

Εργασία Εξαμήνου

Μάθημα: Κατανεμημένα Συστήματα

Ανάπτυξη Διαδικτυακής Υπηρεσίας σε Κατανεμημένο Περιβάλλον

1. Σκοπός

Στο πλαίσιο του μαθήματος θα υλοποιήσετε μία διαδικτυακή εφαρμογή που λειτουργεί ως κόμβος μέσα σε ένα κατανεμημένο οικοσύστημα υπηρεσιών.

Κύριοι στόχοι:

- Δημιουργία / δόμηση υπηρεσίας**
Σχεδιασμός και υλοποίηση μιας καλά δομημένης υπηρεσίας (business logic, persistence, validation, κανόνες, διεπαφή χρήστη UI).
- Έκθεση υπηρεσίας με δομημένο, τεκμηριωμένο και ασφαλή τρόπο**
Ανάπτυξη REST API, τεκμηρίωση με OpenAPI/Swagger, μηχανισμοί authentication & authorization.
- Κατανάλωση εξωτερικής υπηρεσίας (black-box)**
Ενσωμάτωση τουλάχιστον μίας εξωτερικής υπηρεσίας μέσω REST client, με ξεκάθαρο συμβόλαιο (interface), χειρισμό σφαλμάτων και κατάλληλη απομόνωση (ports/adapters).

Προαιρετικά, μπορείτε να υλοποιήσετε:

- Single Page Application (SPA) που καταναλώνει το REST API της εφαρμογής.
- Δεύτερη απλή υπηρεσία που καταναλώνει το REST API.

2. Κοινές Τεχνικές Απαιτήσεις

Οι παρακάτω απαιτήσεις είναι κοινές για όλα τα θέματα:

1. Back-end

- Spring Boot (& Spring Framework modules).
- Layered architecture (Controller – Service – Repository), με επιλεκτική χρήση εννοιών Hexagonal architecture (Ports/Adapters για εξωτερικές υπηρεσίες).
- Validation σε επίπεδο DTOs & domain (π.χ. Bean Validation).

2. Βάση Δεδομένων

- Σχεσιακή βάση H2. Επιτρέπεται και συνίσταται η χρήση άλλων ΒΔ όπως η Postgres.
- Χρήση JPA/Hibernate (entities, repositories).

3. UI / MVC

- Web UI με Spring MVC, Thymeleaf, HTML, CSS και προεραϊτικά JavaScript.
- Φιλική βασική διεπαφή για τους ρόλους χρηστών.

4. REST API & Διαλειτουργικότητα

- Δημιουργία REST API (JSON) που καλύπτει τις βασικές λειτουργικότητες της εφαρμογής.
- Το API πρέπει να μπορεί να καταναλωθεί από άλλες υπηρεσίες ή τρίτα συστήματα (clients, SPA, mobile app κ.λπ.).

5. Ασφάλεια

- UI: Stateful, cookie-based authentication (Spring Security).
- API: Stateless authentication με JWT tokens.
- Διαχωρισμός ρόλων (π.χ. φοιτητής / καθηγητής / admin κ.λπ. ανά θέμα).

6. Τεκμηρίωση API

- OpenAPI Specification.
- Online προβολή μέσω Swagger UI (π.χ. springdoc-openapi)
- Προεραϊτικά, προβολή με Postman ή Bruno.

7. Εξωτερική Υπηρεσία (υποχρεωτικό)

- Κατανάλωση τουλάχιστον μίας εξωτερικής υπηρεσίας μέσω REST client (π.χ. notifications, verification, τρίτο σύστημα, δημόσιο API ή παρεχόμενη mock υπηρεσία από το μάθημα).
- Η εξωτερική υπηρεσία αντιμετωπίζεται ως black box: εσείς ξέρετε μόνο το συμβόλαιο (endpoints, request/response σχήματα).

8. Προαιρετικό Single Page Application (SPA) ή/και Υπηρεσία που καταναλώνει το API

- Στην περίπτωση Single Page Application (SPA):
 - Επιλογή framework: React ή Angular ή Vue.
 - Το SPA καταναλώνει αποκλειστικά το JSON REST API (χωρίς να “βλέπει” Thymeleaf).
 - Χρήση JWT για API calls, client-side routing.
- Στην περίπτωση της Εξωτερικής Υπηρεσίας που καταναλώνει το API
 - Spring Boot με ή χωρίς UI
 - Η εξωτερική υπηρεσία καταναλώνει αποκλειστικά το JSON REST API
 - Χρήση JWT για API calls

3. Οργάνωση Ομάδων, Παραδοτέα, Αξιολόγηση & Προθεσμίες

Η εργασία εκπονείται υποχρεωτικά σε **ομάδες τριών (3) μελών**.

Εξαίρεση γίνεται μόνο σε περιπτώσεις όπου, λόγω του συνολικού αριθμού των φοιτητών, δεν είναι εφικτός ο σχηματισμός πλήρων τριμελών ομάδων.

Κάθε ομάδα **παραδίδει**:

- Πλήρης πηγαίος κώδικας σε αποθετήριο/α κώδικα (π.χ. GitHub/GitLab/...).
- Σύντομη τεχνική αναφορά (~ 4-6 σελίδες) που να περιλαμβάνει:
 - Περιγραφή του προβλήματος και των ρόλων χρηστών που υποστηρίζονται.
 - Περιγραφή της αρχιτεκτονικής (layers, βασικά components, χρήση εξωτερικών υπηρεσιών).
 - Περιγραφή των εξωτερικών υπηρεσιών που καταναλώνετε (συμβόλαιο, τρόπος ενσωμάτωσης, χειρισμός σφαλμάτων).
 - Περιγραφή των μηχανισμών ασφάλειας (authentication, authorization, ρόλοι) και της τεκμηρίωσης του API (OpenAPI/Swagger κ.λπ.).
- Οδηγίες εκτέλεσης (π.χ. README).

Η **αξιολόγηση** της εργασίας γίνεται σε τρία επίπεδα:

- Αξιολόγηση παραδοτέων & κατανόησης από όλα τα μέλη
Εξετάζεται κατά πόσο η ομάδα, και κάθε μέλος της, είναι σε θέση να:
 - Περιγράψει τη λύση της.
 - Εξηγήσει τις επιλογές σχεδίασης.
 - Υποστηρίξει ότι η εφαρμογή λειτουργεί σωστά στο πλαίσιο ενός κατανεμημένου περιβάλλοντος και της λογικής των υπηρεσιών.
- Τεχνική αξιολόγηση βάσει των ενοτήτων 2.x
Αξιολογούνται τα επιμέρους τεχνικά κριτήρια που αντιστοιχούν στις ενότητες: 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7 (υποχρεωτικά μέρη) και το επιπλέον 2.8 (προαιρετικό).
- Παρουσίαση & live επίδειξη
Πραγματοποιείται ανοιχτή παρουσίαση της εργασίας.
 - Κάθε ομάδα παρουσιάζει συνοπτικά το πρόβλημα που έλυσε και την αρχιτεκτονική της.
 - Ακολουθεί live επίδειξη της κατανεμημένης εφαρμογής (UI, API, αλληλεπίδραση με εξωτερικές υπηρεσίες).
 - Κάθε μέλος της ομάδας παρουσιάζει ένα διακριτό μέρος του συστήματος ώστε να αξιολογηθεί η ατομική κατανόηση.

* Η γραφική διεπαφή δεν αξιολογείται σε όρους σχεδιασμού (~~UI/UX, styling, look and feel~~) παρά μόνο στο κατά πόσο υλοποιεί την επιχειρησιακή λογική.

Προθεσμίες:

1. Πρώτη Παράδοση: 04 Δεκεμβρίου 2025 (δεν αξιολογείται - αποσκοπεί σε έλεγχο προόδου)
2. Δεύτερη & Τελική Παράδοση: 12 Ιανουαρίου 2026
3. Παρουσιάσεις: Ενδεικτικά, χρονικό διάστημα από μέσα Ιανουαρίου έως αρχές Φεβρουαρίου 2026. Η ακριβής ημερομηνία και το πρόγραμμα παρουσιάσεων θα οριστικοποιηθούν με σχετική ανακοίνωση.

4. ΔΙΕΣΥΚΡΙΝΙΣΕΙΣ

Η χρήση του κώδικα που αναπτύχθηκε στα εργαστήρια ως βάση της εργασίας όχι μόνο επιτρέπεται αλλά και προτείνεται. Αυτό περιλαμβάνει όλα τα στοιχεία: κλάσεις, security configuration, database configuration, REST API configuration, υλοποιήσεις κ.τ.λ.

Επιτρέπεται η χρήση έτοιμου κώδικα ή εξωτερικών υπηρεσιών, υπό τις εξής προϋποθέσεις:

- Είναι open source ή ελεύθερα προσβάσιμες για ακαδημαϊκή χρήση.
- Γίνεται σαφής αναφορά στην τεχνική αναφορά και/ή στο README.
- Κάθε μέλος της ομάδας είναι σε θέση να εξηγήσει τι κάνει ο ξένος κώδικας και πώς προσθέτει αξία / λύνει πρόβλημα στο πλαίσιο της εφαρμογής.

Ενδεικτικά παραδείγματα επιτρεπόμενων υπηρεσιών:

- Υπηρεσία αποστολής email για ενημέρωση χρηστών (π.χ. επιβεβαίωση εγγραφής, υπενθύμιση ραντεβού, αλλαγή κατάστασης).
- Υπηρεσία αποστολής SMS για σύντομες ειδοποιήσεις (π.χ. κωδικός επιβεβαίωσης, υπενθύμιση κράτησης).
- Υπηρεσία αποθήκευσης αρχείων τύπου S3 για αποθήκευση/ανάκτηση αρχείων (π.χ. εικόνες προφίλ, έγγραφα, αποδείξεις).
- Υπηρεσία μοντέλου τεχνητής νοημοσύνης για την εκτέλεση ενός πολύ συγκεκριμένου, περιορισμένου αντικειμένου, π.χ.: αυτόματη πρόταση κατηγορίας για ένα αίτημα, αυτόματη δημιουργία σύντομης περιγραφής, απλό scoring/ταξινόμηση.
- Υπηρεσία γεωτοποθεσίας (geolocation / maps / distance matrix) για: υπολογισμό αποστάσεων, εκτίμηση χρόνου διαδρομής, εύρεση κοντινών σημείων.
- Υπηρεσία πρόγνωσης καιρού για άντληση καιρικών δεδομένων σε μια περιοχή (π.χ. για outdoor activities, εκδηλώσεις, μετακινήσεις).

Σε περίπτωση που επιθυμείτε να υλοποιήσετε δική σας mock υπηρεσία (π.χ. “ψεύτικη” υπηρεσία πληρωμών, notifications, AI κ.λπ.), αυτό επιτρέπεται υπό την προϋπόθεση ότι:

- Εκτίθεται μέσω σαφώς ορισμένου REST API (endpoints, request/response σχήματα).
- Χρησιμοποιείται από την κύρια εφαρμογή σας σαν να ήταν εξωτερικό τρίτο σύστημα (δηλαδή δεν “μπλέκεται” με το business logic).
- Τεκμηριώνετε το API της mock υπηρεσίας και εξηγείτε τον ρόλο της στην τεχνική αναφορά.

Η χρήση τεχνολογιών πέρα από αυτές που διδάσκονται στο μάθημα επιτρέπεται μόνο κατόπιν συνεννόησης με τους διδάσκοντες. Σε κάθε περίπτωση:

- Πρέπει να εξυπηρετούν τον σκοπό του εργαστηρίου (υπηρεσίες σε κατανεμημένο περιβάλλον).
- Πρέπει να πληρούν όλες τις προϋποθέσεις που αναφέρονται σε αυτό το έγγραφο (ασφάλεια, τεκμηρίωση, εξωτερικές υπηρεσίες κ.λπ.).

Αν επιθυμείτε να υλοποιήσετε **δικό σας θέμα**, θα πρέπει:

- Να το περιγράψετε (πρόβλημα, ρόλοι, λειτουργικότητες, εξωτερικές υπηρεσίες).
- Να το υποβάλετε έγκαιρα για έγκριση.
Εφόσον οι διδάσκοντες κρίνουν ότι το θέμα πληροί όλα τα κριτήρια του εργαστηρίου, εγκρίνεται. Η περιγραφή του θέματος, όπως και όλα τα εγκεκριμένα θέματα, θα κοινοποιηθεί σε όλες τις ομάδες.

5. Ενδεικτικά Θέματα

Θέμα 1: StreetFoodGo – Σύστημα Παραγγελίας & Παράδοσης από Μικρά Καταστήματα / Food Trucks

Σκοπός: Online παραγγελίες από μικρά καταστήματα/food trucks, με επιλογή παραλαβής ή παράδοσης.

Ρόλοι Χρηστών

- Επισκέπτης (ανώνυμος)
- Πελάτης
- Ιδιοκτήτης καταστήματος

Ενδεικτικές Λειτουργικότητες

- Επισκέπτης
 - Προβολή καταστημάτων ανά περιοχή/κουζίνα.
 - Προβολή μενού (χωρίς παραγγελία).
 - Εγγραφή ως πελάτης.
- Πελάτης
 - Login / logout.
 - Διαχείριση διευθύνσεων παράδοσης.
 - Δημιουργία παραγγελίας (κατάστημα, προϊόντα, παραλαβή/παράδοση).
 - Παρακολούθηση κατάστασης παραγγελίας (π.χ. “Σε προετοιμασία”, “Σε παράδοση”, “Ολοκληρωμένη”).
- Ιδιοκτήτης καταστήματος
 - Login / logout.
 - Διαχείριση μενού (πιάτα, τιμές, διαθεσιμότητα).
 - Αποδοχή / απόρριψη παραγγελιών.
 - Ενημέρωση κατάστασης παραγγελίας.

Κανόνες (προτάσεις)

- Δεν επιτρέπεται παραγγελία σε κατάστημα που είναι “κλειστό”.
- Δεν επιτρέπεται τροποποίηση παραγγελίας μόλις αυτή περάσει σε “Σε παράδοση”.
- Όριο ελάχιστης αξίας παραγγελίας ανά κατάστημα.

Εξωτερικές Υπηρεσίες (προτάσεις)

- Υπηρεσία χαρτών/γεωκωδικοποίησης για εκτίμηση χρόνου παράδοσης.
- Υπηρεσία ειδοποιήσεων (push/email/SMS) για ενημέρωση κατάστασης.

Θέμα 2: StudyRooms – Σύστημα Κράτησης Χώρων Μελέτης

Σκοπός: Κρατήσεις θέσεων/δωματίων μελέτης στη βιβλιοθήκη ή σε ειδικούς χώρους.

Ρόλοι Χρηστών

- Ανώνυμος
- Φοιτητής
- Προσωπικό βιβλιοθήκης

Ενδεικτικές Λειτουργικότητες

- Ανώνυμος
 - Προβολή βασικών πληροφοριών για χώρους μελέτης.
 - Εγγραφή φοιτητή.
- Φοιτητής
 - Login / logout.
 - Προβολή διαθεσιμότητας δωματίων/θέσεων σε ημερολόγιο.
 - Δημιουργία / ακύρωση κράτησης.
 - Προβολή ιστορικού κρατήσεων.
- Προσωπικό βιβλιοθήκης
 - Login / logout.
 - Δημιουργία/ενημέρωση χώρων μελέτης (χωρητικότητα, ωράριο).
 - Ακύρωση κρατήσεων σε ειδικές περιπτώσεις (π.χ. κλείσιμο χώρου).
 - Προβολή στατιστικών πληροτητας.

Κανόνες (προτάσεις)

- Κάθε φοιτητής μπορεί να έχει έως X ενεργές κρατήσεις ανά ημέρα.
- Δεν επιτρέπονται κρατήσεις εκτός ωραρίου λειτουργίας χώρου.
- Σε περίπτωση “no-show” (μη εμφάνιση), μπορεί να εφαρμόζεται penalty (π.χ. απαγόρευση νέας κράτησης για κάποιες μέρες – απλοποιημένα).

Εξωτερικές Υπηρεσίες (προτάσεις)

- Υπηρεσία ειδοποιήσεων (email/SMS).
- Δημόσια υπηρεσία εθνικών αργιών (holiday API) για να αποτρέπονται κρατήσεις σε ημέρες που η βιβλιοθήκη είναι κλειστή.

Θέμα 3: FitTrack – Σύστημα Προσωπικής Γυμναστικής & Ραντεβού με Trainers

Σκοπός: Πλατφόρμα οργάνωσης προπονήσεων, ραντεβού με personal trainers και παρακολούθησης προόδου.

Ρόλοι Χρηστών

- Επισκέπτης
- Χρήστης (πελάτης)
- Trainer

Ενδεικτικές Λειτουργικότητες

- Επισκέπτης
 - Προβολή διαθέσιμων trainers (όνομα, ειδίκευση, περιοχή).
 - Εγγραφή ως χρήστης.
- Χρήστης
 - Login / logout.
 - Διαχείριση βασικού “profile” (στόχοι: απώλεια βάρους, μυϊκή ενδυνάμωση κ.λπ.).
 - Κλείσιμο ραντεβού με trainer (ημερομηνία/ώρα, τύπος προπόνησης).
 - Καταγραφή προόδου (π.χ. βάρος, χρόνος τρεξίματος).
- Trainer
 - Διαχείριση διαθεσιμότητας (ωράριο/slots).
 - Αποδοχή/ακύρωση ραντεβού.
 - Καταγραφή σημειώσεων/πλάνου προπόνησης ανά χρήστη.

Κανόνες (προτάσεις)

- Όριο μέγιστων ενεργών ραντεβού ανά χρήστη.
- Δεν επιτρέπεται ραντεβού σε παρελθοντικές ημερομηνίες.
- Δεν επιτρέπονται επικαλυπτόμενα ραντεβού για τον ίδιο trainer.

Εξωτερικές Υπηρεσίες (προτάσεις)

- Υπηρεσία health/fitness API ή απλό public weather API για εμφάνιση καιρού την ημέρα της outdoor προπόνησης.
- Ειδοποιήσεις (email/SMS) για υπενθύμιση ραντεβού.

Θέμα 4: GreenRide – Σύστημα Διαμοιρασμού Διαδρομών (Carpooling) για Μετακίνηση στην Πόλη/Πανεπιστήμιο

Σκοπός: Carpooling μεταξύ φοιτητών/εργαζομένων για να μοιράζονται διαδρομές προς/από το Πανεπιστήμιο ή το κέντρο.

Ρόλοι Χρηστών

- Χρήστης (μπορεί να είναι και οδηγός και επιβάτης)
- Διαχειριστής

Ενδεικτικές Λειτουργικότητες

- Χρήστης
 - Δημιουργία διαδρομής ως οδηγός (αφετηρία, προορισμός, ώρα, διαθέσιμες θέσεις)
 - Αναζήτηση διαθέσιμων διαδρομών ως επιβάτης, κράτηση θέσης, ακύρωση.
 - Προβολή ιστορικού διαδρομών, βασική αξιολόγηση οδηγού/επιβατών.
- Διαχειριστής
 - Διαχείριση κακόβουλων χρηστών, στατιστικά (αριθμός διαδρομών, μέσος όρος πληρότητας).

Κανόνες (προτάσεις)

- Όριο ενεργών διαδρομών ανά οδηγό.
- Δεν επιτρέπεται κράτηση πάνω από τις διαθέσιμες θέσεις.
- Cutoff χρονικής ακύρωσης (π.χ. όχι ακύρωση 10' πριν την αναχώρηση).

Εξωτερικές Υπηρεσίες (προτάσεις)

- Geolocation / Maps API για απόσταση & εκτίμηση χρόνου διαδρομής.
- Weather API για εμφάνιση καιρού κατά την ημέρα διαδρομής (πληροφοριακά).

Θέμα 5: PetCare – Σύστημα Ραντεβού & Ιστορικού Υγείας Κατοικιδίων

Σκοπός: Διαχείριση ραντεβού σε κτηνίατρο και βασικού ιστορικού υγείας για κατοικίδια.

Ρόλοι Χρηστών

- Ιδιοκτήτης κατοικίδιου
- Κτηνίατρος

Ενδεικτικές Λειτουργικότητες

- Ιδιοκτήτης
 - Καταχώριση κατοικιδίων (είδος, ράτσα, ηλικία)
 - Κλείσιμο ραντεβού
 - Προβολή ιστορικού επισκέψεων/εμβολίων.
- Κτηνίατρος
 - Διαθεσιμότητα
 - Επιβεβαίωση/ακύρωση ραντεβού
 - Καταχώριση σημειώσεων/θεραπειών.

Κανόνες (προτάσεις)

- Δεν επιτρέπονται επικαλυπτόμενα ραντεβού για τον ίδιο επαγγελματία.
- Ελάχιστος χρόνος μεταξύ δύο ραντεβού για το ίδιο κατοικίδιο (π.χ. για εμβόλια).
- Ιδιοκτήτης βλέπει μόνο τα δικά του κατοικίδια/ραντεβού.

Εξωτερικές υπηρεσίες (προτάσεις)

- Email/SMS για υπενθύμιση εμβολίων/ραντεβού.
- API για λίστα τυπικών εμβολίων ανά είδος ζώου (mock).

Θέμα 6: MyCityGov – Πύλη Ψηφιακών Αιτημάτων & Ραντεβού με τον Δήμο

Σκοπός: Πλατφόρμα όπου οι πολίτες μπορούν να:

1. υποβάλουν ψηφιακά αιτήματα/αιτήσεις προς τον Δήμο (βεβαιώσεις, άδειες, πιστοποιητικά),
2. δηλώνουν προβλήματα στην πόλη (λακούβες, φωτισμός, καθαριότητα),
3. κλείνουν ραντεβού με δημοτικές υπηρεσίες (ΚΕΠ, Τεχνική Υπηρεσία, Κοινωνική Υπηρεσία κ.λπ.),
4. και οι υπηρεσίες του Δήμου να τα επεξεργάζονται, με πλήρη ιχνηλασιμότητα και ενημέρωση του πολίτη.

Ρόλοι Χρηστών

- Πολίτης
- Υπάλληλος Υπηρεσίας (employee / case handler)
- Διαχειριστής Δήμου (Admin)

Ενδεικτικές Λειτουργικότητες

- Πολίτης
 - Εγγραφή / login (με απλούς κωδικούς ή μέσω εξωτερικής γον ταυτότητας).
 - Προβολή καταλόγου διαθέσιμων “υπηρεσιών”/αιτημάτων (π.χ. “Βεβαίωση μόνιμης κατοικίας”, “Άδεια κατάληψης πεζοδρομίου”, “Αναφορά προβλήματος φωτισμού”).
 - Υποβολή αιτήματος με συγκεκριμένο τύπο, στοιχεία και επισύναψη δικαιολογητικών (αρχεία).
 - Παρακολούθηση κατάστασης αιτήματος.
 - Κλείσιμο ραντεβού με συγκεκριμένη υπηρεσία, επιλογή ημερομηνίας/ώρας.
- Υπάλληλος Υπηρεσίας
 - Login / logout.
 - Προβολή λίστας αιτημάτων που ανήκουν στη δική του υπηρεσία.
 - Ανάθεση αιτήματος σε συγκεκριμένο υπάλληλο (ή ανάληψη).
 - Ενημέρωση κατάστασης, προσθήκη σχολίων/παρατηρήσεων.
 - Έγκριση/απόρριψη αιτήματος με τεκμηρίωση.
 - Διαχείριση ραντεβού (επιβεβαίωση/αλλαγή ώρας/ακύρωση).
- Διαχειριστής Δήμου
 - Διαχείριση τύπων αιτημάτων (προσθήκη/ενεργοποίηση/απενεργοποίηση).
 - Αντιστοίχιση τύπων αιτημάτων με υπηρεσίες/τμήματα.
 - Ορισμός ωραρίων για ραντεβού ανά υπηρεσία.
 - Προβολή στατιστικών

Κανόνες (προτάσεις)

- Κάθε αίτημα έχει μοναδικό αριθμό πρωτοκόλλου/αναφοράς και έχει προθεσμία διεκπεραίωσης (SLA) ανάλογα με τον τύπο του.
- Καταστάσεις αιτήματος: Υποβλήθηκε, Σε παραλαβή, Σε επεξεργασία, Σε αναμονή επιπλέον στοιχείων, Ολοκληρωμένο, Απορρίφθηκε.
- Ο πολίτης μπορεί να δει μόνο τα δικά του αιτήματα και τα δημόσια στοιχεία δημοτικών δράσεων (αν βάλετε).
- Δεν επιτρέπονται επικαλυπτόμενα ραντεβού για τον ίδιο υπάλληλο ή για το ίδιο γκισέ/γραφείο.
- Αν το αίτημα μείνει σε συγκεκριμένη κατάσταση (π.χ. “Σε παραλαβή”) πάνω από X ημέρες, μπορεί να εμφανίζεται ως “εκπρόθεσμο” στις αναφορές του διαχειριστή.

Εξωτερικές υπηρεσίες (προτάσεις)

- Υπηρεσία ταυτοποίησης πολιτών (mock gov login). Για παράδειγμα, ένα external authentication service που δέχεται user token και επιστρέφει βασικά στοιχεία πολίτη (ΑΦΜ, ΑΜΚΑ, ονοματεπώνυμο).
- Υπηρεσία αποστολής email / SMS για ενημέρωση πολίτη για
- Υπηρεσία αποθήκευσης αρχείων τύπου S3 (MinIO)