

ΟΜΑΔΑ 1 [*CROWDFUNDING APPLICATION*]

- Μέλη ομάδας:
 - ✓ Αποστολάκης Δημήτριος aka it2022004
 - ✓ Αποστόλου Ορέστης aka it2022006
 - ✓ Ζαπάντης Νικόλαος aka it2022019



GITHUB REPO [LINK]

- Μπορείτε να δείτε τον κώδικα του project μας στο παρακάτω link (περιέχει εκεί και manual χρήσης για το front-end):

[Github Repository \[Crowdfunding application\]](https://github.com/NikosZapantis/DS_crowdfunding) ή

https://github.com/NikosZapantis/DS_crowdfunding

- Για να δείτε τον κώδικα του backend (Controllers / Entities etc.) ακολουθήστε το παρακάτω path:

[crowdfunding/src/main/java/gr/hua/dit/ds/crowdfunding](#)

- Για να δείτε τον κώδικα του frontend (Html-Javascript-CSS) ακολουθήστε το παρακάτω path:

[crowdfunding/src/main/resources/src \(front-end\)](#)

COMMANDS

- Εντολές εκτέλεσης παραδοτέου

```
mvn clean package
```

```
java -jar .\target\crowdfunding-0.0.1-SNAPSHOT.jar [start]
```

```
Ctrl + C [Stop]
```

ADMIN CREDENTIALS

- Username: admin
- Password: admin

SWAGGER DOCUMENTATION

- Μετά από επιτυχή σύνδεση το swagger επιστρέφει αυτό:

```
Responses
```

```
Curl
```

```
curl -X 'POST' \  
  'http://localhost:8080/api/auth/signin' \  
  -H 'accept: */*' \  
  -H 'Content-Type: application/json' \  
  -d '{  
    "username": "admin",  
    "password": "admin"  
  }'
```

Request URL

```
http://localhost:8080/api/auth/signin
```


Server response

Code	Details
200	<p>Response body</p> <pre>{ "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IWRXbWVzIFJpbey3dhk0IjSTbFVTRVlFV0slnldCIGMTczNjg5MDQxOClZIXhwTjoxNzK2OTc2ODE4fQ.XCLsg3ntqK8tTE_3nEndyHm9cNEPzfURGF28vtUIMc", "id": 1302, "username": "admin", "email": "it2022004@hua.gr", "roles": ["ROLE_ADMIN", "ROLE_USER"], "tokenType": "Bearer" }</pre>

- Από το οποίο πρέπει να κάνετε copy αυτό:

Response body

```
{
  "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTJ0eS9kXRo3bDphdHkiOiJST0xFT0tneS9kXRo3bDphdHkiOiJST0xFTVIRVlFV08taGlhYm9kaWZlcjEwLjE5OTQ0MTA0MTEyZWYhbnR3CnEpczFuOGF28vTUInc",
  "id": 1393,
  "username": "admin",
  "email": "it2022004@hua.gr",
  "roles": [
    "ROLE_ADMIN",
    "ROLE_USER"
  ],
  "tokenType": "Bearer"
}
```

 Download

- Η αυθεντικοποίηση γίνεται ως εξής:

The screenshot shows the Swagger UI interface. At the top, there's a 'Servers' section with a dropdown menu showing 'http://localhost:8080 - Generated server url'. Below this, the 'project-controller' section is expanded, showing a list of endpoints. A modal dialog titled 'Available authorizations' is open in the center, displaying 'bearerAuth (http, Bearer)' and a text input field with the placeholder 'paste here'. The 'Authorize' button is highlighted with a red circle and an arrow pointing to it.

- Τα παραπάνω βήματα είναι απαραίτητα για αυτό το version της εφαρμογής (μόνο back-end) και στο version που θα υπάρχει και front-end θα γίνονται αυτόματα.

ΒΑΣΙΚΕΣ ΠΑΡΑΔΟΧΕΣ ΕΡΓΑΣΙΑΣ

1. Κάθε έργο (fundraiser) έχει ακριβώς έναν δημιουργό.
2. Κάθε έργο έχει προκαθορισμένο κύκλο ζωής με βάση τα παρακάτω:
 - Pending → Αναμένει έγκριση από τον Admin.
 - Active → Έχει εγκριθεί από τον Admin και βρίσκεται στην διάθεση του κοινού.
 - Stopped → Έχει μπει σε παύση από τον Admin και μπορεί να ενεργοποιηθεί ξανά μόνο από αυτόν.
 - Completed → Έχει ολοκληρωθεί (ολοκληρώθηκε το deadline).
3. Ένα έργο παραμένει active μέχρις ότου να ολοκληρωθεί το deadline. Αυτό σημαίνει πως μπορεί να επιτύχει τον στόχο που έχει τεθεί ή και να τον ξεπεράσει.
4. Ο λογαριασμός που ανατίθεται στον Admin της εφαρμογής δημιουργείται στο initialization μία και μόνο φορά και δεν μπορεί ούτε να αλλάξει ούτε να δημιουργηθεί και άλλος (hardcoded).
5. Κάθε χρήστης μπορεί να κάνει submit ένα και μόνο report σε κάθε ξεχωριστό έργο και αν επιθυμεί να καταχωρήσει καινούργιο report το ακριβώς προηγούμενο αντικαθίσταται και αποθηκεύονται τα καινούργια στοιχεία, για το οποίο ενημερώνεται πριν επιχειρήσει να το καταχωρήσει.
6. Η δωρεά έχει κατώτατο όριο 1 ευρώ και το κάθε έργο έχει κατώτατο όριο στόχου 100 ευρώ.
7. Το κάθε έργο έχει κατώτατο deadline για ένα μήνα μετά την δημιουργία του.
8. Όταν πραγματοποιείται μια δωρεά από τον χρήστη, του εμφανίζεται κατάλληλο notification ότι πραγματοποιήθηκε με επιτυχία η συναλλαγή χωρίς να υπάρχει implementation με κάποιο API τράπεζας όπως θα γινόταν σε μια πληρέστερη έκδοση της εφαρμογής μας.

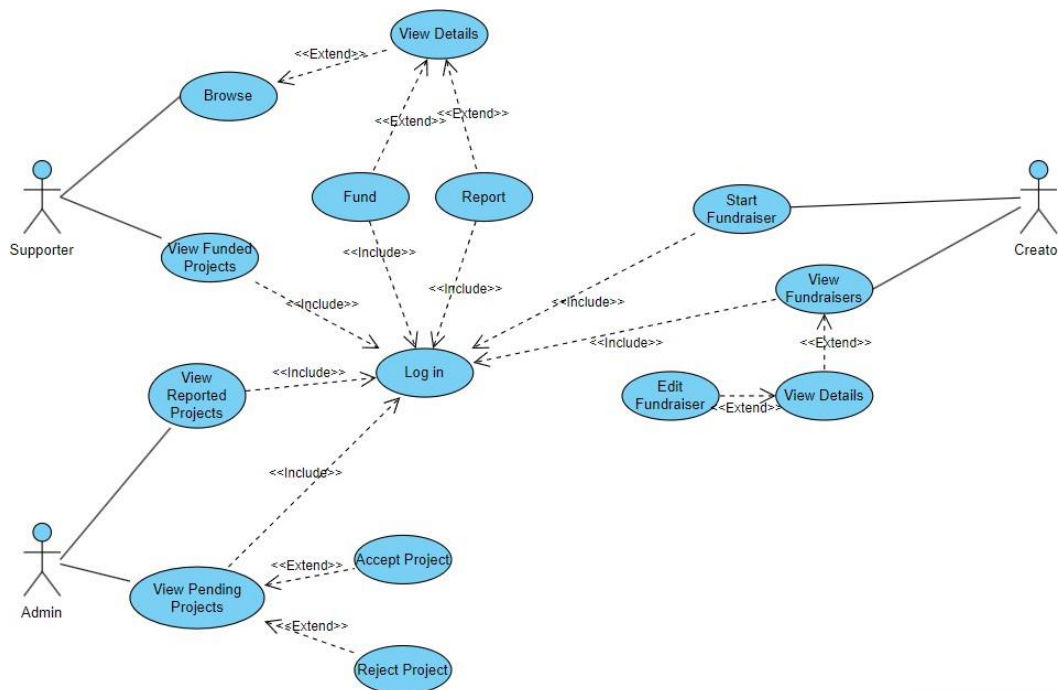
Εργασία Κατανεμημένα Συστήματα (CrowdFunding)

- Το συγκεκριμένο report αφορά την **ομάδα 1** με τα εξής μέλη:
 - Ζαπάντης Νικόλαος με **it2022019**
 - Αποστολάκης Δημήτριος με **it2022004**
 - Αποστόλου Ορέστης με **it2022006**

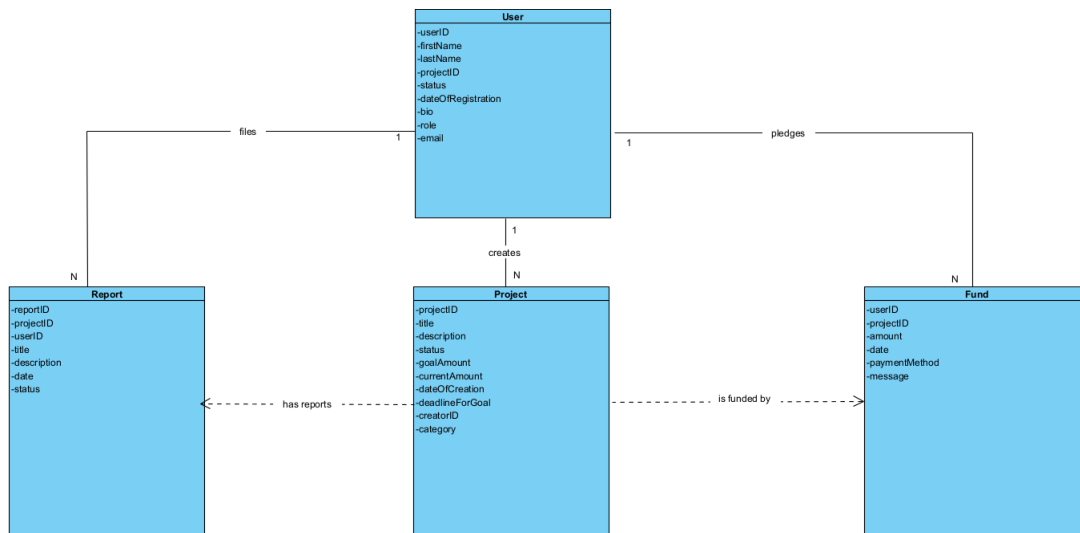
Οι παραδοχές που έχει η εργασία μας είναι οι εξής:

- I. Κάθε ξεχωριστό έργο, fundraiser, έχει ακριβώς έναν δημιουργό.
- II. Οι πληρωμές, που υποθετικά πραγματοποιούνται εντός του crowdfunding, γίνονται από services που δεν θα συνδέσουμε με την εργασία μας, όπως για παράδειγμα ένα API μιας τράπεζας.
- III. Εντός της εφαρμογής δεν είναι δυνατή η αναφορά **(report)** χρηστών από άλλους χρήστες.
- IV. Οι supporters και οι creators έχουν τα ίδια δικαιώματα εντός του συστήματος.

✓ To use case diagram του crowdfunding:



✓ To class diagram tou crowdfunding:



✓ To component diagram tou crowdfunding:

