Canidae Resort

Supplementary Specification

Mureșian Dan-Viorel
30433

Date	Version	Description	Author
18/03/2018	0.1	Document created	MDV
19/03/2018	0.2	Document update 01	MDV
27/05/2018	0,3	Document update 02	MDV

Table of Contents

1. Introduction

The aim of this document is to capture the system requirements that are not readily captured in the use-cases of the use-case model document. It will include non-functional requirements such as availability, performance, security, testability and usability. It will end by specifying some design constraints.

2. Non-functional Requirements

2.1. Availability

The user will be able to access and view offline information regarding its account and his dog profiles at all times, even when other services are offline. Maintenance will take place once per week, with prior announcement, between 3 AM and 4 AM on Thursdays.

2.2. Performance

- The server will be able to perform 1000 bookings per second as well as answer 2000 search requests per second;
- While 20000 regular users will be able to use the application simultaneously, only 5000 users will be served according to the previous statement;
- When used optimally, the application will be able to answer or process any non-payment related request in less than 3 seconds;

2.3. Security

- Highly encrypted and secure payment system;
- Data protection by database encryption and encrypted communication between client and server;
- Data recovery and database failure prevention due to multiple and frequent back-ups as well as multiple back-up servers;
- HTTPS certificate;
- Anti-DDOS mechanisms;

2.4. Testability

Multiple-phase testing:

1. Internal testing: unit, module, black and white box testing;
2. Partnership with a dog hotel (provide the service for free) in order to test server capabilities, followed by alpha client testing with limited availability (~10000 users), payment through application disabled or limited to testing;
3. Beta testing where people can freely download the application with frequent feedback requests, payment through application at user's risk, but it will offer advantageous discounts, new features added throughout the beta phase;
4. Full feature release with rewards for testers.

2.5.   Usability

The application's GUI will provide a user-friendly, interactive experience with the application and will easily enable users to perform all sorts of available operations inside the application.

3. Design Constraints

- Client-Server Architecture;
- Implementation using Java;
- Spring MVC usage;
- Thymeleaf usage for communication between html files and the application;
- Development using the Eclipse IDE.