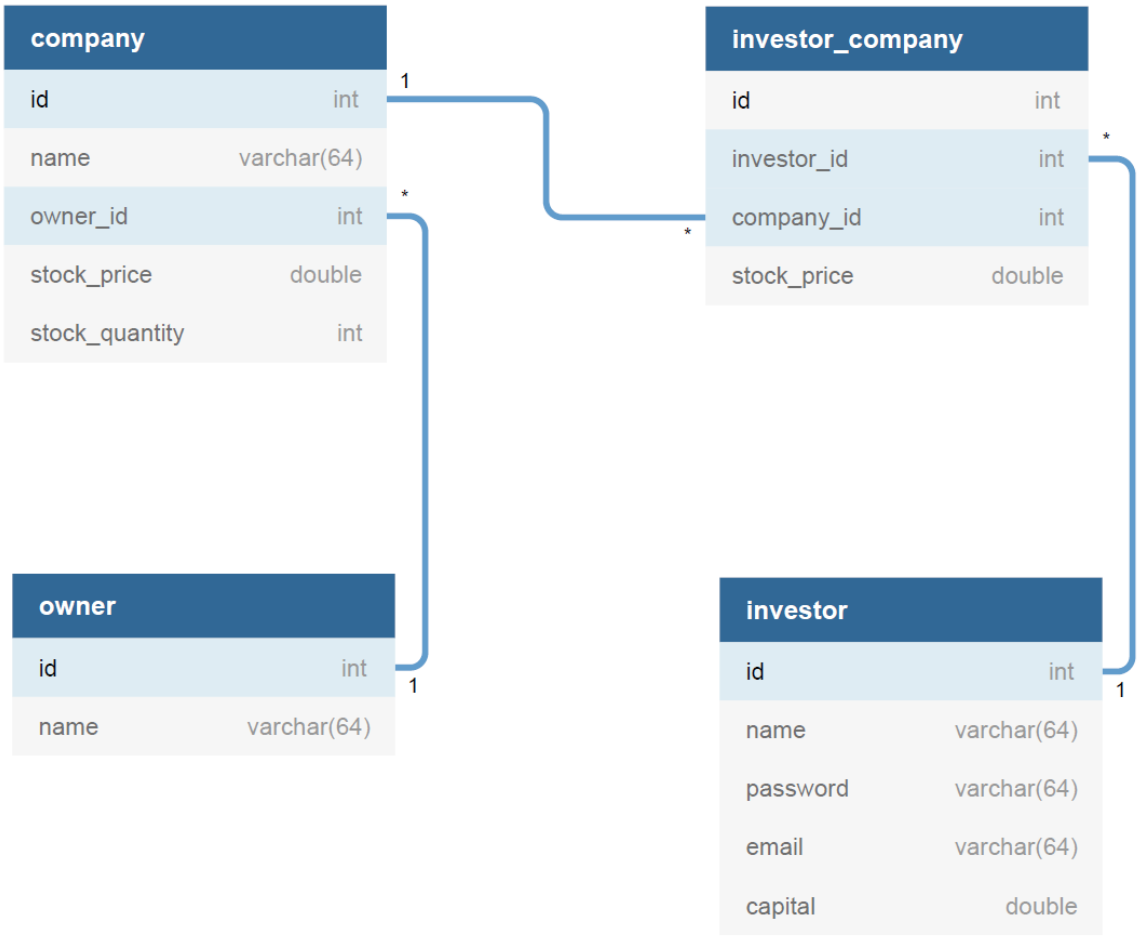
Martuneac
Alexandru
30431

Stock market platform

Analysis and Design Document

I. Elaboration – Iteration 1.1

Domain Model



Architectural Design

Conceptual Architecture

For the back-end, a layered architecture will be implemented to better organize and differentiate between entities, services, business logic, etc. Furthermore, abstract factory (creational pattern) will be present, since it allows to easily switch between the concrete implementation of the objects instantiated. Also, it is easier to write, understand, test and extend.

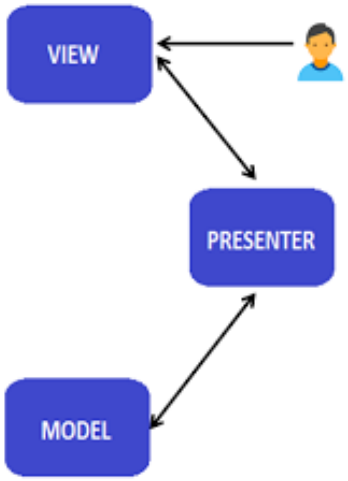
Observer pattern will be implemented to announce the investors that a new company was listed.

Abstract factory pattern for creating families of related objects without specifying their type. This allows us to change the implementation of different classes, in case a new technology appears or we have to use multiple implementations.

The front-end will use a variation of the famous MV* design pattern, more specifically MVP (model-view-presenter).

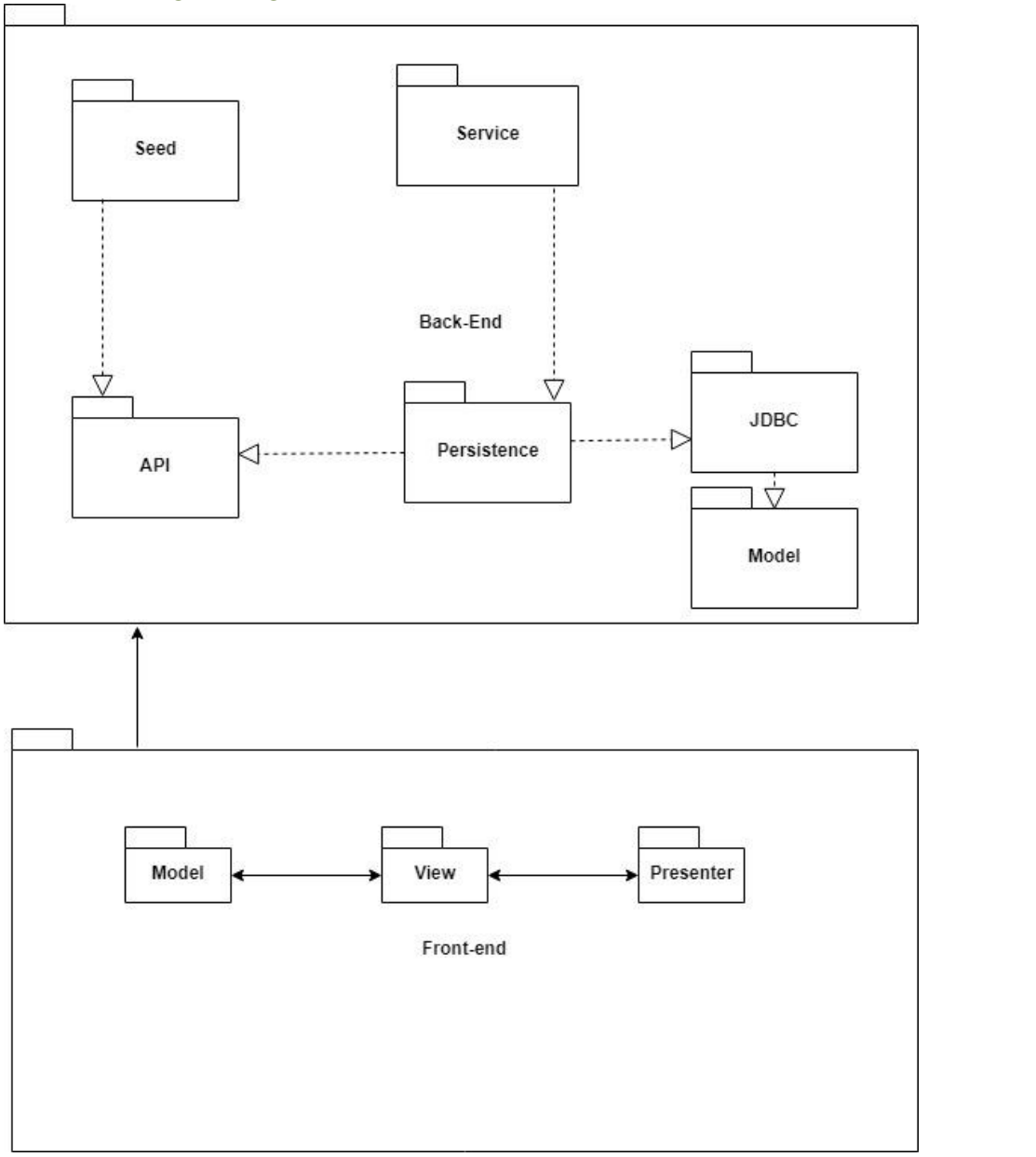
The Presenter receives the input from users via View, then process the user's data with the help of Model and passing the results back to the View.

Presenter communicates with view through interface. Interface is defined in presenter class, to which it passes the required data.

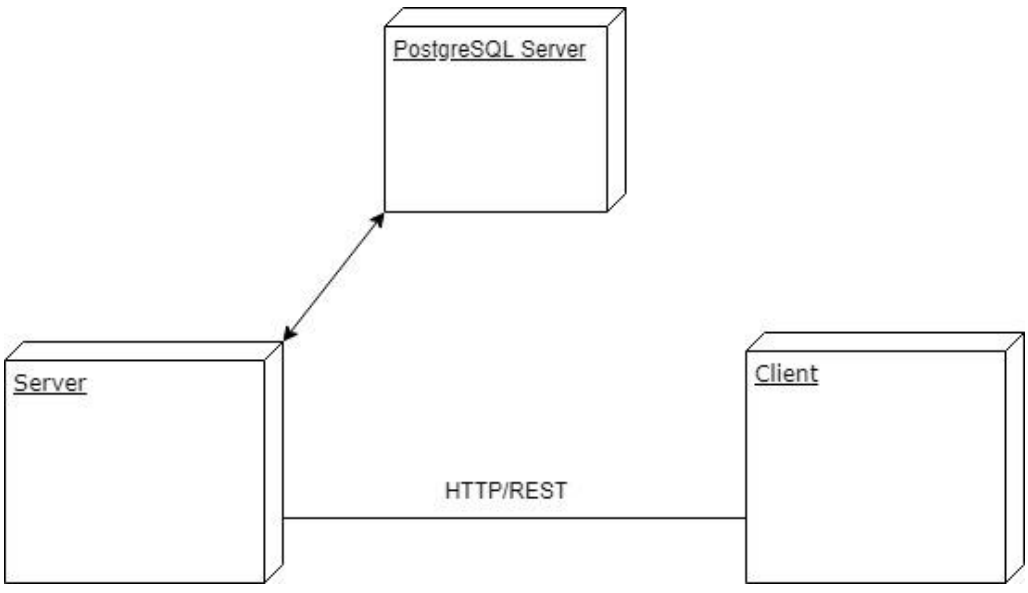


MVP

Package Design



Component and Deployment Diagrams



II. Elaboration – Iteration 1.2

Design Model

Dynamic Behavior

[Create the interaction diagrams (1 sequence, 1 communication diagrams) for 2 relevant scenarios]

Class Design

[Create the UML class diagram; apply GoF patterns and motivate your choice]

1. Data Model

[Create the data model for the system.]

2. Test Strategy

[Present the used testing methods and the associated test case scenarios.]

III. Elaboration – Iteration 2

1. Architectural Design Refinement

[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]

2. Design Model Refinement

[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]

IV. Construction and Transition

1. System Testing

[Describe how you applied integration testing and present the associated test case scenarios.]

2. Future improvements

[Present future improvements for the system]