLE_TENANT
Ansible «8380» & K8s «8280»	• Username: admin Password: admin – Role: ROLE_ADMIN
	• Username: owner Password: owner – Role: ROLE_USER, ROLE_OWNER
	• Username: tenant Password: tenant – Role: ROLE_USER, ROLE_TENANT

## Url δοκιμαστικού περιβάλλοντος

- <http://realestate.ip-ddns.com:8080> => Ανοίγει με Docker
- <http://realestate.ip-ddns.com:8180> => Ανοίγει το Jenkins
- <http://realestate.ip-ddns.com:8280> => Ανοίγει με K8s
- <http://realestate.ip-ddns.com:8380> => Ανοίγει με Ansible

---

## Οδηγίες Χρήσης / Εγκατάστασης

Οδηγίες αυτοματοποίησης της εφαρμογής:

<https://github.com/AthosExarchou/DevOps/blob/main/README.md>

Περιγραφή και οδηγίες εγκατάστασης της εφαρμογής:

<https://github.com/AthosExarchou/DevOps/blob/main/ds-exc-2024/README.md>