Dolendar System Requirements

Inf 43 – Homework 1

###### Oct 17, 2023

Ivan Leung

ID# 67485846

##### Table of Contents

Include a table of contents that lists every section and the page that it falls on.

##### Introduction

This document outlines the requirements specification for a new productivity suite of software, Dolendar, a revolutionary calendar that is developed by the INF43 Software Engineering group and contracted by the founder and president of DoMore Software corp., Dom Dones. What makes Dolendar apart from other calendar apps is that the tasks in Dolendar will flow around the calendar as time moves. If a task is completed, it will disappear from the calendar and the next task will move up and replace current task’s time window so do all following tasks. If a task is not completed within its duration, the current task and all other tasks will be push back.

By investing in the development of Dolendar, Dom Dones, the founder and president of DoMore Software corp., aims to combine a typical calendar app with a to-do list manager to help users mange their time more effectively and combat procrastination.

##### Executive Summary

This section's audience is the non-technical customer and user. Think of a busy executive with little time and little patience to read more than one page. It should address the goals of the system and why it is needed. Describe the major features of the system and the rationale for each (high level only, you will provide details later). Identify and describe other software, processes, hardware, people, and policies that the system may or will affect. List any assumptions made about the existing world or application context (high level only, you will provide details later).

##### Application Context / Environmental Constraints

Now you can provide more detail about “context”, for example will the application run in a business or office, home or outdoors, on a desktop, laptop, tablet, or mobile phone? Also discuss anything you might know about the operating system and “platform”, hardware constraints, design constraints (for example, what should the UI look like), software constraints (such as programming language) etc. Also mention any other software that this software will interact with.

The first version of Dolendar is going to be available on mobile devices only such as Apple iPhone(iOS), Android phones and tablets. The app is installed on the devices themselves. At the time of release, the app will be compatible with the current version of iOS and Android and their previous versions.

The desktop version of Dolendar will be available in the second version. The app will be available on the three most popular desktop platforms such as Windows, Mac and Linux.

There is no specific programming language required. However, Kotlin and Swift are expected to be used in developing for Android and iOS respectively due to Kotlin has numerous advantages over other programming languages and

The app is expected to be run primarily on mobile devices as most people would track their daily tasks on their mobile phones. Users can store their data on their phones locally. The app is free to use, and it is not necessary to create an account. Users have the option to create an account which can potentially unlock paid tier features such as creating and syncing tasks with others.

The UI provides calendar view and to-do list view. The calendar view has both week view and day view. The to-do list view outlines all tasks. The UI should be elegant and easy to use. Further details will be discussed with the UI engineers.

App integration is planned in the future where it can sync with other calendar apps such as Google Calendar, iCalendar, Outlook, etc.

##### Functional Requirements (we will save this section, as well as the use cases for Homework 2)

This section is the heart of the specification document. It clearly describes the proposed software product, including its capabilities and attributes. Do *not* describe details of the user interface, such as “button in the upper left corner” or “press Ctrl-S to save” or “choose from a drop down box”. Instead, focus on the functionality that the system will offer.



Describe the functional requirements of the system in precise detail. When possible, identify the entities (components, sections, areas of functionality) that make up the system. Characterize the properties, states, functions, and interrelationships of each entity. Since this section is the core of the requirements document, it warrants its own brief introduction.

This section will also include a use-case diagram.

**Non-functional Requirements**

Generally speaking, software qualities are everything we want our software to be that is not a feature or functional requirement. The categories are usually multi-syllable words that end with “ility” or at least “ty”, such as safety, security, reliability, portability, and especially usability. Discuss constraints pertaining to those qualities, if relevant, as well as speed, space, robustness, implementation bias, etc. Your audiences here are the system designers and programmers, who will probably not be in direct communication with the users. This section will help them assess trade-offs in the system's implementation. List all of the software qualities and/or non-functional requirements that the customer mentioned.

##### Other Requirements

We often feel we have more to say, in other words somehow the previous two sections did not cover all requirements. This is the place to do it! Any requirements that you are aware of can be described here; you can also add diagrams or other visuals that did not fit in previous sections.

**Glossary**

Define a Glossary of Terms. Include any terms that have a specific meaning for this project.

Dolendar: a revolutionary calendar which combines to-do list and traditional events one may find in a calendar app.

Task: an item in the to-do list. It has duration which should be completed within its specific time windows. If it is not completed within its specific time window, it automatically flows down to the next time windows.

Event: a traditional event in most calendar. It only happens at a specific time. It does not flow around the calendar unlike tasks.

##### Assumptions / Risks

This is the place to cover any assumptions not covered earlier. Also, list any known risks to the project from any and all perspectives, for example financial risks, business risks, legal and ethical risks, project management and development risks, etc. List all of the assumptions/risks that the customer mentioned in this section.

##### Priorities / Implementation Phases

If applicable, and following the request of the user, identify one or more subsets of the system's functionality which can or will be implemented first. Give any milestones that were established, or give an order to the features that were prioritized, as agreed to by the customer.



##### Future Directions and Expected Changes

Again you are providing insight and guidance to the system designers and programmers. List all of the future directions/expected changes that the customer mentioned.