UNIVERSITY OF PISA

MSc in Computer Engineering

Project for Distributed Systems and Middleware Technologies

Real-Time Auction Manager

Professors:

Prof. Alessio Bechini

Students:

Taulant Arapi (645308)

Antonio Ciociola (645324)

Simone Gallo (661993)

Antonio Andrea Salvalaggio (646238)

ACADEMIC YEAR 2025/2026

1 Project Proposal

Auction Manager is a distributed web-app in which users can sell their goods by creating Online Auctions. Registered users have the possibility to join an auction in order to buy a good in case they beat the other users by setting a higher offer on a given limited time.

1.1 Requirements

The functional requirements below are grouped according to their respective themes.

User Management:

- **User Registration**: Non-Registered Users must be able to create an account to become Registered Users.
- **Balance Management**: Registered Users can deposit funds into their personal accounts to maintain a digital balance.

Auction Creation and Management:

- **Auction Creation**: Registered Users can create new auctions by specifying a scheduled start date and time.
- **Configure Auction Parameters**: Registered Users can configure, upon creation, all the auction parameters, like minimum bid and increment.
- **Auction Cancellation**: The creator of an auction (Registered User) may cancel it, provided the auction has not yet started.
- **Viewing Upcoming Auctions**: Users can browse and view a list of scheduled auctions that are pending start.

Participation and Bidding:

- **Registration for Participation**: Each auction, before it begins, has a waiting list. Registered Users can join one waiting list, indicating their will to participate to a specific auction.
- **Restricted Access**: Once an auction starts, participation is strictly limited to those Registered Users who joined the waiting list beforehand.
- **Viewing Access**: All Users, including unregistered ones, can join an ongoing

auction as a spectator and therefore view the countdown and current best offer while not being able to place an offer.

- **Bidding Process**: During a live auction, participating users can submit bids.
- **Real-time Updates**: When a new bid is placed, all participants must see the updated high bid immediately.
- **Dynamic Countdown**: Every new bid resets the auction countdown timer. The auction concludes only when the countdown reaches zero without any further bids.
- **Balance Validation**: A participant cannot place a bid that exceeds their current available balance.

These are the **actors** in our project:

- **Non Registered User**: A guest visitor who can only view public landing pages or register.
- **Registered User**: A logged-in user with the authority to create auctions, manage a balance, and participate in bidding.

1.2 Architecture

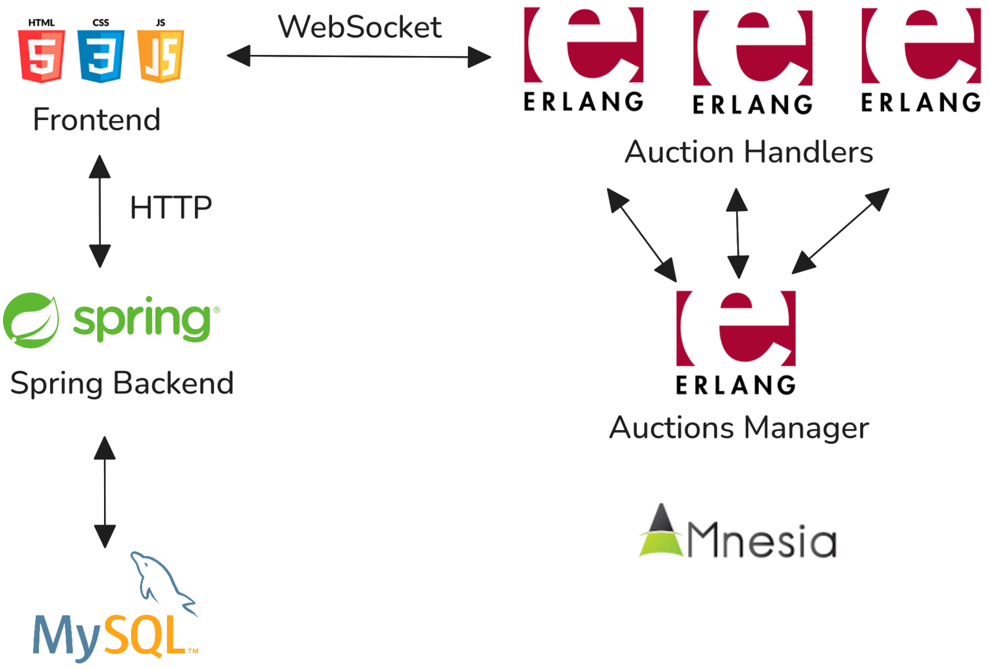

Figure 1.1: Architecture of the system

The **front-end** is a simple web application (HTML+CSS for the presentation, JavaScript for the logic).

The **back-end** is composed of two parts:

- A **Spring Boot** server for handling user registration and management. It also handles the user-facing part of auction creation and serves the web application to the client. It uses a MySQL database to save persistent data.
- An **Erlang** manager that creates nodes to handle auctions in a distributed manner. The nodes connect to the clients participating in auctions using WebSockets. The Erlang nodes use Mnesia for data storage.