Εργασία μαθήματος

«Κατανεμημένα Συστήματα»

Crowd Funding

Παραδοτέο 3

Τελικό Παραδοτέο

Ομάδα 79

Μάξιμος Σπυρίδων Διακουμάκος, it2021027

Χριστίνα Μπαχού, it2021066

Αναστάσιος Πούτος, it2021084

Ιανουάριος 2025

Περιεχόμενα

[**Εισαγωγή** 3](#_Toc189847328)

[**Σύνδεσμος για το Github Repository** 3](#_Toc189847329)

[**Παραδοχές** 3](#_Toc189847330)

[**Γενικές Παρατηρήσεις** 4](#_Toc189847331)

[**Εκτέλεση εφαρμογής** 4](#_Toc189847332)

[**Υλοποίηση Backend** 4](#_Toc189847333)

[**Config** 4](#_Toc189847334)

[**Entity** 4](#_Toc189847335)

[**Controller** 5](#_Toc189847336)

[**Payload** 6](#_Toc189847337)

[**Repository** 6](#_Toc189847338)

[**Service** 7](#_Toc189847339)

[**Διαγράμματα UML** 8](#_Toc189847340)

[**Υλοποίηση του Frontend** 13](#_Toc189847341)

[**Αρχεία Views** 13](#_Toc189847342)

[**Components** 14](#_Toc189847343)

[**Εγχειρίδιο χρήσης Frontend** 15](#_Toc189847344)

# **Εισαγωγή**

## **Σύνδεσμος για το Github Repository**

<https://github.com/MaximosSpyridonDiakoumakos/DS_Assignment_2024-CrowdFunding>

## **Παραδοχές**

Για την εφαρμογή αιτήσεων αγροτικών συνεταιρισμών, παραθέτουμε κάποιες παραδοχές/υποθέσεις που έχουν γίνει:

1. Υποθέτουμε ότι ο πελάτης έχει πρόσβαση σε ένα smartphone ή υπολογιστή με σύνδεση στο διαδίκτυο για να χρησιμοποιήσει την εφαρμογή.
2. Υποθέτουμε ότι η εφαρμογή θα λειτουργεί σε ένα αξιόπιστο δίκτυο και δεν θα υπάρχουν συνεχή προβλήματα συνδεσιμότητας.
3. Σε περίπτωση που κάποιος θέλει να χρησιμοποιήσει την ιστοσελίδα είτε είναι δημιουργός κάποιου project είτε θέλει να υποστηρίξει κάποιο project πρέπει να δημιουργήσει λογαριασμό εάν δεν έχουν.
4. Θεωρούμε ότι οι χρήστες γνωρίζουν ότι πρέπει να συνδεθούν με τον λογαριασμό τους, ή να δημιουργήσουν έναν στην περίπτωση που δεν έχουν, για να χρησιμοποιήσουν την ιστοσελίδα.
5. Και οι υποστηρικτές και οι δημιουργοί θεωρούνται χρήστες οπότε δεν έχουν πρόσβαση ούτε στη διαχείριση χρηστών ούτε στη διαχείριση έργων.
6. Για τις ενημερώσεις και τις ειδοποιήσεις ο υποστηρικτής θα έχει ξεχωριστή σελίδα.
7. Ένας δημιουργός ενημερώνει όλους τους υποστηρικτές του με γενικές ανακοινώσεις και δεν στέλνει στον καθένα ξεχωριστά.

Για την διαχείριση των χρηστών και των έργων αποφασίσαμε ότι ο διαχειριστής του συστήματος θα κάνει login στη σελίδα από την frontend εφαρμογή, έπειτα θα εμφανίζονται μόνο σε αυτόν οι επιλογές του διαχειριστή καθώς ο λογαριασμός του χρήστη θα έχει το ρόλο του admin. Δεν θα γίνεται κάποιος να δημιουργήσει λογαριασμό με δικαιώματα διαχειριστή καθώς ο διαχειριστής του συστήματος δεν είναι ένας καθημερινός χρήστης.

## **Γενικές Παρατηρήσεις**

Για την υλοποίηση του προγράμματος και της βάσης δεδομένων χρησιμοποιήθηκε το docker και postgresql

Το Backend χρησιμοποιεί το port:9090 ενώ το Frontend το port:9000.

Μετά την σύνδεση ενός χρήστη και την ταυτοποίηση του έχει παρατηρηθεί ότι κάποιες φορές απαιτείται να γίνει refresh η σελίδα για να φορτώσει το AppHeader.

## **Εκτέλεση εφαρμογής**

Για την εκτέλεση της εφαρμογής, αφού βεβαιωθούμε ότι τρέχουμε το Docker, χρησιμοποιούμε το τερματικό του Intellij IDEA, όπου αφού έχουμε πάει στο directory του backend χρησιμοποιούμε την εντολή:

docker-compose up -–build.

# **Υλοποίηση Backend**

Για την υλοποίηση του backend δημιουργήθηκαν τα εξής packages:

1. **Config**
2. **Controller**
3. **Entity**
4. **Payload**
5. **Repository**
6. **Service**

## **Config**

Αναλυτικότερα, στο **Config** βρίσκονται τα classes AppConfig η οποία είναι απαραίτητη για την δοκιμή του backend και των API request μέσω του Swagger, AuthEntryPointJwt η οποία διαχειρίζεται απόπειρες πρόσβασης χωρίς εξουσιοδότηση, εφαρμόζει το AuthenticationEntryPoint του Spring Security και στέλνει απάντηση 401 όταν ένας χρήστης προσπαθεί να έχει πρόσβαση σε κάτι χωρίς εξουσιοδότηση, χωρίς τον απαραίτητο ρόλο, είτε όταν το JWT token του έχει ή λήξει ή δεν υπάρχει, AuthTokenFilter η οποία αναχαιτίζει κάθε HTTP request, αντλεί το JWT token από το Authorization header, το επικυρώνει, φορτώνει τα στοιχεία του χρήστη εάν αυτό είναι έγκυρο και ρυθμίζει το πλαίσιο ελέγχου, JwtUtils η οποία παρέχει τις μεθόδους για την δημιουργία, την ανάλυση και την επικύρωση των JWT tokens, χρησιμοποιεί secret key και χρόνο λήξης από το application.properties, RestConfiguration η οποία αποκαλύπτει τα ID για τα entities: Project, Supporter, Creator και Pledge. Τέλος η κλάση SecurityConfig η οποία διαμορφώνει τις ρυθμίσεις του Spring Security, ορίζει τις πολιτικές CORS, καθορίζει public και protected endpoints, ρυθμίζει τον έλεγχο ταυτοποίησης JWT και ενεργοποιεί τα security annotations σε επίπεδο μεθόδου.

## **Entity**

Έπειτα υπάρχει ένα πακέτο για τα **Entity classes, entity**, τα οποίας είναι υπεύθυνα για το setup των tables της βάσης δεδομένων. Υπάρχει μία κλάση για τους χρήστες (User.java), τα project (Project.java), τις υποστηρίξεις αυτών (Pledge.java), τους ρόλους (Role.java), καθώς και για τις ενημερώσεις των έργων (Project Update.java).

* User.java: περιέχει ακέραιο αριθμό id, το username του χρήστη, το πρώτο όνομα καθώς και το επίθετό του, το email το οποίο πρέπει να έχει μορφή @Email, τον κωδικό του, καθώς και joint Table το οποίο χρησιμοποιεί τα ids των χρηστών και των ρόλων για την ανάθεση του ρόλου σε κάθε χρήστη.
* Project.java: περιέχει ακέραιο αριθμό id, τον τίτλο του έργου, το στόχο χρηματοδότησης, την τρέχουσα χρηματοδότηση, την ημερομηνία λήξης, μεταβλητή Boolean για το εάν έχει εγκριθεί, λίστα με τις υποστηρίξεις που έχει λάβει και λίστα με τις ενημερώσεις που έχει δεχθεί.
* Pledge.java: ακέραιο αριθμό id, το ποσό του pledge, την τοπική ώρα δημιουργίας του και joint columns με το id του σχετικού έργου καθώς και του σχετικού υποστηρικτή.
* Role.java: περιέχει ακέραιο αριθμό id, το όνομα του ρόλου καθώς και μέθοδο για την ανάθεση ρόλων στους χρήστες.
* ProjectUpdate.java: περιέχει ακέραιο αριθμό id, το περιεχόμενό του το οποίο είναι έως 1000 χαρακτήρες, μέθοδο createdAt η οποία αποθηκεύει την τοπική ώρα στην οποία δημιουργήθηκε η ενημέρωση και joint columns με το id του σχετικού έργου καθώς και του σχετικού δημιουργού.

## **Controller**

Ύστερα υπάρχει το πακέτο **controller** οι οποίοι. Συγκεκριμένα υπάρχει η κλάση MainController η οποία διαχειρίζεται την βασική περιήγηση, επιστρέφει τα templates και ελέγχει την πρόσβαση στα dashboards (πχ ADMIN only dashboards κ.λπ.).. Η κλάση PledgeController η οποία είναι για την διαχείριση των χρηματοδοτήσεων που έχουν προσφέρει οι υποστηρικτές. Η κλάση ProjectController η οποία είναι για την διαχείριση τον CRUD operations των πρότζεκτ καθώς και τις εγκρίσεις αυτών. Η κλάση ProjectUpdateController η οποία είναι υπεύθυνη για την διαχείριση των ενημερώσεων των πρότζεκτ καθώς και τις εγκρίσεις αυτών. Η κλάση UserController η οποία είναι υπεύθυνη για την διαχείριση τον διεργασιών για τους χρήστες. Τέλος η κλάση AuthController η οποία διαχειρίζεται την ταυτοποίηση των χρηστών, την εγγραφή τους, δίνει JWT tokens σε ταυτοποιημένους χρήστες, επικυρώνει τα διακριτικά των χρηστών και αναθέτει ρόλους κατά την εγγραφή.

* MainController: επιστρέφει το home page το οποίο είναι δημόσιο, το endpoint για το admin dashboard το οποίο είναι διαθέσιμο μόνο για τον χρήστη με το ρόλο του διαχειριστή και τα dashboards των δημιουργών και των υποστηρικτών αντίστοιχα τα οποία είναι διαθέσιμα για τους χρήστες οι οποίοι έχουν τους σχετικούς ρόλους.
* UserController: Περιέχει τα endpoints για την εύρεση όλων των χρηστών, την διαγραφή ενός χρήστη και την διαχείριση ενός χρήστη. Επίσης διασφαλίζει ότι αυτά είναι προσβάσιμα μόνο στον admin.
* AuthController: έχει το endpoint /signin για την ταυτοποίηση ενός χρήστη κατά την σύνδεσή του και την επιστροφή του JWT tokens και το /signup για την εγγραφή ενός νέου χρήστη τα οποία είναι δημόσια.
* ProjectUpdateController: παρέχει το endpoint για την προβολή όλων των ενημερώσεων ενός έργου το οποίο είναι διαθέσιμο και τα endpoints για τις διεργασίες των ενημερώσεων (πχ τη δημιουργία καινούργιας ενημέρωσης) τα οποία είναι διαθέσιμα μόνο σε χρήστες με το ρόλο του δημιουργού.
* ProjectController: έχει τα δημόσια endpoints για τα έργα τα οποία έχουν εγκριθεί καθώς και για τις πληροφορίες του, τα endpoints /pending, /approve, /reject τα οποία είναι διαθέσιμα μόνο στο διαχειριστή και είναι για την εύρεση των έργων τα οποία αναμένουν έγκριση καθώς και για την αποδοχή ή απόρριψη αυτής και παρέχει και τα POST endpoints για την δημιουργία νέων έργων από τους δημιουργούς τα οποία είναι διαθέσιμα μόνο για τους χρήστες με τον ρόλο του δημιουργού.
* PledgeController: παρέχει τα endpoints για την δημιουργία νέων pledge και για την προβολή των pledge που έχει κάνει ένας υποστηρικτής τα οποία είναι διαθέσιμα μόνο σε χρήστες με το ρόλο του υποστηρικτή, την διαγραφή ενός pledge το οποίο είναι διαθέσιμο μόνο στον διαχειριστή και για την προβολή των υποστηρίξεων ενός έργου το οποίο είναι δημόσιο.

## **Payload**

Ύστερα υπάρχει το πακέτο **payload** στο οποίο βρίσκονται οι κλάσεις LoginRequest SignupRequest οι οποίες χρησιμοποιούνται στα /api/auth/login και /api/auth/signup endpoints. Πιο συγκεκριμένα η LoginRequest είναι για την λήψη του κωδικού και του ονόματος ενός χρήστη όταν αυτός θέλει να συνδεθεί ενώ η SignupRequest είναι για την λήψη των δεδομένων ενός χρήστη όταν αυτός θέλει να εγγραφεί στη πλατφόρμα. Επίσης υπάρχουν και ακόμα δύο κλάσεις, η JwtResponse η οποία διαχειρίζεται την μεταφορά δεδομένων μεταξύ του frontend και του backend κατά την διάρκεια ταυτοποίησης και επιστρέφει τα αποτελέσματα μετά από μία επιτυχημένη ταυτοποίησης (JWT token, ρόλος χρήστη, όνομα χρήστη και email) και η MessageResponse η οποία επιστρέφει απλές ειδοποιήσεις για επιτυχίες ή σφάλματα (πχ για την επιτυχή εγγραφή ενός χρήστη).

* LoginRequest: δέχεται το όνομα χρήστη κα τον κωδικό του χρήστη.
* SignupRequest: δέχεται το όνομα χρήστη το οποίο πρέπει να είναι από 3 εώς 30 χαρακτήρες, το ονοματεπώνυμό του (έως 50 χαρακτήρες τόσο για το όνομα όσο και για το επίθετό του), το email του το οποίο πρέπει να έχει τη μορφή @Email (έως 50 χαρακτήρες), και τον κωδικό του οποίος πρέπει να είναι από 6 χαρακτήρες και μέχρι 40.

## **Repository**

Έπειτα υπάρχει το πακέτο **repository** τα οποία διαχειρίζονται την διασύνδεση του database με την εφαρμογή. Υπάρχει ένα repository για κάθε ένα από τα Entity classes. Για παράδειγμα υπάρχει UserRepository για το User Entity, αντίστοιχα και για τα υπόλοιπα.

* UserRepository: επιτρέπει την εύρεση ενός χρήστη με το username του καθώς και έλεγχο ύπαρξής του με το username του ή με το email του.
* PledgeRepository: επιτρέπει την εύρεση ενός pledge, είτε με το id του σχετικού έργου, είτε με id του χρήστη που το έκανε.
* ProjectRepository: επιτρέπει την εύρεση ενός έργου είτε με το id του δημιουργού χρήστη του, είτε με το εάν έχει εγκριθεί ή όχι.
* ProjectUpdateRepository: επιτρέπει την εύρεση μίας ενημέρωσης με τη χρήση του συ σχετιζόμενου έργου.
* RoleRepository: επιτρέπει την εύρεση ενός ρόλου με τη χρήση του ονόματός του.

## **Service**

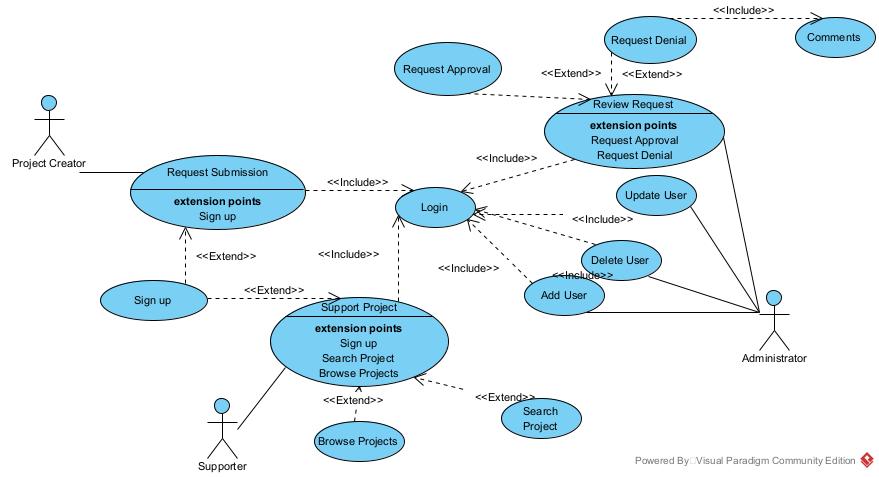
Τέλος, υπάρχει το πακέτο **service.** Σε αυτό μπορεί να βρεθεί η κλάση CreatorService η οποία δημιουργεί, διαγράφει, ανακτά τα προφίλ των δημιουργών και τα συνδέει με τους αντίστοιχους χρήστες. Η κλάση SupporterService λειτουργεί αντίστοιχα για τους υποστηρικτές. Η κλάση Project πέρα από την δημιουργία, διαγραφεί και ανάκτηση των πρότζεκτ, τα φιλτράρει ανάλογα με το εάν έχουν εγκριθεί ή εάν αναμένουν την έγκρισή τους. Η κλάση PledgeService διαχειρίζεται την ανάκτηση και την διαγραφή μίας υποστήριξης που μπορεί να προσφέρει ένας χρήστης σε κάποιο έργο ενώ ταυτόχρονα την επικαιροποιεί ελέγχοντας εάν το έργο έχει εγκριθεί και εάν έχει περάσει η προθεσμία λήξης του. Η UserDetailsServiceImpl χειρίζεται την ταυτοποίηση των χρηστών και φορτώνει τα στοιχεία τους για το Spring Security. Η InitialDataService αρχικοποιεί την βάση δεδομένων με κάποια αρχικά δεδομένα, δημιουργώντας κάποιους χρήστες με διάφορους ρόλους (1 διαχειριστή, 10 υποστηρικτές και 5 δημιουργούς), ενώ δημιουργεί και 2 έργα για τον κάθε δημιουργό. Τέλος, η UserDetailsImpl χρησιμοποιείται κατά την ταυτοποίηση του χρήστη και υλοποιεί τη διεπαφή UserDetails του Spring Security, μετατρέπει το User entity σε format το οποίο γίνεται κατανοητό από το Spring Security, αποθηκεύει στοιχεία του χρήστη για την ταυτοποίησή του, δηλαδή τα βασικά στοιχεία του (το id, το όνομα χρήστη του και το email του) , το ρόλο του και τον κρυπτογραφημένο κωδικό του για τον οποίο χρησιμοποιούμε @JsonIgnore για να μην επιστραφεί. Οι κλάσεις αυτές χρησιμοποιούν Transactional συναλλαγές για να εξασφαλίσουν την ακεραιότητα των συναλλαγών με την βάση δεδομένων και των δεδομένων.

Επίσης υπάρχει η κλάση SpringbootDemoApplication για η main της εφαρμογής. Επιπλέον, το αρχείο application.properties δίνει το βασικό configuration αυτής. Πιο συγκεκριμένα θέτει το configuration του server, της βάσης δεδομένων καθώς και της ταυτοποίησης JWT**.** Το server port του backend καθορίζεται ως το 9090.

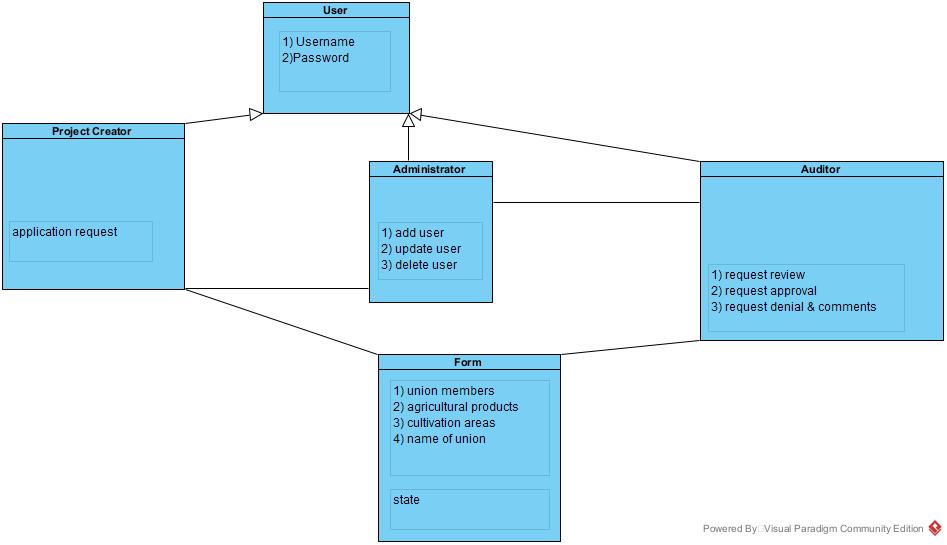
Για την δημιουργία του Docker container υπάρχει το αρχείο nonroot.Dockerfile και το αρχείο docker.Jenkinsfile για την δημιουργία του Docker image. Επίσης, το αρχείο docker-compose.yaml διασφαλίζει ότι η εκκίνηση της εφαρμογής θα γίνει με την σωστή σειρά (το Backend αναμένει την βάση δεδομένων, το Frontend αναμένει το Backend), δίνει το mapping για την βάση δεδομένων (port: 5432) και για το Backend (port:9090).

# **Διαγράμματα UML**

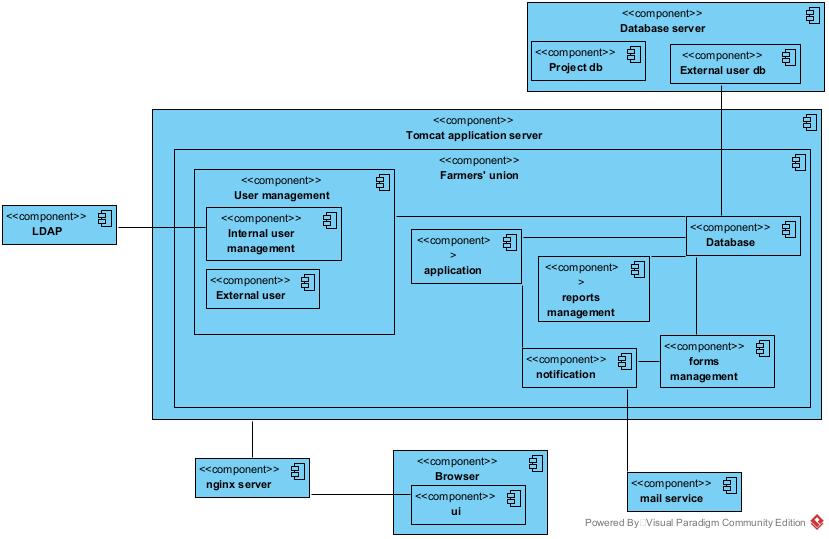
Διάγραμμα Use Case



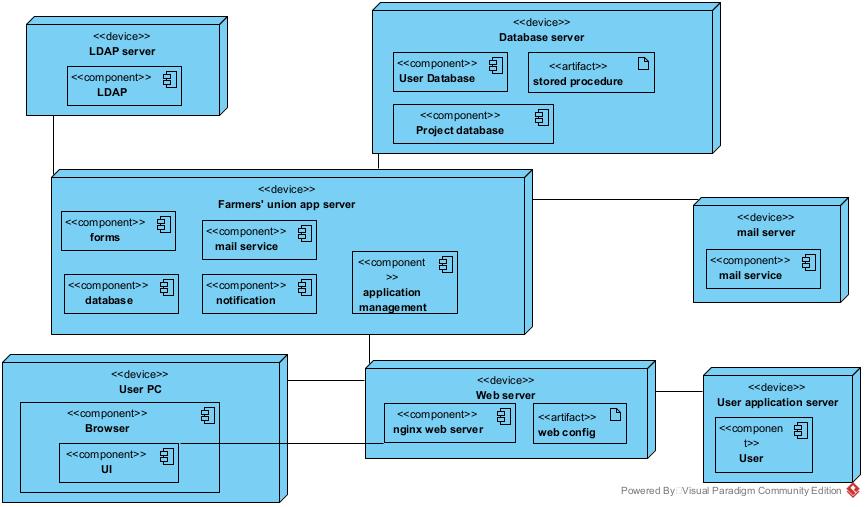
Class Diagram



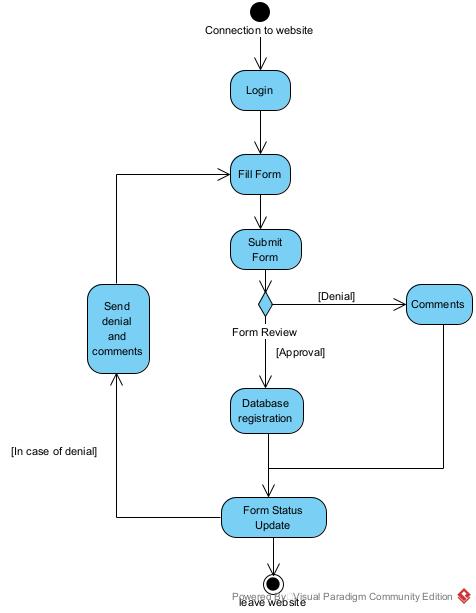
Component Diagram



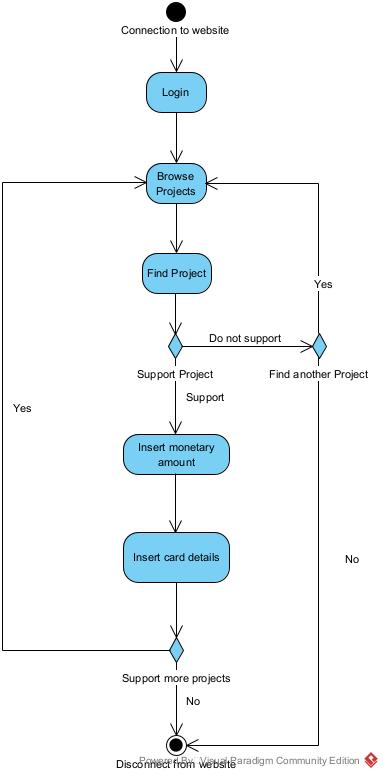
Deployment Diagram



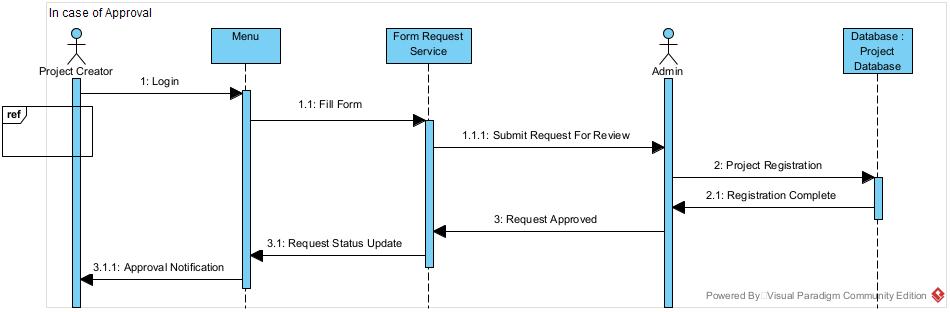
Activity Diagram: Creator



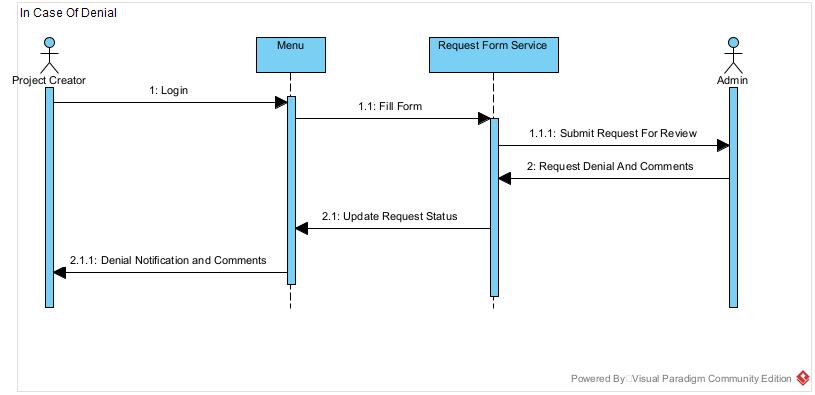
Activity Diagram: Supporter



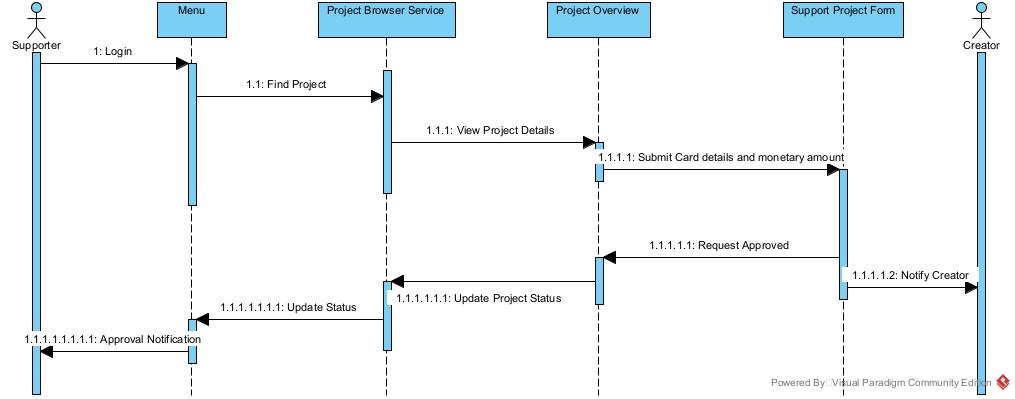
Sequence Diagram: Project Acceptance



Sequence Diagram: Project Rejection



Sequence Diagram: Supporter



# **Υλοποίηση του Frontend**

To Frontend υλοποιήθηκε με την χρήση του Vuejs.

Για την υλοποίηση του Frontend δημιουργήθηκαν:

* Ο router index.js ο οποίος παρέχει τα paths για τις σελίδες και πρόσβαση με βάση τους ρόλους όπου αυτό είναι απαραίτητο μέσω των μεθόδων
  + requiresSupporter
  + requiresCreator
  + requiresAdmin
* Το store application.js για την τοπική αποθήκευση των στοιχείων του χρήστη και του JWT token ταυτοποίησης
* Το main.js στο οποίο γίνεται η αρχικοποίηση του Vuejs, και το set up των plugins και των βιβλιοθηκών
* Το useRemoteData.js για την διαχείριση τον API calls, όπως η προσθήκη του JWT token σε αυτά
* Το App.vue το οποίο λειτουργεί ως ο wrapper της εφαρμογής
* Αρχεία views και components τα οποία θα αναλυθούν στην συνέχεια

## **Αρχεία Views**

* + - LoginView.vue: Είναι η σελίδα σύνδεσης η οποία δέχεται τα απαραίτητα στοιχεία και είτε ανακατευθύνει τον χρήστη στο HomeView είτε βγάζει μήνυμα σφάλματος. Επίσης υπάρχει κουμπί ανακατεύθυνσης στο SignupView
    - SignupView.vue: Είναι η σελίδα εγγραφής. Δέχεται τα στοιχεία του χρήστη και είτε ανακατευθύνει στη σελίδα σύνδεσης μετά από επιτυχία είτε βγάζει μήνυμα σφάλματος
    - HomeView.vue: Κεντρική σελίδα της ιστοσελίδας
    - ProjectListView.vue: είναι το ευρετήριο των έργων στο οποίο παρουσιάζονται
    - LogoutView.vue: σελίδα αποσύνδεσης
    - ProfileView.vue: σελίδα προβολής των στοιχείων του λογαριασμού του χρήστη
    - ProjectDetailsView.vue: Είναι η σελίδα παρουσίασης των στοιχείων ενός έργου. Οι δημιουργεί έχουν και πρόσβαση σε κουμπί για την δημιουργία νέας ενημέρωσης
    - CreateProjectView.vue: είναι η σελίδα για την δημιουργία νέων έργων. Δέχεται ως εισόδους τον τίτλο, την περιγραφή, τον στόχο χρηματοδότησης και το deadline ενός έργου.
    - AdminDashboardView.vue: Η σελίδα διαχείρισης του admin μέσο της οποίας μπορεί να διαχειρίζεται χρήστες και να εγκρίνει ή να απορρίπτει έργα
    - SupporterDashboardView.vue: Είναι η σελίδα στην οποία προβάλλονται τα pledges ενός υποστηρικτή καθώς και ειδοποίηση σε περίπτωση που έχει ενημερωθεί κάποιο από τα έργα που έχει υποστηρίξει
    - CreatorDashboardView.vue: Πρόκειται για την σελίδα μέσω της οποία ένας δημιουργός μπορεί να διαχειριστεί τα έργα του.
    - TermsOfServiceView.vue: μία απλή σελίδα για την προσομοίωση σελίδας η οποία θα παρουσίαζε τους όρους χρήσης της ιστοσελίδας
    - PrivacyPolicyView.vue: Σελίδα με γενικές πληροφορίες για την χρήση τον cookies κλπ.

## **Components**

* AppHeader: Header στην εφαρμογή μέσω του οποίου ο χρήστης έχει πρόσβαση στο HomeView.vue, στο LogoutView.vue, στο ProfileView.vue, στο ProjectListView.vue και στο ανάλογο dashboard ανάλογα με τον ρόλο του εκάστοτε χρήστη
* AppFooter: footer της εφαρμογής το οποίο περιέχει κάποιους γρήγορους συνδέσμους για το HomeView.vue, το ProjectListView.vue, το TermsOfServiceView.vue, το PrivacyPolicyView.vue, για τα κοινωνικά δίκτυα της ιστοσελίδας και πληροφορίες σχετικά με το copyright
* PledgeForm.vue: Pop up καρτέλα για την δημιουργία νέου pledge
* PledgeList.vue: component για την παρουσία των pledges στο SupporterDashboardView.vue
* ProjectCard.vue: Καρτέλα προεπισκόπησης βασικών πληροφοριών ενός έργου η οποία χρησιμοποιείται στο HomeView.vue και στο ProjectListView.vue
* ProjectTable.vue: πίνακας παρουσίασης έργων ο οποίος χρησιμοποιείται
* ProjectUpdate.vue: pop up καρτέλα για την δημιουργία ενημερώσεων έργων
* ProgressBar.vue: Μπάρα προόδου χρηματοδότησης η οποία χρησιμοποιείται στο ProjectCard.vue και στο ProjectDetailsView.vue για να δείξει πόσο ποσοστό % έχει συσσωρευθεί για την επίτευξη του στόχου χρηματοδότησης

# **Εγχειρίδιο χρήσης Frontend**

Login page περιέχει:

* Πεδίο συμπλήρωσης ονόματος ή email χρήστη “Email address or Username”
* Πεδίο συμπλήρωσης κωδικού χρήστη “Password”
* Κουμπί “Login” το οποίο χρησιμοποιείται μετά την συμπλήρωση των παραπάνω πεδίων για την σύνδεση
* Κουμπί “Signup” για ανακατεύθυνση στη σελίδα εγγραφής

Signup page περίεχει:

* Πεδίο για την συμπλήρωση του ονόματος χρήστη “Username”
* Πεδίο για την συμπλήρωση του μικρού ονόματος “First Name”
* Πεδίο για την συμπλήρωση του επιθέτου “Last Name”
* Πεδίο για την συμπλήρωση της ηλεκτρονικής διεύθυνσης “Email Address”
* Πεδίο για την συμπλήρωση του κωδικού με προειδοποίηση ότι ο κωδικός πρέπει να είναι από 6 έως 40 χαρακτήρες “Password”
* Πεδίο για την επαναληπτική συμπλήρωση του κωδικού για επιβεβαίωση ορθότητάς του “Confirm Password”
* Πεδίο επιλογής ρόλου “Role”. Οι διαθέσιμοι ρόλοι είναι “Supporter” και “Creator” ενώ ο προεπιλεγμένος είναι ο “Supporter”
* Κουμπί για την δημιουργία την εγγραφή του χρήστη “Sign Up”. Σε περίπτωση επιτυχημένης εγγραφής ο χρήστης ανακατευθύνεται στην σελίδα σύνδεσης αλλιώς παρουσιάζεται μήνυμα σφάλματος
* Κουμπί για την ανακατεύθυνση στη σελίδα σύνδεσης “Back to Login”

AppHeader, διαθέσιμο έπειτα από επιτυχημένη σύνδεση ενός χρήστη και περιέχει:

* Κουμπί για την ανακατεύθυνση στη κεντρική σελίδα του ιστότοπου “Crowdfunding Platform”
* Κουμπί για την ανακατεύθυνση στο ευρετήριο έργων “Browse Project”
* Κουμπί για την ανακατεύθυνση στη σελίδα προβολής στοιχείων του προφίλ “Profile”
* Κουμπί για ανακατεύθυνση στη σελίδα αποσύνδεσης “Logout”
* Κουμπιά ανακατεύθυνσης τα οποία είναι διαθέσιμα με βάση το ρόλο του χρήστη. Πιο συγκεκριμένα:

1. Κουμπί ανακατεύθυνσης στη σελίδα διαχείρισης του διαχειριστή “Admin Dashboard”
2. Κουμπί ανακατεύθυνσης στη σελίδα διαχείρισης έργων ενός δημιουργού “My Projects”
3. Κουμπί ανακατεύθυνσης στη σελίδα προβολής των pledges ενός υποστηρικτή “My Pledges”

AppFooter περιέχει:

* Μήνυμα με το σκοπό της ιστοσελίδας
* Πληροφορίες για το copyright
* Συνδέσμους:
  + - * 1. Για το Home page
        2. Για το Server Browser
        3. Για τη σελίδα των Privacy Policy
        4. Για την σελίδα Terms of Service
* Συνδέσμους επικοινωνίας κοινωνικών δικτύων

Home page περιέχει:

* Μήνυμα καλωσορίσματος στην σελίδα
* Κουμπί ανακατεύθυνσης μπλε χρώματος στο ευρετήριο έργων “Browse Projects”
* Κουμπί ανακατεύθυνσης στη σελίδα προφίλ του χρήστη γκρι χρώματος “My Profile”
* Κουμπί ανακατεύθυνσης στη σελίδα αποσύνδεσης κόκκινου χρώματος “Logout”
* Τομέα προεπισκόπησης των υπάρχων έργων “Featured Projects”

Logout page: δημιουργήθηκε για να επιβεβαιώσει ένας χρήστης ότι θέλει να αποσυνδεθεί. Περιέχει κουμπί αποσύνδεσης “Logout”.

Profile page: περιέχει καρτέλα η οποία παρουσιάζει τα στοιχεία του χρήστη “Account Details”. Το όνομα χρήστη του “Username”, την διεύθυνση του ηλεκτρονικού του ταχυδρομείου “Email” και τον ρόλο του “Role”.

Terms of Service περιέχει ένα απλό μήνυμα που επεξηγεί τον σκοπό της σελίδας καθώς βρίσκεται εκεί για τυπικούς λόγους.

Privacy Policy περιέχει πληροφορίες για:

* Τα δεδομένα που συλλέγονται
* Τη χρήση αυτόν τον δεδομένων
* Για το πως και με ποιον ενδέχεται να διαμοιραστούν αυτά τα δεδομένα
* Για την ασφάλεια των δεδομένων
* Για τα δικαιώματα του χρήστη σχετικά με τα δεδομένα
* Για την χρήση των cookies
* Για την επικοινωνία με τους υπεύθυνους της ιστοσελίδα για ερωτήσεις σχετικά με τη πολιτική απορρήτου

Project Browser περιέχει προεπισκόπηση έργων με:

* Τίτλο του έργου
* Περιγραφή του έργου
* Μπάρα ποσοστού προόδου χρηματοδότησης
* Τον στόχο της χρηματοδότησης
* Το τρέχων ποσό της χρηματοδότησης
* Κουμπί μπλε χρώματος για ανακατεύθυνση στη σελίδα πληροφοριών του έργου
* Κουμπί πράσινου χρώματος για την δημιουργία pledge το οποίο είναι διαθέσιμο μόνο σε χρήστης με το ρόλο του υποστηρικτή. Με το πάτημά του εμφανίζεται pop up καρτέλα στην οποία υπάρχει:

1. Πεδίο με το τίτλο του έργου “Project”
2. Πεδίο συμπλήρωσης ποσού προς δωρεά καθώς “Amount ($)”
3. Ένδειξη εναπομείναντος ποσού μέχρι την επίτευξη του στόχου χρηματοδότησης “Remaining goal: xx”
4. Κουμπί για ακύρωση της διαδικασίας “Cancel”
5. Κουμπί για την αποστολή της υποστήριξης “Submit Pledge”

Project Details περιέχει:

* Κουμπί επιστροφής στο ευρετήριο έργων “Back to Projects”
* Το τίτλο του έργου
* Καρτέλα με την περιγραφή του έργου “Project Description”
* Καρτέλα “Funding Progress” με την μπάρα προόδου της χρηματοδότησης, την τωρινή χρηματοδότηση “Current Funding $X”, τον στόχο χρηματοδότησης “Funding Goal $X” και τις μέρες που απομένουν μέχρι το Deadline “Time Left X days”
* Καρτέλα με τις ενημερώσεις του έργου “Project Updates”
* Καρτέλα με τα πρόσφατα pledges “Recent Pledges”
* Διαθέσιμο μόνο στο δημιουργό του έργου, κουμπί δημιουργίας ενημέρωσης “New Update”. μόλις πατηθεί εμφανίζει pop up καρτέλα στην οποία υπάρχει:

1. Πεδίο συμπλήρωσης του κειμένου της ενημέρωσης
2. Κουμπί για την ακύρωση της διεργασίας “Cancel”
3. Κουμπί για την αποστολή της ενημέρωσης “Post Update”

Create Project περιέχει:

* Κουμπί ανακατεύθυνσης στο ευρετήριο έργων “Back to Projects”
* Πεδίο συμπλήρωσης τίτλου “Project Title”
* Πεδίο συμπλήρωσης περιγραφής με ένδειξη για την κατανόηση της συμπλήρωσης της περιγραφής “Description”
* Πεδίο συμπλήρωσης στόχου χρηματοδότησης “Funding Goal ($)”. Ο χρήστης μπορεί είτε να τον πληκτρολογίσει είτε να τον προσαρμόσει χρησιμοποιόντας τα δύο βελάκια στο δεξί μέρος του πεδίου
* Πεδίο συμπλήρωσης ημερομηνίας Deadline “Project Deadline”. Ο χρήστης μπορεί είτε να την πληκτρολογήσει είτε να την επιλέξει χρησιμοποιώντας το ημερολόγιο στο δεξί μέρος του πεδίου. Περιέχει ένδειξη που ενημερώνει τον χρήστη ότι η ημερομηνία πρέπει να είναι τουλάχιστον 7 ημέρες από την τρέχουσα ημερομινία.
* Κουμπί ακύρωσης της διεργασίας “Cancel” το οποίο ανακατευθύνει τον χρήστη στο ευρετήριο των έργων
* Κουμπί ολοκλήρωσης της διεργασίας και δημιουργίας του έργου “Create Project”

My Projects, είναι το Creator Dashboard διαθέσιμο μόνο σε δημιουργούς και περιέχει:

* Κουμπί ανακατεύθυνσης στη σελίδα δημιουργίας νέου έργου
* Καρτέλες προεπισκόπησης των έργων του δημιουργού οι οποίες περιέχουν:

1. Το τίτλο του έργου
2. Την περιγραφή του έργου
3. Την μπάρα προόδου χρηματοδότησης
4. Το ποσό χρημάτων που έχουν λάβει “$X raised”
5. Τον στόχο χρηματοδότησης “$X goal”
6. Πράσινη ένδειξη σε περίπτωση που έχουν εγκριθεί “Approved”
7. Κίτρινη ένδειξη σε περίπτωση που έχουν εγκριθεί “Pending Approval”
8. Κουμπί μετάβασης στη σελίδα του έργου “View”
9. Κουμπί κόκκινου χρώματος για την διαγραφή του έργου “Delete”

My Pledges, είναι το Supporter Dashboard διαθέσιμο μόνο σε υποστηρικτές και περιέχει καρτέλα με τα pledges που έχει κάνει ο χρήστης. Για το κάθε Pledge παρουσιάζει:

* Το όνομα του χρήστη
* Το ποσό της δωρεάς
* Το τίτλο του σχετικού έργου, ο οποίος μπορεί να πατηθεί για να ανακατευθύνει τον χρήστη στη σελίδα πληροφοριών του έργου
* Την ημερομηνία την οποία πραγματοποιήθηκε η προσφορά
* Ένδειξη καμπανάκι δίπλα από τον τίτλο ενός έργου εάν αυτό το έργο έχει λάβει ενημερώσεις, το οποίο μπορεί να πατηθεί για την ανακατεύθυνση του χρήστη στη σελίδα πληροφοριών του έργου

Admin Dashboard, είναι σελίδα διαθέσιμη μόνο σε διαχειριστές και περιέχει:

* Καρτέλα με μήνυμα καλωσορίσματος
* Κουμπί “Users” με το πάτημα του οποίου παρουσιάζεται η διαχείριση των χρηστών η οποία περιλαμβάνει:

Τον αριθμό των χρηστών “Total users: X”

Πίνακα στον οποίου παρουσιάζονται οι χρήστες και ο οποίος περιέχει:

Το όνομα του χρήστη “Username”

Το μικρό του όνομα “First Name”

Το επίθετό του “Last Name”

Την διεύθυνση του ηλεκτρονικού του ταχυδρομείου “Email”

Τον ρόλο του “Role”

Κουμπί διαγραφής του χρήστη “Delete”. Χρήστε με το ρόλο του διαχειριστή δεν έχουν αυτή την επιλογή. Σε περίπτωση δημιουργού η διαγραφή τους διαγράφει και όλα τα έργα του εν λόγο χρήστη.

Κουμπί για την επεξεργασία του χρήστη με το πάτημα του οποίου παρουσιάζεται καρτέλα η οποία περιέχει

Α) Το όνομα χρήστη “Username”

Β) Πεδίο επεξεργασίας της ηλεκτρονικής του διεύθυνσης “Email”

Γ) Πεδίο αλλαγής του ρόλου του χρήστη “Role”

Δ) Κουμπί για την ακύρωση της διεργασίας “Cancel”

Ε) Κουμπί για την ολοκλήρωση της διεργασίας και αποθήκευσης των αλλαγών “Save Changes”

* Κουμπί “Projects” με το πάτημα του οποίου παρουσιάζεται η διαχείριση των έργων η οποία περιλαμβάνει:

1. Πίνακα “Pending Projects” με καρτέλες των έργων τα οποία αναμένουν επιβεβαίωση. Η κάθε καρτέλα περιέχει

Το τίτλο του έργου

Την περιγραφή του

Κουμπί πράσινου χρώματος “Approve” με το πάτημα του οποίου εμφανίζεται pop up παράθυρο το οποίο του ζητάει να επιβεβαιώσει την έγκριση. Με το πάτημα του κουμπιού «ΟΚ» εγκρίνεται το έργο

Κουμπί κόκκινου χρώματος “Reject” με το πάτημα του οποίου εμφανίζεται pop up παράθυρο το οποίο του ζητάει να επιβεβαιώσει την απόρριψη. Με το πάτημα του κουμπιού «ΟΚ» απορρίπτεται το έργο και διαγράφεται

1. Πίνακα “Approved Projects” με τις καρτέλες των έργων τα οποία έχουν εγκριθεί. Η κάθε καρτέλα περιέχει το τίτλο και την περιγραφή του έργου