<Task Manager>

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.

# Positioning

This section presents the declaration of the problem (section 2.1) and the product’s positioning (section 2.2).

## Problem Statement

[Provide a statement summarizing the problem being solved by this project. The following format may be used:]

|  |  |
| --- | --- |
| The problem of | remembering all daily duties, |
| affects | the team members and the worker himself, |
| the impact of which is | delaying the delivery of projects, |
| a successful solution would be | set reminders of all daily tasks. |

## Product Position Statement

[Provide an overall statement summarizing, at the highest level, the unique position the product intends to fill in the marketplace. The following format may be used:]

|  |  |
| --- | --- |
| For | team members of a company, |
| Who | have trouble remembering all their tasks on a project, |
| The (product name) | is a task manager, |
| That | keeps record of all tasks and their due dates. |
| Unlike | Trello, |
| Our product | also has alarms that can be set according to specific needs. |

[A product position statement communicates the intent of the application and the importance of the project to all concerned personnel.]

# Stakeholder Descriptions

## Stakeholder Summary

| **Name** | **Description** | **Responsibilities** |
| --- | --- | --- |
| Customer | Company’s manager (Meister Aemon) | Monitors the project’s progress and approves funding; |
| Decision Maker | Team leader (Jon Snow) | Monitors the tasks and the team’s progress on them and delegates tasks; |
| User | Team member (Samwell Tarly) | Executes the tasks delegated to him; |
| Competitor | Trello and RunRun.it | Show that our software has the differential of implementing alarms; |
| Domain Expert | ? | Instruct on the business; |
| Industry Analyst | ? | Instruct on tendencies for the future. |

## User Environment

[Detail the working environment of the target user. Here are some suggestions:

Number of people involved in completing the task? Is this changing?

Usually each task is attributed to one team member. No.

How long is a task cycle? Amount of time spent in each activity? Is this changing?

Today there’s no control of the tasks cycles, only observations.

Any unique environmental constraints: mobile, outdoors, in-flight, and so on?

No.

Which system platforms are in use today? Future platforms?

There are no systems being used today. Future platform includes the usage of Apple software.

What other applications are in use? Does your application need to integrate with them?

No.

This is where extracts from the Business Model could be included to outline the task and roles involved, and so on.]

Main processes: creation, registering and attributing.

# Product Overview

## Needs and Features

[Avoid design. Keep feature descriptions at a general level. Focus on capabilities needed and why (not how) they should be implemented. Capture the stakeholder priority and planned release for each feature.]

|  |  |  |  |
| --- | --- | --- | --- |
| **Need** | **Priority** | **Features** | **Planned Release** |
| Create teams | N/A | N/A | N/A |
| Register team members | N/A | N/A | N/A |
| Create tasks | N/A | N/A | N/A |
| Attribute tasks to enrolled members | N/A | N/A | N/A |
| Set date and time for alarms | N/A | N/A | N/A |
| Set colored labels for defining priority | N/A | N/A | N/A |
|  |  |  |  |

# Other Product Requirements

[At a high level, list applicable standards, hardware, or platform requirements; performance requirements; and environmental requirements.

Define the quality ranges for performance, robustness, fault tolerance, usability, and similar characteristics that are not captured in the Feature Set.

Note any design constraints, external constraints, assumptions or other dependencies that, if changed, will alter the **Vision** document. For example, an assumption may state that a specific operating system will be available for the hardware designated for the software product. If the operating system is not available, the **Vision** document will need to change.

Define any specific documentation requirements, including user manuals, online help, installation, labeling, and packaging requirements.

Define the priority of these other product requirements. Include, if useful, attributes such as stability, benefit, effort, and risk.]

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Priority** | **Planned Release** |
| iOS platform | D/A | D/A |
| User-friendly | D/A | D/A |