User Stories - user (all types)

- As a **user** I would like to be able to:
  - Be able to **access** the platform (**login** and/or **register**) with **google account**
  - Setup a **profile**, comprised of **email, phone number, age, last job description, last graduation level, gender, location, profile description, domain** and **name**.
  - **View** my profile
  - **update** data
  - **View** offer in a **popUp data** when click on a offer in **main page**
  - Choose if I'm a **recruiter** or a **candidate** (next slides)
  - **View** offers from market

User Stories - admin

- Admin:
    - **View** users **accounts**
    - **Filter accounts** by **type** (recruiter or candidate)
    - **Delete** users accounts
    - **Filter offers** by **category** or **view verified/unverified** offers
    - **Delete, update** and **verify** an offer

User Stories - recruiter

- **Recruiter** profile would like to be able to:
  - **Filter** offers by **category** or view **my** offers
  - **View** my profile
    - See data about me
    - See list of offers posted by me and be able to see a extended page with data about post and a list of applicant
  - **Post** offert to market page
  - **Update/Delete** my offers
  - **Mark** an offer **as expired**
    - By pressing a button from popUp or form extended page, after expire in the corner on extended page a red icon appear
  - **View offer detail page**
  - **View candidates** that applied to my offers
  - **Change** or update the **application's status**

User Stories - candidate

- **Candidate** profile would like to be able to:
    - **Filter** offers by **category** and by **status (accepted or rejected)**
    - **Apply to** offer on market page
    - **Search** by offert's name
    - **View** my application **status**
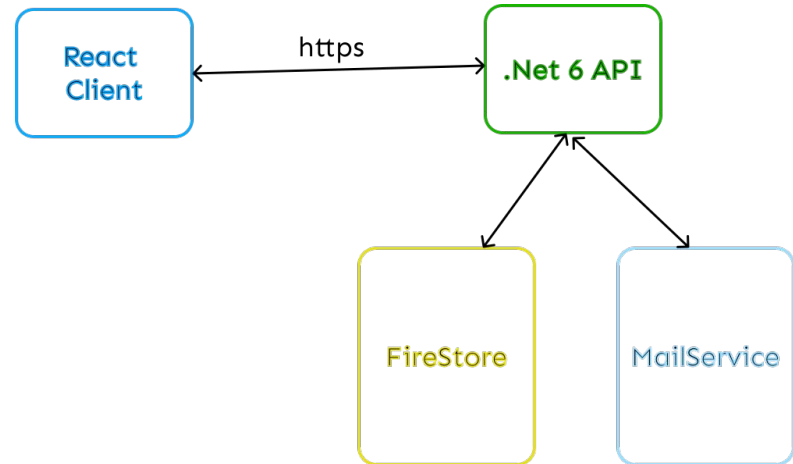   - **Receive email** when **status** of application **changes**

Summary of the technologies used

Frontend - React with Typescript (with NSwag generator for backend-frontend integration)

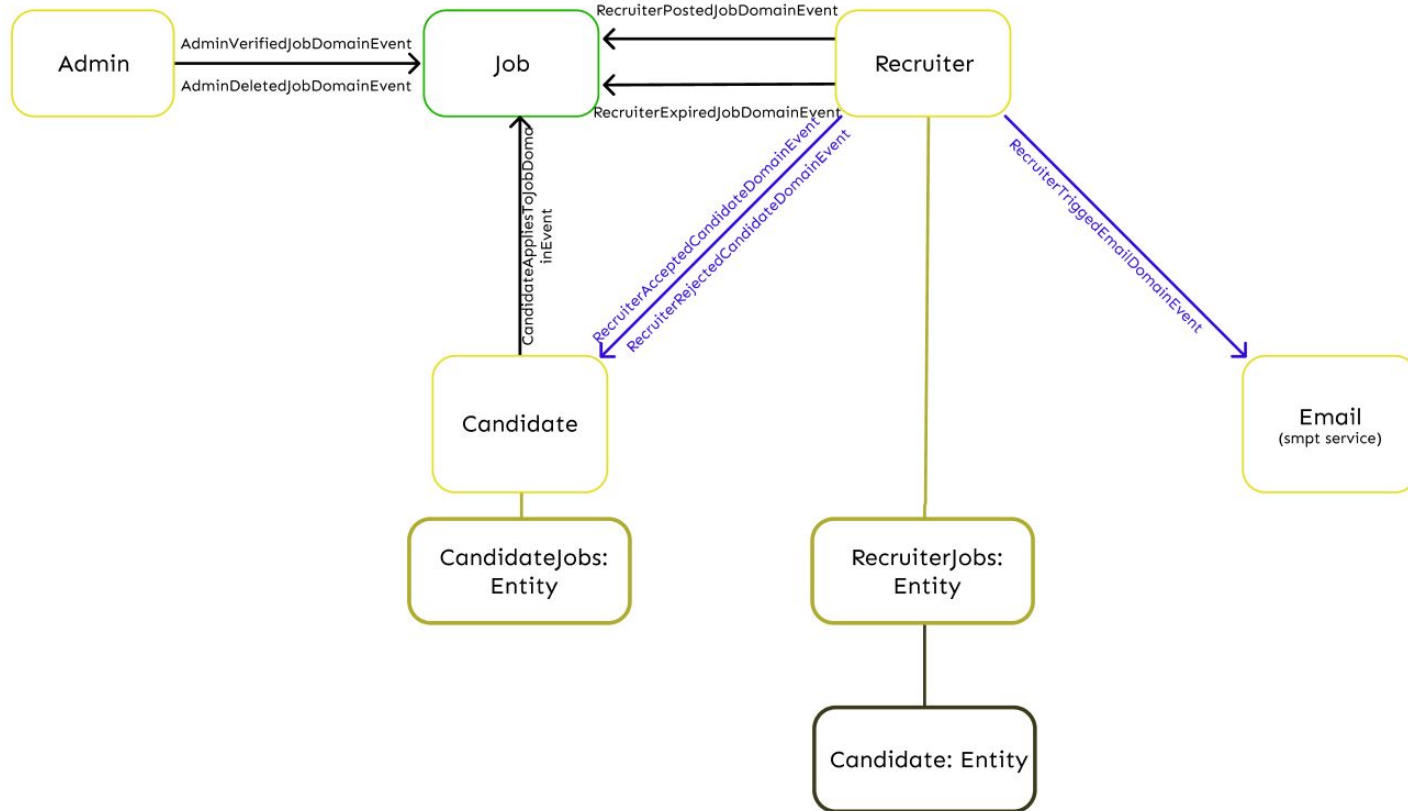Backend - .Net 6 API (with swagger)

DataBase - Firestore (for on demand data)

Authentication: Firebase Auth (for a better integration with firestore)

Entities ([Figma](#))

Arhitectura macro (la nivel de sistem)

Entities ([Figma]())

Arhitectura de domeniu la nivel de date

Admin
- documentId: string
- email : string
- name: string
- phone: string
- location : string
- description: string
- domain: string
- type : string
- isAdmin : bool

AdminVerifiedJobDomainEvent

AdminDeletedJobDomainEvent

Job
- documentId: string
- name: string
- numbrEmp: int
- numberApplicants: int
- address : string
- description: string
- domain: string
- isCheck: bool
- isExpired: bool
- recruiterID : string
- Date : dateTime

Recruiter
- documentId: string
- email : string
- name: string
- phone: string
- location : string
- description: string
- domain: string
- type : string
- isAdmin : bool
- jobList : List<RecruiterJobs>

RecruiterPostedJobDomainEvent

RecruiterExpiredJobDomainEvent

RecruiterAcceptedCandidateDomainEvent

RecruiterRejectedCandidateDomainEvent

RecruiterTriggeredEmailDomainEvent

CandidateAppliesToJobDomainEvent

Email
(smpt service)
- username: string
- emilAddress: string
- subject: string
- emailBody: string

Candidate
- documentId: string
- email : string
- name: string
- phone: string
- location : string
- gender: string
- description: string
- domain: string
- type : string
- age: int
- last_level_grad: string
- description_last_job: string
- isAdmin : bool
- jobList : List<CandidateJobs>

**CandidateJobs:
Entity**
- anuntId: string
- candidateID: string
- status : int
- apply_date: Datetime
- jobDetails: job

**RecruiterJobs:
Entity**
- documentId: string
- jobId: string
- recruiterID: string
- candidateList: List <CandidateJob>

**CandidateJob:
Entity**
- documentId: string
- conidateId: string
- jobId: string
- status : int
- candidate: user
- applyDate: dateTime
- lastStatusDate: dateTime