



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Πανεπιστήμιο Πατρών

Τμήμα Μηχανικών Η/Υ και Πληροφορικής

Τεχνολογία Λογισμικού

Εαρινό εξάμηνο 2024-2025



Domain-model-v0.2

Github: <https://github.com/giorgosmelanis/EventHub>

Μέλη Ομάδας:

- | | |
|------------------------|-------------|
| • Αλεξίου Κωνσταντίνος | AM: 1058083 |
| • Μελάνης Γιώργος | AM: 1067378 |
| • Ρέρρας Νικόλας | AM: 1067411 |
| • Φεφές Αλέξανδρος | AM: 1040926 |
| • Φραγκούλης Σπύρος | AM: 1090070 |

Ρόλοι για use-cases-v0.1:

- Editor: Αλεξίου Κωνσταντίνος
- Contributors: Μελάνης Γεώργιος
- Peer Reviewer/Quality Manager: Φραγκούλης Σπύρος

Εργαλεία:

- Word (Συγγραφή κειμένου)
- Microsoft Visio (Διάγραμμα)

Περιεχόμενα:

1. Εισαγωγή
2. Αλλαγές από v0.1 σε v0.2
3. Domain-model-diagram-v0.2
4. Περιγραφές κλάσεων και attributes

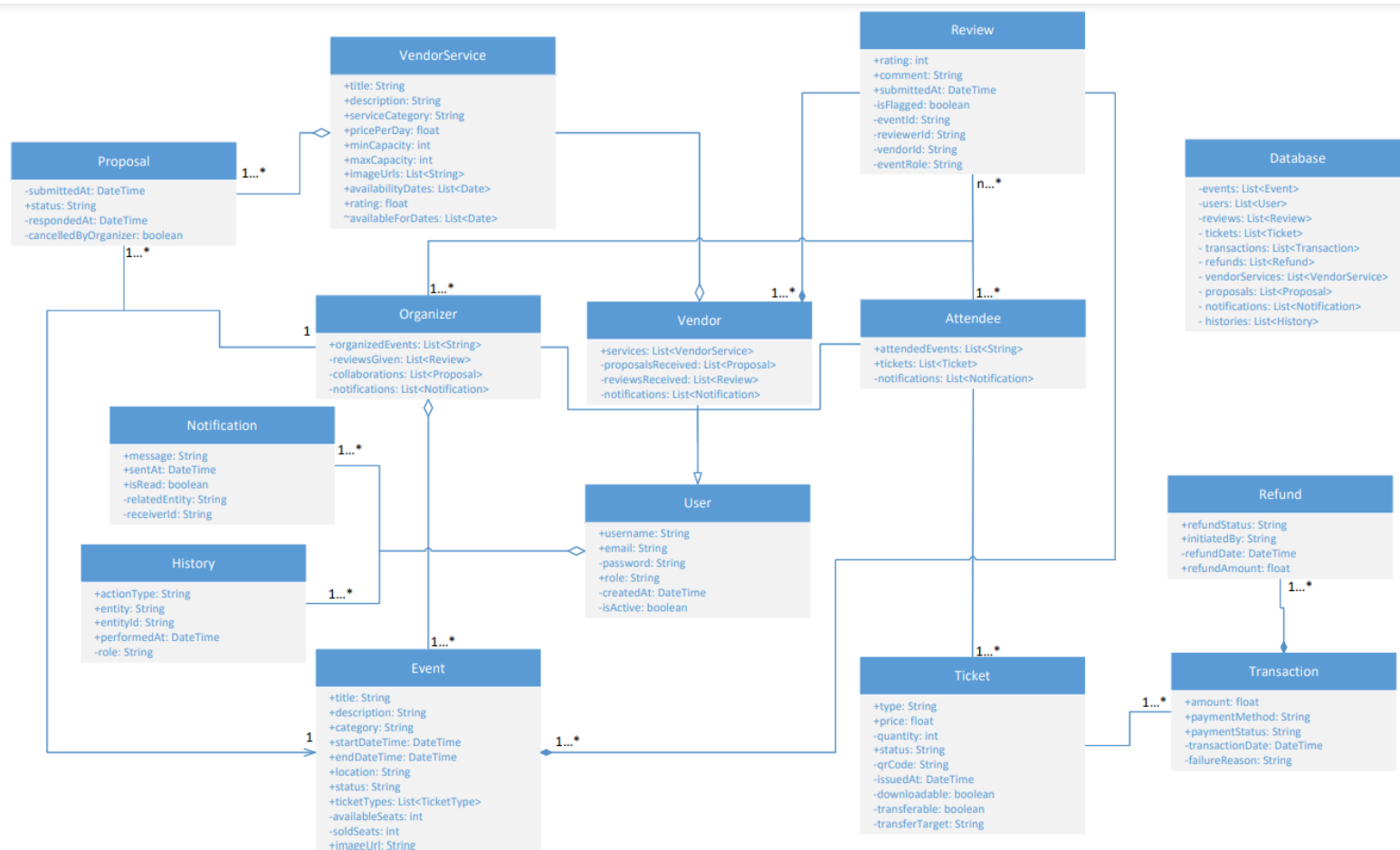
1. Εισαγωγή

Στο συγκεκριμένο κείμενο θα αναλύσουμε το **domain model** της εφαρμογής **EventHub**. Την σύνταξη του τελικού κειμένου ανέλαβε ο Αλεξίου Κωνσταντίνος, με τον Γιώργο Μελάνη να συνεισφέρει στην οπτικοποίηση του διαγράμματος μέσω Microsoft Visio. Τον τελικό έλεγχο και εξασφάλιση ποιότητας του τελικού παραδοτέου ανέλαβε ο Φραγκούλης Σπύρος.

2. Αλλαγές από v0.1 σε v0.2

Η παρούσα έκδοση v0.2 αποτελεί εμπλουτισμένη μορφή της έκδοσης v0.1 του domain model με την προσθήκη attributes στις κλάσεις του διαγράμματος. Οι περιγραφές των attributes ακολουθούν στο **4. Περιγραφές κλάσεων και attributes** με **κόκκινη** χρωματική επισήμανση.

3. Domain-model-diagram-v0.2



4. Περιγραφές κλάσεων και attributes

User: Αφηρημένη κλάση που εκπροσωπεί κάθε χρήστη του συστήματος. Από αυτήν κληρονομούν οι Attendee, Organizer και Vendor.

+ username: String: Μοναδικό όνομα χρήστη – εμφανίζεται παντού στην πλατφόρμα

+ email: String: Email επικοινωνίας – χρειάζεται για ειδοποιήσεις και login

+ role: String: Ο ρόλος του χρήστη (Attendee, Organizer, Vendor)

- createdAt: DateTime: Ημερομηνία δημιουργίας

- isActive: boolean: Αν είναι ενεργός λογαριασμός – χρησιμοποιείται σε έλεγχο συστήματος

Attendee: Χρήστης που συμμετέχει σε εκδηλώσεις και αγοράζει εισιτήρια. Μπορεί να αξιολογήσει εκδηλώσεις και vendors.

+ attendedEvents: List<String>: Εκδηλώσεις στις οποίες έχει παρευρεθεί – εμφανίζονται στο προφίλ του

+ tickets: List<Ticket>: Εισιτήρια που έχει αγοράσει – προβολή και διαχείριση από τον ίδιο

- notifications: List<Notification>: Ειδοποιήσεις – αποστέλλονται και διαβάζονται μέσω συστήματος

Organizer: Χρήστης που δημιουργεί και διαχειρίζεται εκδηλώσεις. Έχει τη δυνατότητα να επιλέγει vendors και να αξιολογεί τις υπηρεσίες τους.

+ organizedEvents: List<String>: Λίστα με IDs εκδηλώσεων που έχει δημιουργήσει ο Organizer

- reviewsGiven: List<Review>: Αξιολογήσεις που έχει υποβάλει για Vendors

- collaborations: List<Proposal>: Προτάσεις/συνεργασίες που έχει στείλει σε Vendors

Vendor: Χρήστης που προσφέρει υπηρεσίες για εκδηλώσεις. Υποβάλλει προτάσεις (proposals) και δέχεται αξιολογήσεις.

+ services: List<VendorService>: Οι υπηρεσίες που έχει ανεβάσει ο Vendor – προβολή σε organizers

- proposalsReceived: List<Proposal>: Προτάσεις συνεργασίας από Organizer

- reviewsReceived: List<Review>: Αξιολογήσεις που έχει λάβει – επεξεργάζεται από backend

- notifications: List<Notification>: Ειδοποιήσεις για νέες προτάσεις, αξιολογήσεις κ.λπ.

Event: Αντικείμενο που αναπαριστά μια εκδήλωση. Δημιουργείται από Organizer, έχει συμμετοχές από Attendees και συνεργασίες με Vendors.

+ title: String: Τίτλος της εκδήλωσης – εμφανίζεται στο UI

+ description: String: Περιγραφή – για το κοινό

+ category: String: Κατηγορία (π.χ. μουσική, αθλητικά)

+ startDateTime: DateTime: Ημερομηνία και ώρα έναρξης

+ endDateTime: DateTime: Ημερομηνία και ώρα λήξης

+ location: String: Τοποθεσία διεξαγωγής

+ status: String: Κατάσταση (draft, published, cancelled)

+ ticketTypes: List<TicketType>: Τύποι εισιτηρίων (VIP, General κ.λπ.)

- availableSeats: int : Εσωτερική πληροφορία για κρατήσεις

- soldSeats: int : Πόσα εισιτήρια έχουν πουληθεί – εσωτερικό

+ imageUrl: String: Εικόνα προώθησης για την κάρτα του event

Ticket: Εισιτήριο που αγοράζεται από Attendee για συμμετοχή σε Event. Συνδέεται με συναλλαγή πληρωμής.

- + type: String : Ο τύπος εισιτηρίου (π.χ. VIP, General Admission) – είναι δημόσιο γιατί προβάλλεται στον χρήστη
- + price: float : Η τιμή του εισιτηρίου – δημόσιο για προβολή και υπολογισμό κόστους
- quantity: int : Αριθμός εισιτηρίων που αγοράστηκαν – εσωτερική χρήση
- status: String : Κατάσταση εισιτηρίου (π.χ. active, cancelled) – αλλάζει από το σύστημα
- qrCode: String : Μοναδικός κωδικός QR για check-in – δεν εκτίθεται απευθείας
- issuedAt: DateTime : Χρονική στιγμή έκδοσης – για εσωτερικό έλεγχο
- downloadable: boolean : Αν επιτρέπεται λήψη του εισιτηρίου – σχετίζεται με πολιτική
- transferable: boolean : Αν επιτρέπεται μεταβίβαση του εισιτηρίου
- transferTarget: String : Παραλήπτης μεταβίβασης – εσωτερικό πεδίο

Transaction: Οικονομική συναλλαγή για την πληρωμή εισιτηρίων ή υπηρεσιών. Συνδέεται με Refund αν απαιτείται.

- + amount: float : Ποσό που πληρώθηκε – πρέπει να εμφανίζεται στον χρήστη
- + paymentMethod: String : Τρόπος πληρωμής (π.χ. κάρτα, PayPal)
- + paymentStatus: String : Κατάσταση πληρωμής (π.χ. success, failed)
- transactionDate: DateTime : Πότε πραγματοποιήθηκε η πληρωμή – τεχνικό στοιχείο
- failureReason: String : Αιτιολογία αποτυχίας – εσωτερικό για logs/υποστήριξη

Refund: Αντικείμενο που αναπαριστά επιστροφή χρημάτων, σε περίπτωση ακύρωσης εισιτηρίου ή άλλου προβλήματος.

- + refundStatus: String : Κατάσταση επιστροφής χρημάτων (π.χ. pending, success, failed) – εμφανίζεται στον χρήστη
- + initiatedBy: String : Ποιος ξεκίνησε την επιστροφή (Attendee ή Organizer) – εμφανίζεται στον χρήστη
- refundDate: DateTime : Ημερομηνία εκκίνησης της επιστροφής – εσωτερικό για tracking
- + refundAmount: float : Το ποσό που επιστράφηκε – εμφανίζεται στον χρήστη

Review: Αξιολόγηση που υποβάλλεται από Attendee ή Organizer, σχετική με ένα Event ή έναν Vendor.

- + rating: int : Βαθμολογία (1 έως 5) – δημόσια γιατί εμφανίζεται στο event/vendor
- + comment: String : Σχόλιο του χρήστη – εμφανίζεται δημόσια
- + submittedAt: DateTime : Ημερομηνία και ώρα υποβολής – εμφανίζεται δημόσια
- isFlagged: boolean : Αν έχει επισημανθεί ως ακατάλληλη – για moderation
- eventId: String : Αναγνωριστικό του event – τεχνικό συσχετισμού
- reviewerId: String : Χρήστης που υπέβαλε την αξιολόγηση – για backend
- vendorId: String : Vendor που αξιολογήθηκε – εσωτερικό συσχετισμού
- eventRole: String : Ο ρόλος του vendor στο event (π.χ. DJ, Catering)

VendorService: Υπηρεσία που προσφέρεται από Vendor (π.χ. ήχος, catering, φωτισμός). Χρησιμοποιείται για συμμετοχή σε Proposals.

- + title: String : Τίτλος της υπηρεσίας (π.χ. “DJ πακέτο”)
- + description: String : Περιγραφή της υπηρεσίας που προσφέρει ο vendor
- + serviceCategory: String : Κατηγορία (π.χ. Catering, Lighting, Sound)

- + pricePerDay: float : Κόστος παροχής ανά ημέρα/πακέτο
- minCapacity: int : Ελάχιστος αριθμός ατόμων που εξυπηρετεί – για φίλτρα & έλεγχο
- maxCapacity: int : Μέγιστος αριθμός ατόμων που εξυπηρετεί
- + imageUrls: List<String> : URLs εικόνων/πολυμέσων για προβολή υπηρεσίας
- + availabilityDates: List<Date> : Ημερομηνίες κατά τις οποίες η υπηρεσία είναι διαθέσιμη
- + rating: float : Μέση βαθμολογία υπηρεσίας από organizers
- ~ availableForDates: List<Date> : Υπολογιζόμενο πεδίο διαθεσιμότητας για συγκεκριμένο event

Proposal: Πρόταση συνεργασίας μεταξύ ενός Organizer και ενός Vendor. Περιλαμβάνει συγκεκριμένες υπηρεσίες και όρους.

- submittedAt: DateTime : Ημερομηνία και ώρα υποβολής της πρότασης από τον vendor
- + status: String : Τρέχουσα κατάσταση πρότασης (π.χ. pending, accepted, rejected)
- respondedAt: DateTime : Ημερομηνία και ώρα απάντησης από τον organizer
- cancelledByOrganizer: boolean : Αν ακυρώθηκε από τον organizer πριν την απάντηση

Notification: Ειδοποίηση που σχετίζεται με τον χρήστη και τα Events του. Ενημερώνει για σημαντικές ενέργειες ή αλλαγές.

- + message: String : Περιεχόμενο της ειδοποίησης (π.χ. “Η πληρωμή ολοκληρώθηκε”)
- + sentAt: DateTime : Ημερομηνία και ώρα αποστολής της ειδοποίησης
- + isRead: boolean : Αν έχει διαβαστεί η ειδοποίηση από τον χρήστη
- relatedEntity: String : Σχετική οντότητα (π.χ. Event, Ticket) – εσωτερική πληροφορία

- receiverId: String : Χρήστης που τη λαμβάνει – τεχνικό συσχετισμού

History: Ιστορικό ενεργειών και δραστηριοτήτων του χρήστη στην πλατφόρμα. Περιλαμβάνει παρελθόντα γεγονότα.

+ actionType: String : Τύπος ενέργειας (π.χ. “Αγορά Εισιτηρίου”) – εμφανίζεται στο ιστορικό του χρήστη

+ entity: String : Τύπος οντότητας στην οποία έγινε η ενέργεια (π.χ. Ticket, Event)

+ entityId: String : Αναγνωριστικό της συγκεκριμένης οντότητας

+ performedAt: DateTime : Ημερομηνία και ώρα εκτέλεσης της ενέργειας

- role: String : Ρόλος του χρήστη κατά την ενέργεια (π.χ. Attendee, Vendor)

Database: Αφηρημένη οντότητα που δηλώνει την αποθήκευση των δεδομένων. Πιθανόν να μην παραμείνει ως κλάση στο τελικό μοντέλο. Τυπικά μπορούμε να πούμε ότι θα περιέχει στοιχεία όπως:

users: List<User> : Οι χρήστες του συστήματος (Attendees, Organizers, Vendors).

events: List<Event> : Όλες οι εκδηλώσεις που εμφανίζονται στο σύστημα.

reviews: List<Review> : Όλες οι αξιολογήσεις – εφόσον πρόκειται να προβληθούν ή φιλτραριστούν σε dashboard (αναφέρονται και στο UI).

tickets: List<Ticket> : Τα εισιτήρια χρηστών, χρήσιμα για ιστορικό, QR, refunds και user dashboard.

transactions: List<Transaction> : Οι οικονομικές κινήσεις που σχετίζονται με αγορές εισιτηρίων ή υπηρεσιών.

refunds: List<Refund> : Οι επιστροφές χρημάτων, συχνά ενεργοποιούνται από cancel actions.

vendorServices: List<VendorService> : Οι διαθέσιμες υπηρεσίες που προσφέρουν οι Vendors (φαίνονται στο UI).

proposals: List<Proposal> : Οι υποβληθείσες προτάσεις συνεργασίας (μετά τη φάση εύρεσης events για vendors).

notifications: List<Notification> : Χρήσιμες για τα dashboards όλων των ρόλων. Αναφέρονται και στα use cases.

histories: List<History> : Για ιστορικό χρηστών ανά ρόλο (Attendee, Organizer, Vendor), ειδικά για tracking.

Τέλος domain-model-v0.2