Task Manager

System-Wide Requirements Specification

Usage note: There is procedural guidance within this template that appears in a style named InfoBlue. This style has a hidden font attribute allowing you to toggle whether it is visible or hidden in this template. Use the Word menu Tools🡪Options🡪View🡪Hidden Text checkbox to toggle this setting. A similar option exists for printing Tools🡪Options🡪Print.

# Introduction

This project is an exercise for the discipline of Software Engineering II, on the Faculdade Senac Porto Alegre, second semester of 2015, based on a system for task management.

# System-Wide Functional Requirements

[Statement of system-wide functional requirements, not expressed as use cases. Examples include auditing, authentication, printing, reporting.]

To authenticate users, determining if the user is a leader or a member of that team in which he logged in.

# System Qualities

[Qualities represent the URPS in FURPS+ classification of supporting requirements.]

The system is very simple to use, and it is based on already well-known systems, turning the experience pleasant and intuitive.

## Usability

[Describe requirements for qualities such as easy of use, easy of learning, usability standards and localization.]

Adjustable font size and system colors according to user's preferences. Also, there's a label system for tasks.

## Reliability

[Reliability includes the product and/or system's ability to keep running under stress and adverse conditions. Specify requirements for reliability acceptance levels, and how they will be measured and evaluated. Suggested topics are availability, frequency of severity of failures and recoverability.]

Login is mandatory for all users. All changes made on tasks are recorded in a log, so team leaders can see who made which alteration and when it was made. The system is kept in two different servers, in two different countries, for safety. Backups are constantly made.

## Performance

[The performance characteristics of the system should be outlined in this section. Examples are response time, throughput, capacity and startup or shutdown times.]

Response time of the system is immediate.

## Supportability

[This section indicates any requirements that will enhance the supportability or maintainability of the system being built, including adaptability and upgrading, compatibility, configurability, scalability and requirements regarding system installation, level of support and maintenance.]

The system supports eventual updates, and is supported by any operational system, because it runs online. It can accessed through the main web browsers, such as Google Chrome, Mozilla Firefox, Microsoft Edge, Safari and Opera and is responsive to other platforms, like mobile. Detected errors and bugs can be reported through the webpage itself, and will be handled by the development team as they come along. The system supports eventual updates, and is supported by any operational system, because it runs online. It can accessed through the main web browsers, such as Google Chrome, Mozilla Firefox, Microsoft Edge, Safari and Opera and is responsive to other platforms, like mobile. Detected errors and bugs can be reported through the webpage itself, and will be handled by the development team as they come along.

# System Interfaces

[Interface Requirements are part of the + in the FURPS+ classification of supporting requirements. Define the interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, and so forth, so that the software can be developed and verified against the interface requirements.]

The system interface focus on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions.

## User Interfaces

[Describe the user interfaces that are to be implemented by the software. The intention of this section is to state requirements relating to the interface. Interface design may overlap the requirements gathering process.]

Interfaces to be implemented by the system.

### Look & Feel

[Provide a description of the spirit of the interface. Your client may have given you particular demands such as style, colors to be used, and degree of interaction and so on. This section captures the requirements for the interface rather than the design for the interface.]

The interface will be initially mustard yellow and black, but it can be personalized by the user according to his preferences. A yet to be defined serif-font will be used to facilitate the reading.

### Layout and Navigation Requirements

[Capture requirements on major screen areas and how they should be grouped together.]

1. Input Controls: buttons, text fields, checkboxes, radio buttons, dropdown lists, list boxes, toggles, date field;
2. Navigational Components: breadcrumb, slider, search field, pagination, icons, labels;
3. Informational Components: tooltips, icons, progress bar, notifications, message boxes, modal windows;
4. Container: accordion.

### Consistency

[Consistency in the user interface enables users to predict what will happen. This section states requirements on the use of mechanisms to be employed in the user interface. This applies both within the system and with other systems and can be applied at different levels: navigation controls, screen areas sizes and shapes, placements for entering / presenting data, terminology.]

N/A.

### User Personalization & Customization Requirements

[Requirements on content that should automatically displayed to users or available based on user attributes. Sometimes users allowed to customize the content displayed or to personalize displayed content.]

The user will be able to personalize the system’s base colors and also font sizes.

## Interfaces to External Systems or Devices

[Are there any external systems with which this system must interface? Are there any constraints on the nature of the interface between this system and any external system, such as the format of data passed between these systems, and any particular protocol used? Consider both provided and required interfaces.]

N/A.

### Software Interfaces

[This section describes software interfaces to other components of the software system. These may be purchased components, components reused from another application or components being developed for subsystems outside of the scope of this SRS, but with which this software application must interact.]

N/A.

### Hardware Interfaces

[This section defines any hardware interfaces that are to be supported by the software, including logical structure, physical addresses, expected behavior, and so on.]

N/A.

### Communications Interfaces

[Describe any communications interfaces to other systems or devices such as local area networks, remote serial devices, and so on.]

N/A.

# Business Rules

[Business rules are statements that define or constrain some aspect of the business. Business rules are often represented as production rules when they are meant to be directly executed by an IT System: a production rule is an independent statement of programming logic that specifies the execution of one or more actions in the case that its conditions are satisfied. Production Rules define the operation semantic for the system in a technologic independent way. They constrain the behavior expressed in system use cases.

Organize this document on rule classes, a high level grouping of candidate or actual rules about one **business concept** with a specific kind of **logic processing**, example: Driver Risk Assessment Rules or Customer Validation Rules.]

N/A

## <Rule class name>

### <Rule name and ID>

[The description defines the rule. It can be made in natural language typically following a decision table or a pattern like: if [condition-list] then [action-list], example:

If there are at least 3 items of the same type in the customer shopping cart and each item’s value is greater than $30 then give to the customer a voucher whose value is 10% of the cheapest item.]

# System Constraints

[Constraints are part of the + in the FURPS+ classification of supporting requirements. Describe any design; implementation or deployment constraints on the system being built that have been mandated and must be adhered to. Examples include software implementation languages, prescribed use of developmental tools, third-party components or class libraries, platform support, resource limits and requirements on the shape, size or weight of the resulting hardware housing the system.]

The system’s language will be in English by default.

# System Compliance

## Licensing Requirements

[Define any licensing enforcement requirements or other usage restriction requirements that are to be exhibited by the software.]

N/A.

## Legal, Copyright, and Other Notices

[This section describes any necessary legal disclaimers, warranties, copyright notices, patent notice, wordmark, trademark, or logo compliance issues for the software.]

N/A.

## Applicable Standards

[This section describes by reference any applicable standards and the specific sections of any such standards that apply to the system being described. For example, this could include legal, quality and regulatory standards, industry standards for usability, interoperability, internationalization, operating system compliance, and so forth.]

Standards for interface usability by the *usability.gov* website.

# System Documentation

[Describes the requirements, for on-line user documentation, help systems, help about notices, and so on. Set expectations for the documentation and to identify who will be responsible for creating it.]

Documentation will be created by the developer, teaching how to use the system with a step-by-step user guide.