

<b>WattaFood</b>	
Supporting Requirements Specification	Date: <10/04/18>

# WattaFood

## System-Wide Requirements Specification

### 1. Introduction

Este documento detalha requerimentos específicos do sistema.

### 2. System-Wide Functional Requirements

Consistência do banco de dados, disponibilidade do sistema de alertas.

### 3. System Qualities

#### 3.1 Usability

O sistema deve ser fácil de usar – faremos testes com pessoas e nenhuma deve necessitar consultar documentação para fazer um cadastro, pelo menos. Alguma documentação em relação ao uso de detalhes de dietas é aceitável.

#### 3.2 Reliability

O serviço per se será bem leve – não há necessidade de preparos para spikes acentuados de carga. Entretanto, queremos disponibilidade permanente. Deverá haver um servidor (rápidamente substituível em caso de dano) somente para balanço de carga, e servidores dedicados para o próprio serviço.

#### 3.3 Performance

Os aplicativos irão ter um cache local de dados. A sincronização com o servidor principal deve ocorrer pelo menos uma vez por minuto.

#### 3.4 Supportability

A instalação será automática pelo sistema de aplicativos centralizados das plataformas mobile. Detalhes sobre configuração do serviço e atualização cairão sob supervisão do sistema de deploy.

Escalamento do serviço será feito manualmente, da seguinte maneira: Faremos controle de carga/número de usuários com um certo valor de “parada”. Ao chegarmos a atender este número, iremos escalar com mais hardware.

Upgrade do software (correção de bugs, atualização de regras de negócio) deverão ser feitas em um servidor de teste separado. Ao longo do tempo, coletaremos casos reais de uso para colocar no servidor de teste para executar testes automatizados toda vez que uma nova versão deve ser testada.

### 4. System Interfaces

Na camada de acesso, teremos a interface entre o aplicativo e o serviço no back-end. Suportaremos, inicialmente, dois protocolos de acesso: REST e SOAP (via XML).

#### 4.1 User Interfaces

As interfaces de usuário deverão ser “não-atômicas”, ou seja, cada ação deverá ser possível de ser desfeita antes de confirmada. Também deve ser possível fazer zoom de qualquer parte e em qualquer momento.

<b>WattaFood</b>	
Supporting Requirements Specification	Date: <10/04/18>

#### *4.1.1 Look & Feel*

Deve ter um “feel” de aplicação médica. Não deve parecer com interface de serviço de pedir comida, por exemplo.

#### *4.1.2 Layout and Navigation Requirements*

Os links de acesso ao menus mais acessados (dieta atual, etc) devem ser levemente mais visualmente destacados do que os demais (fazer cadastro, por exemplo). Deve ser possível voltar ao menu anterior sem perder as interações com o menu atual. Deve ser possível avançar e voltar a navegação entre menus.

#### *4.1.3 Consistency*

Manter o texto e os ícones consistentes. Iremos fazer testes com pessoas afim de detectar “surpresas” - especialmente antes de lançar uma versão nova.

#### *4.1.4 User Personalization & Customization Requirements*

Dependendo da interação com o nutricionista, algumas dietas e/ou avisos deverão ser mostrados no menu principal da aplicação. Não será permitido customizar a aparência.

### **4.2 Interfaces to External Systems or Devices**

Teremos acesso a um sistema de banco de dados separado do sistema principal. Isto será gerenciado automaticamente por um componente de acesso fora do nosso controle.

#### *4.2.1 Software Interfaces*

Usaremos JDBC para acesso ao banco de dados, que residirá em outra máquina. Detalhes sobre o JDBC estão fora do escopo deste documento.

#### *4.2.2 Hardware Interfaces*

Usaremos apenas hardware de prateleira, não havendo, assim, requisitos de hardware específico.

#### *4.2.3 Communications Interfaces*

Utilizaremos acesso de comunicação através de rede local IPv6 entre o serviço, o banco de dados, e a máquina de distribuição de carga. Usaremos o kit de desenvolvimento das plataformas mobile para fazer acesso através da internet entre aplicação e serviço

<b>WattaFood</b>	
Supporting Requirements Specification	Date: <11/4/2018>

## 5. Business Rules

### 5.1 <Regras de cadastro>

#### 5.1.1 <Cadastrar cliente>

Se o telefone suportar a feature, o aplicativo deverá poder pré-preencher os dados no formulário com os dados do dono do telefone.

#### 5.1.2 <Cadastrar nutricionista>

Ao cadastrar um nutricionista, o sistema deverá fazer uma consulta para validar as credenciais do nutricionista.

### 5.2 <Regras de acesso>

#### 5.2.1 <Acesso do cliente>

O cliente poderá optar por manter-se conectado ao serviço sem a necessidade de redigitar suas credenciais.

#### 5.2.2 <Acesso do nutricionista>

O mesmo que para o cliente, entretanto a cada acesso o sistema deve consultar a atual validade deste profissional com o Conselho Regional de Nutricionistas.

## 6. System Constraints

O sistema será desenvolvido em Java 8. Deverá ter cobertura de código com testes unitários. O balanceador de carga será escrito em C++17.

Os aplicativos mobile serão escritos na língua nativa de cada plataforma (Java para Android, Swift para iOS).

De início, esperamos usar 1 computador para balanço de carga (+1 reserva), e 2 servidores de produção +1 de reserva. Para o banco de dados, teremos inicialmente 1 computador +1 reserva. Todas as reservas serão espelhadas. Estas máquinas deverão ser interligadas através de redes locais.

## 7. System Compliance

### 7.1 Licensing Requirements

Todos os direitos reservados, sem exceções. Todos os gráficos e marcas usadas serão reservadas.

### 7.2 Legal, Copyright, and Other Notices

Deverá haver uma nota de copyright no menu principal, e também deverá estar explicitado no acordo com o usuário final.

### 7.3 Applicable Standards

Atenderemos todos os padrões regulamentatórios do Conselho Regional de Nutricionistas.

<b>WattaFood</b>	
Supporting Requirements Specification	Date: <11/4/2018>

## 8. System Documentation

Toda documentação necessária deverá estar disponível online. Não haverá sistema de ajudas no próprio aplicativo – não deve ser necessário, temos procedimentos justamente para evitar esta necessidade.