Recruitment Manager
Proposal

Problem statement: Currently the administration of the recruitment process is done through Trello, in this tool the profile of the applicant that the client requires is defined and from there the recruitment process begins, once an applicant is found goes through a series of tests or interviews that are finally uploaded to Trello, the interview is requested to one of the interviewers, This process is done by slack and then a Google-Meet is scheduled with the candidate, in this meeting (interview) is where the technical knowledge is evaluated and a standard format (SQUARE) is filled out for all senioritis and at the end a concept is issued, at this point the interviewer is not clear what the vacancy is looking for or what should be emphasized, this information must be transcribed to the trello board to keep track.

Overview: Design a web application to manage the recruitment process from the opening of the vacancy to the same recruitment process. This application will enter the specific vacancies of each client, the necessary and additional skills, seniority and a comment to clarify any aspect of the vacancy, the application will connect with Trello and create the tracking card (This process will be additional to maintain compatibility with other departments that use Trello's information), each vacancy will have a process flow 1. Find the aspirant, 2. Perform English test (Example), 3. Conducting a technical interview, 4. etc. When you find the first candidate, you must upload the basic information to the app and schedule the necessary interviews with the interviewers according to the task, this process will send an email to the interviewer and schedule the meeting on Google-Meet, for the technical interview, the interviewer will receive the email with the specific link of the app where you will find a specification of the vacancy and the candidate's data, additionally the topics that must be validated according to the skills sought and seniority, additionally the candidate is shared a link to the application where he will find a questionnaire with very specific questions that allow to validate the knowledge in a cognitive way, this questionnaire will be answered online at the time of the interview the interviewer can show the questions according to each and to evaluate procedural knowledge with the interview, the procedural assessment will be based on very clear and objective headings (does not have the knowledge, needs to improve, has the knowledge, exceptional). Both the cognitive and procedural assessment will remain in the application, once the interviewer is finished the interviewer should comment or add questions that he believes are relevant and will be added to the assessment (Additional questions go through a validation process and are then included in the corresponding topic), and close the interview, this process sends an email to the interviewer informing that the interview is over and the interviewer will be able to consult the assessment, the app will automatically sort the applicants according to their assessments, the client will have access to the app and will be able to follow the recruitment process and observe the interviews and evaluations carried out. and choose your candidate. The vacancy closes once the recruitment is done.

Targets:

1. Design the software architecture.
2. Design the data model.
3. Develop an API that provides the services of:
    a. Create, modify and consult Vacancies.
    b. Create, modify and consult Skills.
    c. Create, modify and consult Topics.
    d. Create, modify and consult Questions.
    e. Create, modify and consult Applicants.
    f. Create, modify and consult Rubrics.
    g.
    h. Upload applicant's documents.
    i. Add applicant to vacancy.
    j.

Architecture:

A 7-layer architecture is proposed:

Services Layer: It is in charge of providing the access point to the different proposed services.

Business Layer (Core): This layer is in charge of the business logic, defines the use cases and models the behavior.

Entity Layer: This layer is in charge of defining the data model (the entities) and the data exchange objects (Dtos).

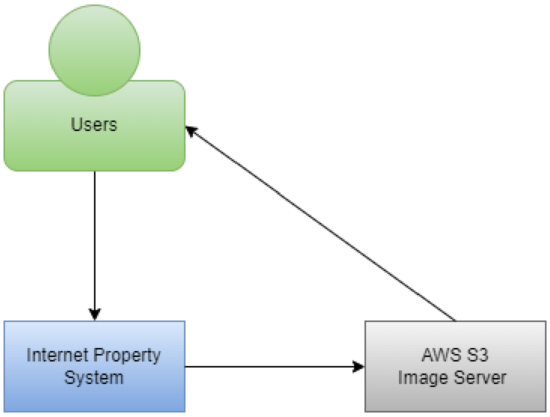
Contracts Layer: This layer is in charge of defining the contracts required for data access and allows the injection of dependencies.

Repository Layer: This layer is responsible for connecting to the data through the Repository pattern creates total independence from the data layer, allowing the data engine to be changed without the business layer being aware of it.

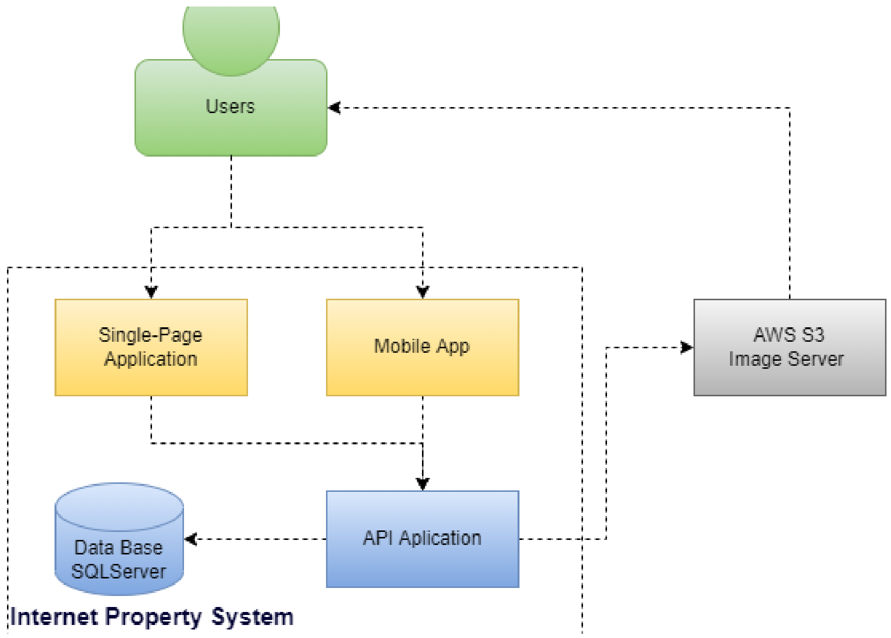
Data Access Layer: This layer is responsible for connecting to the database and providing the mechanisms for accessing, inserting, updating, deleting and querying data.

Presentation Layer: This layer is in charge of the user interface, it has total independence, it can be created in any technology (React, Angular, Vue, Blazor, etc).

Context Diagram.



Container Diagram:



Components Diagram:

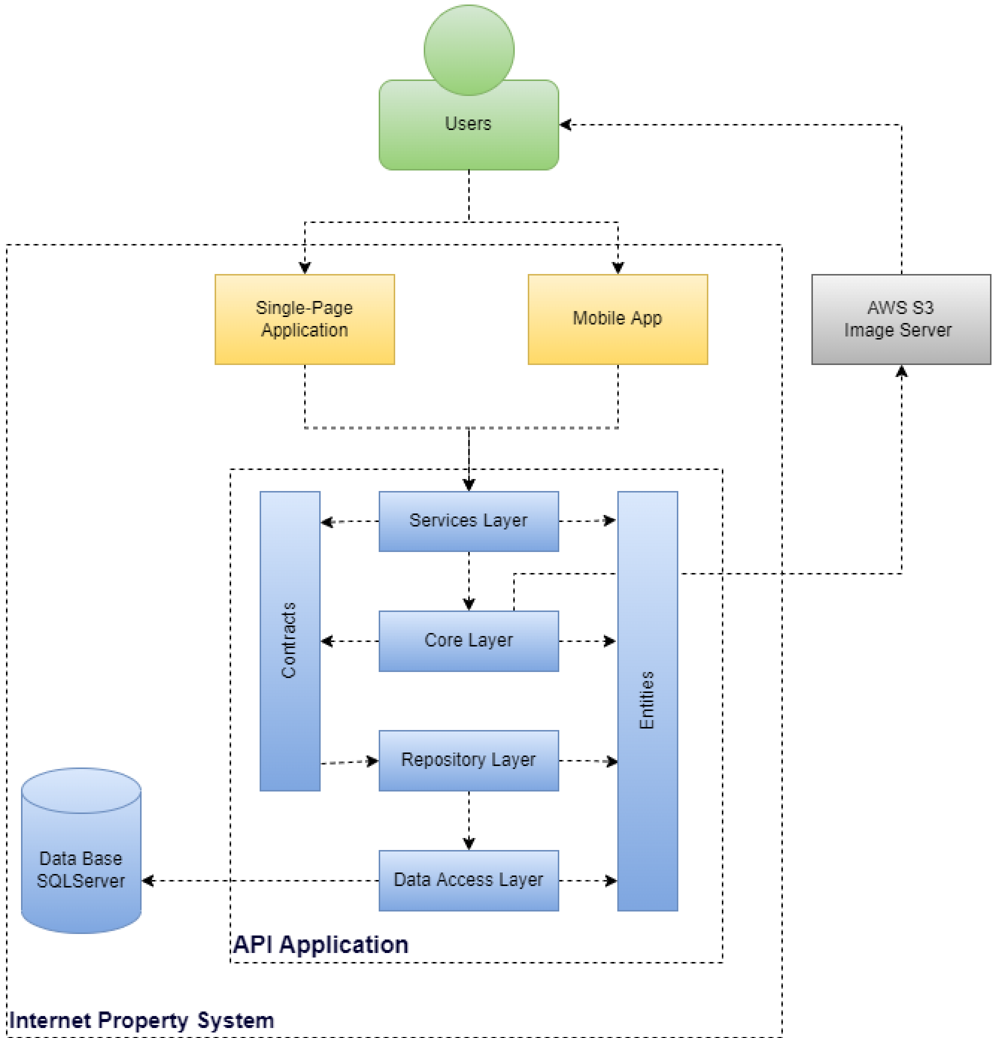


Diagram Process:

Data Model:

Views: