# Restaurant ordering System

During the development, we completed all requirements including bonus parts. React and Spring boot actively used for frontend, backend and their interactions. Firstly, we implemented our backend architecture entities, repos, and business logic which are at service implementation package. We designed their relationships and focused on how these entities will interact and be fetched when needed. For easy use during development, we directly used in-memory h2 database. Once these parts are ready, we simultaneously started to work on frontend and remaining backend components. On backend, security concept is addressed with spring security authentication and authorization system leveraging the JWT token-based approach. Also, each entity has its DTO to ensure that they are not directly exposed to outside. Coming to Redis, it is implemented for both token and order management. From token side, it handles whitelisted and blacklisted tokens. And order manager caches orders for frequent access while preserving consistency between database and Redis cache.

Besides all this, we also have QR generation for tables, error handling, and tests which include integration, UI, and unit tests.

We lastly also implemented Order processing workflow using RabbitMQ service for message delivery which is consumer producer service. It was easy to implement at the end since it doesn’t disturb general interaction between our frontend and backend (it was called inside of controllers, general working principes of controller wouldn’t be disturbed)

|  |  |  |
| --- | --- | --- |
| Name | contribution | percentage |
| Ali Ahmadli | Backend frontend | 25% |
| Farid Garayev | Backend frontend | 25% |
| Ahmad Huseynli | backend | 25% |
| Fidan Aliyeva | backend | 25% |