Prenotalo: project high-level specification

Cecchetto A., d'Antonio S., Perrella C., Roccia E., Zerpa Ruiz J. E.

January 2024

Glossary

Term	Definition
Consumer	Someone that requests or books a service offered by a Producer
Producer	Someone that provides services to Consumers
User	Platform user, so both Consumer and Producer
Event	A defined time-slot in which a Producer concretely provides a service to one or multiple Consumers
Service	A service that can be offered by a Producer to a Consumer in a time-slotted manner

1 Introduction

In this document, we provide the initial idea and high-level specification of the project describing what are its objectives, what are its core functionalities, what are the potential users and the most important use cases.

The aim of this project is to create a distributed system, made up of different micro-services.

2 Objectives

With objectives of the project we mean what the system aims to facilitate or solve. The system "Prenotalo" provides booking management functionalities, integrating also additional services and utilities like notification and analytics.

It facilitates interaction between consumers and producers. It aims to provide a simple and straightforward way for consumers to find, book, and manage their appointments or reservations. It helps producers manage their resources effectively.

3 Functionalities

For clarity sake we define as *producer* an user that provides some kind of services that can be offered in a time-slotted manner. Instead the *consumer* is the one that requests or uses these services.

For functionalities, we mean what the application should do in order to satisfy concretely the objectives. The functionalities of the system will be:

- Provide a way to register user data.
- Provide a way for a producer to declare offered services
- Provide a way for a producer to create new events.
- Provide a way for a producer to cancel events.
- Provide a way for a consumer to look for available events.
- Provide a way for a consumer to book an event.
- Provide a way for a consumer to look for booked events.
- Provide a way for a consumer to cancel a booking.
- Provide a way for a consumer to pay for an attended event.
- Send consumers notifications about booked events, e.g. reminders.

- Provide a way for a producer to create notifications about an event, e.g. broadcast messages.
- Collect surveys about completed events.
- Provide data analytics obtained by the system to producer.
- Provide a web application interface to use the system.
- Provide a mobile application interface to use the system.

Follows a more detailed and lenghty description of the core functionalities of the system:

- Producers can declare and show to consumers what services they offer and optionally their price. Example: consider as a producer a dance school. They can offer as services individual hip-hop dance classes and salsa classes, both for a duration of one hour.
- A consumer can ask to book a service offered by a producer, resulting in the creation of an event. Example: let's consider a consumer for the dance school aforementioned. He wants to book an individual salsa class for the afternoon. He selects the salsa class and then chooses from the time availabilities. In this way he sends a request to the producer to create a dedicated event for him.
- A producer can create "public" events, to which any consumer can partecipate. Example: the dance school offers a group course of hip-hop dance every saturday afternoon to 18:00 to 20:00. The dance school can create a public event for this group classes. Consumers can see these public events and send a request to partecipate.
- Events can be cancelled by producers. Consumers can also cancel their booking to events. The cancellation of an event triggers a notification that is sent to the other party.
- Producers can also send notifications to consumers that partecipate to an event. Example: a producer can send a notification to all those that partecipate to a public event, the same happens for events that are created after a consumer request.

4 Potential users

This booking system aims to be general enough to be useful any time there is a service offered in slots of time. Hence, some examples of potential users might be lawyers, doctors, any self-employed practitioner or small buisnesses like hair salons, schools like driving schools, dancing schools, etc. Naturally, also clients of such businesses are potential users of the system since it would be used to book the services offered by them.

5 Main use cases

We decided to not follow strictly the UML structure of the use cases both for ours easiness and of the reader. However it is included something that resembles the main flow of them.

The most relevant use cases of the system are the following:

- **A producer defines its offered services**: a producer can declare to customers what set of services they offer. When creating a new service the producer must define its name and a duration in hours. Optionally the price for that service can be declared.
- **A producer creates an event**: a producer defines a name for the event, specifying its duration and a start time. Optionally it can be declared a partecipation cost. Then the event is published and so is viewable by consumers.
- **A consumer requests an event**: a consumer finds the appropriate provider that offers the needed services. Then the consumer selects the service from the list of offered services, of the relative producer. Then selects a time slot and sends a request for an event to the producer. The producer is then notified and can consequently accept or discard the request.
- **A consumer books an event**: a consumer searchs for a producer, cheking if there are events to which he can partecipate. The consumer then selects the event of interest and sends a booking request. The producer then recieves a request of partecipation that can either be accepted or discarded.
- **A consumer pays for a booked event**: a consumer checks for events that he has booked. Selects an event that supports payment trough the system. The payment sequence is then managed by an external service that then notifies wheter the payment was succesful or not.