POLITECNICO
DI MILANO

POMASANA

Documentation

Giacomo Bresciani

Service Technologies 1
Prof. Sam Jesus Alejandro Guinea Montalvo
Politecnico di Milano

Contents

1 Introduction **2**
- 1.1 Purpose . . . 2
- 1.2 Description . . . 2
- 1.3 Glossary . . . 3
  - 1.3.1 Pomodoro Technique Terms . . . 3
  - 1.3.2 Asana Terms . . . 4
  - 1.3.3 Pomasana Terms . . . 4
- 1.4 Goals . . . 4
- 1.5 References . . . 5

2 Requirements Analysis **6**
- 2.1 Functional Requirements . . . 6
- 2.2 Non-Functional Requirements . . . 7
  - 2.2.1 System Architecture . . . 7
- 2.3 Specifications . . . 8

3 Use Cases **9**
- 3.1 Actors . . . 9
- 3.2 Use Cases . . . 9
  - 3.2.1 Registration . . . 9
  - 3.2.2 Login . . . 10
  - 3.2.3 Profile editing . . . 10
  - 3.2.4 PomoTask creation . . . 11
  - 3.2.5 PomoTask editing . . . 11
  - 3.2.6 Pomodoro creation . . . 12
  - 3.2.7 Sending report . . . 12

4 Architectural Design **13**

Chapter 1

Introduction

1.1 Purpose

The purpose of this document is to illustrate the process followed during the development of the software, from the analysis of requirements to the system design.
This project is developed in the context of the course Service Technologies 1 of Politecnico of Milan. The goal to achieve is to find public web services and integrate them to create a new service that adds value and functionality to the existing.

1.2 Description

The system that is going to be developed is named Pomasana, from the mix of the words "Pomodoro" and "Asana". The idea is to integrate the "The Pomodoro Technique®" and Asana. To better clarify what is Pomasana is necessary to explain more in details "The Pomodoro Technique®" and list some of the main functionalities of Asana:

- "The Pomodoro Technique®" is a way of managing time and becoming more productive by managing tasks in 25 minute intervals followed by a short break of 5 minutes. Here are the steps:

    1. Identify the task

2. Set a timer to 25 minutes (a Pomodoro)
3. Focus on work until the Pomodoro ends
4. Take a 5 minutes break
5. every four Pomodoro take a longer break (15-30 minutes)

The technique includes other aspects such as internal/external interruptions and planning. For more details refer to the section 1.5.

- Asana is an online project management tool, with the possibility to define workspaces, set tasks and organize them in projects, assign tasks to people, define deadlines and many other features. For Pomasana is fundamental that it exposes a set of public API that allows to access much of the data in the system.

Pomasana is going to integrate the main functionalities of Asana with the concept described by "The Pomodoro Technique®" offering an easy way to extract the best from the two services.

1.3 Glossary

Below are some terms that are useful to understand this document:

1.3.1 Pomodoro Technique Terms

Pomodoro
A single unit of time, composed by 25 minutes of work and 5 minutes of break: it is the base concept of "The Pomodoro Technique®".

Interruptions
In the context of "The Pomodoro Technique®" can occur in the middle of a Pomodoro and can be both internal and external.

Unplanned activity
An activity that is created in the middle of a Pomodoro following an Interruption.

Todo Today Sheet
List of activities planned for the current day of work.

Activity Inventory sheet
List from which to choose a subset of activities to be moved into the Todo Today Sheet.

1.3.2 Asana Terms

Asana Task
A task that exist on the Asana system.

Asana Project
Container for multiple tasks.

Asana Workspace
A collection of people and the projects and tasks they work on together.

1.3.3 Pomasana Terms

PomoTask
An Asana Task enriched with some element derived from "The Pomodoro Technique®".

1.4 Goals

The main goal of this project is to create a web service composed by a set of API that allows other developers to exploit it and produce web application or application for mobile phones.
However I'll develop a simple application (web or android) with the only purpose of show the functionalities of the service.
To clarify the goals of the project is useful to list the possibilities that a user of a final application, exploiting the Pomasana API, will have:

- Register and login to the service.
- Edit his personal informations.
- Create a PomoTask that will be automatically added also to his Asana workspace.

- Modify or delete a PomoTask that will be synchronized with the corresponding Asana task.
- "Complete" a Pomodoro and add it to a PomoTask.
- Mark a Pomodoro with interruptions and add notes to it.
- Send a daily report to an email.

1.5 References

In this section can be found some useful references for a more complete understanding of this document and the project in general.

asana.com
The Asana homepage.

developers.asana.com/documentation
The Asana Api Documentation.

pomodorotechnique.com
"The Pomodoro Technique®" homepage.

Chapter 2

Requirements Analysis

2.1 Functional Requirements

In this section, based on the goals described in the section 1.4 I will identify the functional requirements of the APIs of Pomasana; they will coincide mainly with the requirements of an application based on them because all the functionalities will be exposed to the public.

Register and login to the service

- Provide a registration functionality integrated with the Asana one.
- Provide a login functionality always with Asana login.

Edit his personal informations

- Provide a functionality that allows the user to modify the personal informations.

Create a PomoTask, automatically added also to Asana workspace

- Provide a way to create a new PomoTask.
- When a new PomoTask is created it is automatically created also in the Asana user account.

Modify or delete a PomoTask, synchronized with the corresponding Asana task

- Provide a function to make an estimate of required Pomodori on a particular PomoTask.
- Let the user mark the PomoTask as completed, synchronizing its state with Asana.
- Allow to delete a PomoTask, also deleting the corresponding Asana task.

'Complete" a Pomodoro and add it to a PomoTask

- Provide a way to "complete" a pomodoro, adding it to a PomoTask.

Mark a Pomodoro with interruptions and add notes to it

- For every Pomodoro added to a PomoTask make it possible to add the count of internal and external interruptions.
- Make it possible to add Notes to a Pomodoro.

Send a daily report to an email

- Let the user choose to send a daily report of the completed PomoTask and the Pomodoro used to an email.

2.2 Non-Functional Requirements

In this section will be described some requirements that are not related with functionalities of the system that is going to be developed. They are independent from the application domain but they are relevant for design of the system.

2.2.1 System Architecture

The nature of the project imposes that the service is available through the Internet, making possible for every application all around the world to access the API with simple Http request like GET and POST. So it's fundamental that system is available twenty-four hours a day, with a stable connection

and appropriate band; they are also important requirements like reliability and scalability.
To achieve that it was decided to rely on Google infrastructure exploiting its PaaS Google App Engine (more on this topic in chapter 4).

2.3 Specifications

In this section are listed some specifications and assumptions needed to meet the requirements of the system:

- An user that is going to register to Pomasana must already be registered to Asana, otherwise there is no possibility to exploit the functionalities of the system.
- add more...

Chapter 3

Use Cases

This chapter is dedicated to explain the main use cases that can occur when using an application (in this case a web application) that takes advantage of the Pomasana API. Again, despite the following scenarios and use cases describe the interaction with an application and not directly with the system that is going to be developed, it is correct to assume that they reflect how the API are used; in fact all the functionalities of Pomasana are entirely exploitable through its API.

3.1 Actors

The only two possible type of actors are:

Unregistered User He can only access the application Home Page and eventually register to Pomasana.

Registered User He can exploit all the Pomasana functionalities.

3.2 Use Cases

3.2.1 Registration

Actor Unregistered User

Entry condition None

Event Flow

- The user opens the Home Page of Pomasana.
- The user is redirected to an Asana page that ask for permission.
- The system add the new user to the database.
- The system sends a confirmation email.

Exceptions If the user is not registered to Asana he is redirected to the Asana registration page.

Consequences The user is registered to Pomasana and ready to use the system.

3.2.2 Login

Actor Registered User

Entry condition The user must be registered to Pomasana.

Event Flow

- The user performs a login request on the Home Page.
- See how Asana OAuth works....
- ...
- The user is redirected on his personal page.

Exceptions ...

Consequences The user is now logged into the system and can use all its functionalities.

3.2.3 Profile editing

Actor Registered User

Entry condition The user must be logged into the system.

Event Flow

- The user access his personal page.
- He modify the personal info and confirm.
- The system update the database with the new data.

Exceptions The new data are not complete and the system notify the error.

Consequences The personal infos of the user are changed.

3.2.4 PomoTask creation

Actor Registered User

Entry condition The user must be logged into the system.

Event Flow

-

Exceptions

Consequences

3.2.5 PomoTask editing

Actor Registered User

Entry condition The user must be logged into the system.

Event Flow

-

Exceptions

Consequences

3.2.6 Pomodoro creation

Actor Registered User

Entry condition The user must be logged into the system.

Event Flow

-

Exceptions

Consequences

3.2.7 Sending report

Actor Registered User

Entry condition The user must be logged into the system.

Event Flow

-

Exceptions

Consequences

Chapter 4

Architectural Design