Rares-Sebastian Moldovan
30433

Students' algorithm teaching tool
Vision

Version 1.0

Revision History

Date	Version	Description	Author
19/03/2018	1.0	First version of the vision documentation.	Rares-Sebastian Moldovan

Table of Contents

Vision

1. Introduction

1.1 Purpose

The purpose of this document is to provide a high-level view over the Algorithm teaching tool that will be implemented. It describes the main aspects of the tool, such as users and stakeholders, as well as the main terminology used in the project. It focuses on the necessary capabilities needed by all users at a high level, which are to be explained and detailed in the use-case and supplementary specification documents.

1.2 Scope

The Vision document has as main scope an introduction to the "Students algorithm teaching tool" project realized for the Software Design project assignment. Besides being a school project, the tool can be transformed into a really useful platform for student inter-communication and aiding. This document provides the users with an understanding of the main aspects that are involved in the development and characterization of this software tool.

1.3 Definitions, Acronyms, and Abbreviations

SATT = Students' Algorithm teaching tool.

Implementation technology = programming language in which an algorithm can be implemented. Some examples are Python, Java, ML, Prolog, Java.

Stakeholder = an accountant, group, organization, member, or system that affects or can be affected by an organization's actions.

Algorithm = sequence of steps needed for obtaining an observable result from a given specification (input to output).

CRUD = basic operations when manipulating data: Create, Read, Update and Delete.

Data structure = a particular way of organizing data which usually aids some computational problem.

Algorithm administrator = user which is responsible of a certain algorithm on the application.

Regular user = the user which is not implied in any kind of organizational practices with the tool.

1.4 References

- Project Use case model
- Project glossary
- Project Supplementary specification

1.5 Overview

The rest of the Vision document contains the problem statement of the problem that the tool engages to solve, as well as basic explanations about the users, the stakeholders and some high-level requirements of the system (such as platforms, environment and very little about the performance and portability of the system).

Section 2 covers the Positioning of the system, involving the problem statements in a very readable manner as well as the usefulness of the tool and the way it distinguishes itself in a world in which the world wide web seems to provide the fundamental and necessary platform for acquiring any type of information, as well as how the structure of the system proves this belief wrong.

Section 3 covers the users and stakeholders' description in both text-like manner and organized in tables.

Section 4 introduces some of the high-level requirements, without going into any kind of detail about the design of the system, or even the real performance issues.

2. Positioning

2.1 Problem Statement

The problem of	*Implementing a common platform for algorithm understanding and implementation in different technologies via the interest of students in inter-communication and helping each other.*
affects	*Students and potentially professors.*
the impact of which is	*A simpler way for finding answers to commonly-asked questions regarding algorithm meaning, usefulness, complexity and examples.*
a successful solution would be	*Welcome by all students across the computer science department, or by other students which have a sort of interest in computational algorithms.*

2.2 Product Position Statement

For	*The computer science student*
Who	*Needs to understand algorithm(s) better and understand an implementation such that, eventually, he/she is capable of proposing his/her own implementation.*
The "SATT"	*is a teaching tool for students*
That	*is addressed to a specific organization (the university) and the students of the computer science department.*
Unlike	*other computer science websites which are not organized by subjects/fields.*
Our product	*provides a more university-dedicated web page.*

3. Stakeholder and User Descriptions

The stakeholders for this tool may be seen as the two main representative components of a university: the personnel and the students.

From the personnel, it is obvious that we include the teaching staff (the professors and their assistants) which can scan the content of the web page for errors, false or redundant information. The tool can be a common ground for the professors to provide their students the necessary information for understanding the subject. In programming-related disciplines with many theoretical aspects (such as data structures, algorithms, programming techniques, logic/functional programming, distributed systems, software design and engineering).

The students have the biggest interest in the usage of such a tool since they can be both helpful and helped. While some students may like data structures and working with complex algorithms for manipulating data, some others may like hardware-related algorithms (such as algorithms for developing integrated circuits, memory management, or any other digital system design techniques). The former may engage in explaining the data structures and contributing to the software department of the tool, while the latter can provide explanations for basic computer systems development algorithms. This can solve the problem of using several platforms (such as Google drive) for students to share common information. Obviously, no plagiarism is supported in a university or any other type of school, so the students are to explain and implement the purely-theoretical problems and the users must interpret, understand and use them with great care.

3.1 Stakeholder Summary

Name	Description	Responsibilities
University	*The organization which provides the students the necessary information for graduating in a field, as well as the specialization necessary for people to acquire a job.*	*This stakeholder ensures that the tool does not become a purely plagiarism tool and makes sure that the information and implementations are correct and no false information is provided to the students.*
Student organization	*Organizations which are responsible for representing the students in the university meetings.*	*This stakeholder makes sure that the users keep interest and it assigns the tasks to each type of user (simple users, algorithm administrators and main administrators).*

3.2 User Summary

Name	Description	Responsibilities	Stakeholder
Administrator	*They represent the main responsible users for the system.*	*Coordinates the system at all times by allowing users to become "algorithm administrators" and is capable of creating, deleting, updating and reading information about all users and all algorithms. (CRUD operations).*	*University professors and main responsible students (tool admnsitrators).*

Algorithm administrators	*They represent persons in charge of the individual pages of the algorithms.*	*One algorithm administrator is responsible for both providing their own information about the subject as well as accepting solutions and information from other users.*	
Regular users	*They represent the most frequent type of users which are interested in learning and also helping to develop the pages with new information.*	*They must identify themselves and, in case of contributing users, must ask the algorithm administrators to validate their contribution.*	

3.3 User Environment

The target users number might be very large and constantly changing since the university accepts over 400 students in the computer science department annually. Therefore, the page must be suitable for large amounts of users and continually-changing administrators.

A task cycle may take from seconds to hours depending on users and their purposes. A user providing an implementation must spend time in understanding the algorithm on the page, watch some other implementations and eventually submitting a personal implementation for the administrators to verify. On the other hand, a regular user might use the system for informative purposes (such as logging in to see the complexities of some algorithms and then log out).

There are no environmental constraints since, being a web application, it is portable on a mobile (using a browser-type application) and any sort of platform that supports basic operating systems (Windows, Unix, Android, iOS etc.).

The only constraint may be belonging to the university the tool addresses to, but this is not very restrictive since the tool can be implemented in both ways: users could need to be validated by an administrator or not.

The application supports the usage of text editors and doc, pdf formats for uploading information about the algorithms and, covering such a vast domain as algorithms, it will not imply any constraints on the type of files that ca be uploaded (since extensions may range from .java, .doc to .ml, .pl and other not so common extensions).

4. Product Requirements

The product is required to work on both desktop and mobile, which is easy to acquire due to it being implemented as a web application. In order to access it, a browser-type application is required (Internet Explorer, Google Chrome, Mozzilla Firefox, Safari, Opera etc.).

Regarding performance, the computation time must not be obviously delayed, meaning that the pages must be accessed quickly and requests to the server (such as noticing administrators that a user wants to submit an implementation) must be handled quickly, in the worst case in a matter of seconds. Therefore the system response time must be close to optimal.

Besides the usage of a web browser, no other environmental requirements are imposed. Neither are hardware requirements in the limelight since the Java-based implementation is obviously hardware-independent.