UNIVERSITY OF PUERTO RICO AT MAYAGÜEZ
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Panda Code Reviews

A PROJECT PROPOSAL SUBMITTED AS A PARTIAL REQUIREMENT OF THE COMPUTER ENGINEERING PROJECT DESIGN COURSE ICOM-5047

by:

Nelián E. Colón Collazo - Project Manager
Samuel A. Rodríguez Martínez - Document Editor
Daniel A. Santiago Rivera

For: Professors Nayda Santiago and José Fernando Vega
Course: ICOM 5047
Date: September 1, 2013

Executive Summary

Assignments for Software Engineering classes have particular characteristics that make them harder to grade by professors and teaching assistants. In the majority of cases, they must be handed in electronically. The grader then runs the code with some specific input to verify the correctness of the code. Students might choose to use different programming languages, different IDEs, and different tests, making it even more difficult for grading. Additionally, many students write code in different styles, which makes it harder to read for a person that is used to a particular coding style. Furthermore, communication between the grader and the student is very poor, often relying in slow e-mail responses. All of this makes grading software assignments an enduring task.

This product is being designed to eliminate the difficulties that come with grading software assignments. It is a web-based application that will host repositories that contain students' code. When a student uploads code for an assignment, the system notifies the grader and automatically runs test cases that were specified by the grader. The result of the test cases is then sent to the grader. The grader also has the option to add line-by-line comments to the student's code, and run linter programs to verify code quality and style. Finally, the grader will have the option to assign a grade to the student.

The deliverables for this project will be a fully hosted web application. The goal with this application is to greatly reduce the time that it takes for professors to grade assignments, and help communication between students and graders. The web application will also promote good coding practices by giving feedback to the students when they submit their code.

The development milestones for this project will be the finalization of the front end web client, a built RESTful back end server, the completion of the accounts and repositories manager and the culmination of the testing and quality framework. The end goal of the team is to integrate all of these components and finally deploy this service in the cloud.

The total cost for the project will be $50,757.26 of development costs and additional scalable cost for hosting infrastructure in Amazon Web Services EC2. Expected profitability for this product is very high. At $10,000 per educational institution license, the return of investment (ROI) will be nearly balanced to $0 at 5 licenses sold, but the team expects to sell at least 25 educational licenses.

Table of Contents

1. Problem Statement

Most educational institutions manually grade coding assignments completed by students. Manual evaluation of code is inefficient: it takes a great deal of time and is prone to bad evaluations due to the substantial amount of different ways a program can be written. This leads the grader to only evaluate the program's results, i.e. its output, based on an specific input. Such method of evaluation can be maliciously exploited and does not measure code quality at all. Missing the evaluation of code quality can lead professors to pass students with mediocre coding skills. Currently, most professors and teaching assistants evaluate output by manually importing code into a workstation, most probably the grader's own personal machine, and then manually executing the program through some test cases. This is highly insecure because the code is not properly sandboxed [1]. The code could be malicious and cause damage to the running workstation or even worse: get a hold of sensitive data.

Zach Cross [1], a Teaching Assistant (TA) from the University of North Carolina at Chapel Hill, comments when he is asked if he has a grading tool for students code:

> "Nothing well integrated and highly functional, mostly hacky bash/python scripts..."

Moreover, he comments on the current system his institution uses for grading:

> "We don't have a clean web interface or anything and they aren't functional in the sense that they break for most assignments as soon as errors occur e.g. they can grade the 100%'s but not much else."

Jesús E. Luzón [2], a TA from the University of Puerto Rico at Mayagüez, comments when asked about the time it takes to grade a 50 line coding assignment:

> "The way I grade the assignments is by creating a testing project that only includes the student's code. I check out the project from the students repository and then run the code and compare the output. For a well written assignment, I take about 3 to 4 minutes evaluating it. But for an assignment that has errors, it takes me about double the time: from 6 to 8 minutes. This happens because I need to figure out where exactly the error comes from, so that I can give the student some useful feedback."

When asked about how many students he grades, he comments:

[1] In computer security, a sandbox is a security mechanism for separating running programs. It is often used to execute untested code, or untrusted programs.

> "I have 23 students, which doesn't seem like too many, but the time it takes to grade their assignments adds up quickly. In the worst case, I can spend 3 hours grading a single assignment."

The proposed solution makes the grading of coding assignments more efficient by providing an easy to use web-based application that handles submissions and automatic grading of coding assignments. It is expected that the proposed solution will reduce the time it takes to evaluate a 50 line coding assignment by 75%, from a maximum of 8 minutes to a maximum of just 2 minutes. It will also completely eliminate all overhead preparation efforts like setting up individual projects and checking in many different code repositories. It will not only grade based on the program's output but also on code quality and performance. This solves the problem of inaccurate grading by taking into account other factors that measure a student's academic achievements. The proposed solution tackles each of the problems specified and also adds extra value by integrating features that can help increase the student's programming skills, e.g. it provides helpful feedback that promotes better coding practices on the student's submitted code, and it promotes usage of version control systems. This product will directly benefit the Professors, Teacher Assistants (TAs) and the students. It will also indirectly benefit the educational institution that uses it. The project will create a solution that is secure, reliable, and easy to use.

2. Project Antecedents

2.1 Previous Work Experience

The members of Team Aguacate have known each other for several years, many of which have been spent developing projects for several courses and extracurricular activities. They have participated together in programming competitions, hackathons and projects, such as the Microprocessor Interfacing course project on the spring semester of 2013, WaveSphere [3]. These previous experiences have allowed the team to build their teamwork abilities.

The team's participation in programming competition have exposed them to several online judges for programming competitions, such as UVa [4] and SPOJ [5]. Team Aguacate has also gone through the experience of submitting code for programming labs. These previous experiences has lead the team came up with the idea of creating something similar to an online judge to facilitate not only their own work by making a tool to turn in programming assignments easier and getting grades faster, but also the instructors' work that have to grade hundreds of these assignments.

2.2 Similar Projects

Educational Institutions outside of Puerto Rico have similar systems to the one being proposed. In 1994, the University of Maryland presented Kassandra: The Automatic Grading System [6]. As the title suggests, Kassandra is an automatic grading system that is used for grading assignments in scientific computing. This system is used by students to check the correctness of their program assignments. This is achieved by comparing the program output with an expected result. In 2000, the Virginia Polytechnic Institute and the Microsoft Research Corporation developed Curator, a web-based electronic submission that supports automated grading of elementary programming assignments [7]. Its grading mechanism is similar to Kassandra, the programs are compiled and their output result is tested against a correct output from an specific input. A few years ago, in the University of Puerto Rico at Mayagüez (UPRM), a TA for the Data Structures course and former student, José Santuche [8], developed a simple script that would checkout his students' code, run some tests, compare the output and send an email to the student with the results. However, it was unreliable and had to be manually changed between assignments. The most currently competitive product is another web-based tool named Web-CAT. Web-CAT is an open source automated grading system that is used to grade students on how well they test their own code. This is different than just comparing output results. Web-CAT needs to be installed on an institution's own infrastructure [9]. Another similar product is Praktomat [10], an open source web-based tool for automatic grading. This product differs in that it also enables peer- to-peer reviews, but does not have a sophisticated testing framework.

2.3 Proposed Approach

From the aforementioned projects, Panda Code Reviews (Panda) integrates correctness checking and privacy awareness. Panda differs from the previous efforts because it is fully web-based and hosted in the cloud out of the box, so there is no installation needed and the update process is transparent to institutions. Competing products, like Web-CAT, Praktomat, Curator, and Kassandra require additional infrastructure and a painful set up process. They all have the hassle of having to manually back up the database and distressingly updating the infrastructure's software. Moreover, Panda provides the missing feature in all other similar products, the code quality analysis. This means that the tool creates reports of the code's quality that help the students excel in their area and follow best coding practices. It will also include in-line commenting capabilities for enhancing the communication between the graders and students.

2.4 Project Importance

As stated by previous and current TAs, such as José Santuche, it is important to have an automatic grading tool because they become exhausted of having to grade the same

assignment over and over again, it is a slow and irritating process, especially when the graders have hundreds of students. As Jesús E. Luzón mentioned (quoted above), with only 23 students, grading a single assignment might take him up to 3 hours of his time. As Santuche mentions:

> "Since the process of grading is so exhaustive, the graders normally begin by verifying all the details, but at some point they just verify the output without looking at the code."

. From the students perspective, the process of submitting code and waiting for a grade is slow and annoying as well. It is prone to mistakes when sending an e-mail or virus infections if submitting the assignment by some removable and portable storage device. Panda aims to tackle these problems by providing a centralized and fully integrated web-based system in which students will be able to submit their assignments and it will automatically grade them based on code quality and test cases submitted by the TA or professor. The automatic grading will not be applied immediately to the student, since the professor could override the system and use the code quality tools to determine a grade for the student. This tools helps the grader as much as possible, without removing power from the grader. This project will incorporate the use of repositories, a tool that is not much used in UPRM, so that it also helps students to learn how to use version control and source code management technologies. All of these solutions that the tool will provide makes it a great marketable product. Institutions can greatly benefit by purchasing the service of this project.

2.5 Standards and Regulations

Since the tool will manage student data, it needs to be secure and private. It also needs to maintain the test cases uploaded by the instructor private to the students. To maintain the information secure and not visible by anybody, an HTTPS connection will always be required between Panda and the client's computers. All information will be encrypted in the system, including credential information, account data, and repository data. Also, a very sophisticated, third party cloud serving infrastructure will be used to guarantee data backup and up to 99.99% uptime: ensuring that the system will be available always when needed.

3. Objectives

3.1 Objectives Outcomes and Metrics

To build the complete code review system, the team will accomplish the following objectives by November 15, 2013:

- Create a visually modern front end interface that includes an account manager, repository manager, test manager, source code viewer with syntax highlighter, source code annotator and source code quality analyzer.
- Build a testing framework that will run tests for programs written in C++, Java and Javascript.
- Create a test coverage analyzer and integrate it with the testing framework.
- Construct a code quality framework that will include linters for C++, Java and Javascript.
- Create an accounts and git repository back end module.
- Create a cloud based back end REST API for communications between the front end interface and both the testing and code quality frameworks.
- Deploy the system in Amazon Elastic Compute Cloud systems (Amazon Web Services EC2).

3.2 Deliverables

The deliverables for this project will be a cloud based web application with the featured Code Review tool by November 15, 2013. The Code Review tool will be able to highlight syntax, run test cases, and analyze code quality for C++, Java and Javascript source files. It will also be able to run programs with an input specified by the grader, and will be able to compare the program's output with an output specified by the grader. The web application will be fully hosted in a provided cloud infrastructure without additional installation or configuration. Grader and student accounts will be able to be created by the client, and it will be able to be used immediately after launch.

4. General Approach

Due to the nature of project, most of the work will be completed trough developer workstations. At first, the team will be developing different parts of the system as they are all loosely coupled. Test driven development will be done, meaning that after each part gets successfully developed and united tested, only then they are integrated and tested from a bigger perspective. The project is divided into 4 major parts, these are: Front end Client, Back end Server, Test Framework, and Repository Manager, as shown in Figure 1. The team will be using a single cloud hosted Git repository (GitHub) where the project's code will live, and will follow best revision control practices. Moreover the team will follow the Scrum agile development process and will confine to good planning and documentation.

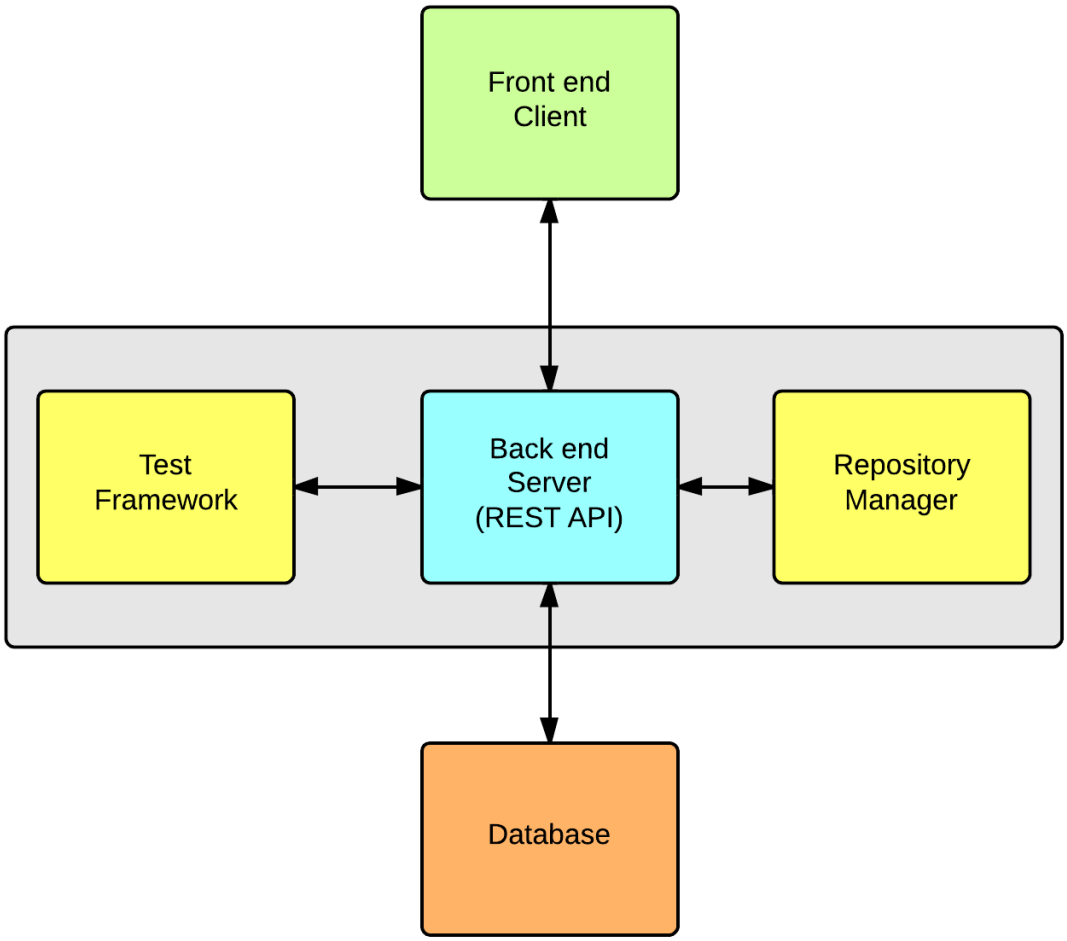

Figure 1: General overview of the system

4.1 Meetings and Work Hours

Team Aguacate will hold weekly meetings on Tuesdays from 3:00 pm to 4:00 pm to discuss progress and project details. In each meeting, each member will state his/her progress, any needs he/she might have and if their schedule has changed. Work hours on assigned tasks will occur every Monday, Wednesday, and Friday from 9:30 am to 11:20 am unless there is a seminar scheduled for that day. Other work hours will be every possible day from 6:00 pm to 9:00 pm, minimum. The team will work on sprints, so a sprint planning meeting will occur on Thursdays from 3:00 pm to 4:00 pm whenever a new sprint comes. Meeting agendas minutes will be posted in the project blog: `http://pandacodereview.wordpress.com/`.

4.2 Team Management

The meetings will be as described above and other meetings will be scheduled when needed. One of the purposes of these meetings is to make sure that each team member is aware of the other members' progress and concerns. Changes to these hours are to be discussed and decisions will depend on majority vote and project manager opinion. Each task will have a leader and an assistant. The leader of each task is responsible for it and the assistant will assist whenever the leader gets stuck on a problem. Whenever a conflict arises, the members involved are responsible of looking for a mediator as soon as possible to help them solve the problem.

4.3 Documentation Standards

See Appendix B to see documentation standards.

5. Schedule

One of the most important things to do when planning a project is to set up the work schedule, divide tasks and estimate time of completion.

5.1 Gantt Chart

A Gantt chart showing the project schedule is included separately with this proposal. Project critical path is shown in red. Usual work schedule does not include Saturdays and Sundays, but some tasks were put on weekends to avoid a bigger critical path. If this was not done, the whole Implementation phase would be on the critical path, and that is where more problems are likely to arise. Weekends work hours might be included or excluded depending on project progress.

5.2 Work Breakdown Structure

The Work Breakdown Structure, showing the tasks hierarchy (outlined), people responsible of completing them, and a time estimation of task completion, can be observed in the Gantt Chart included. Critical path tasks are highlighted. To see the cost associated to task completions see Section 8.

6. Personnel

The most important resource of a project is the personnel. It is important to have people that have the necessary skills to perform tasks inside a project. This section discusses the skills required and how the team will be meeting those necessities. To see the team composition and a small biography of each of the members, look at Appendix A.

6.1 Skills Required

Table 1 shows a list of tasks for this project. Each task has a lead and and assistant, which is also shown in that table. For a more detailed list of requirements and task distribution see Appendix C as well as the Gantt Chart included as a separate file.

Table 1: Task Distribution

Task Title	Nelián	Samuel	Daniel
Project Management and Team Organization	Lead	Assist	
Web Front end		Lead	Assist
Back end Server		Assist	Lead
Test Framework	Assist		Lead
Accounts & Repositories	Lead	Assist	

6.2 Training Needs

For the development of this project, the members of the team are responsible of learning about the new technologies they have to work on, self train themselves and ask for help when needed.

6.3 Consultancy Resources

- Front-End Web Development
  - Consultant Expert: César Andreu
- Teaching Assistant feedback
  - Consultant Expert: Jesús E. Luzón

  - Consultant Expert: José Santuche
  - Consultant Expert: Zach Cross
  - Consultant Expert: Divit Singh
- Database Management and Web Development
  - Consultant Expert: Dr. Manuel Rodríguez

6.4 Legal Requirements and Conditions

For this project, there are no legal requirements or conditions related to personnel needed for the project.

7. Resource Requirements

7.1 Required Resources

- Software Developers (Personnel)
- Computer Workstations (Minimum of 3)
- Workings Desks
- Microsoft Project
- Free Open Source IDE
- Free Open Source Database
- Internet Connection
- Hosting servers for cloud infrastructure (Amazon EC2)
- Consulting

7.2 Resources Provided by UPRM

- 1 Computer Workstation
- Working Desks
- Microsoft Project
- Internet Connection

8. Budget

Accurate estimates of cost is extremely important for ensuring that the team does not go over budget. Since this is a software project, the majority of the cost is attributed to

Software Developers and Human Resources. Development costs remain low as long as computer workstations and an internet connection is provided. At the last step of the development of the project, an Amazon Web Services EC2 account needs to be purchased in order to deploy the system.

8.1 Project Cost

The total estimated cost of the project is $50,757.26. The break down details follow.

8.1.1 List of Components

- 2 Workstations minimum ($0 - Provided by team). The workstations are needed for software development.
- Amazon Web Services EC2 ($528 yearly after deployment). The service is needed for deploying the project. The service is the most reasonable for the project, since it is able to scale automatically and the pricing changes depending on the amount of server usage.

8.1.2 Human Resources

Table 2: Human Resources

Employee	Position	Annual Salary	Hourly Wage	Hours per Week	Working Weeks	Payment
Nelián	Project Manager and Software Engineer	$55,000	$26.44	20	15	$7,932.69
Samuel	Software Engineer	$45,000	$21.63	20	15	$6,490.38
Daniel	Software Engineer	$45,000	$21.63	20	15	$6,490.38
					Total	$20,913.46

8.1.3 Employee Fringe Benefits

Table 3: Employee Fringe Benefits

Sick Leave (0.5 days per month)	Paid Vacation (5 days)	Employee Health Insurance	Employee Fringe Benefits
$800	$2,000	$2,000	$4,800
		Total	$14,400

8.1.4 Overhead Hours

Table 4: Overhead Hours

Task	Overhead Hours	Total Hours (3 members)
Team Meetings	49	147
Writing Reports	96	288
Course Meetings	65	195
Presentations	4	12
	Total Hours	642
	Total	$14,915.80

9. Assessment Methods

To help manage the project and collect information on each member's individual progress on their tasks, the project manager decided to use Zoho Projects. This web tool allows to divide tasks. It also helps to know which tasks are still need to be completed, and lets each team member specify how much they have progressed in a specific task. The team will also have a Git repository where all the code that has been written will be stored, so that progress can also be measured from the amount of code already written. Project reports will also reside in a Git repository. Team Aguacate will work in sprints and each task will have a weight according to their priority, which will be decided in the sprint planning meeting. Project progress can also be measured by the weight of the tasks done.

10. Risk Management

Table 5: Risk Management Table

Risk	Probability	Impact	Actions
Loss of Data	Low	Low	Backup repository will be used to store code and data. Images of VMs will be on the cloud.
Damages to Developer's Workstation	Medium	Medium	University's computers will be used instead. If necessary and budget allows it, developer buys a new workstation.
Downtime in Hosting Infrastructure	Low	High	Work will still be performed on local workstations and as soon as infrastructure is back online, deployment is done with a newer version.
Natural Disaster	Medium	Low	Using revisioned repository, each developer can safely work remotely. If possible, online meetings will be held. If no internet connection is available, and if needed, the team will hold meetings in Samuel's house.
Team Member Becomes Ill	Medium	Medium	If a team member is unable to work, task distribution will be delegated considering task priorities and their ability to block other tasks.
Team Member Withdraws From Class	Low	High	Agreement has been reached that if team member withdraws from class they must still collaborate on the their respective project part. If unwilling to cooperate, task delegation will have to be performed. Higher working hours will be needed.
Out of Budget	Low	Medium	Funding will be needed from a different source. Team's personal money will be used as last resort.

11. Impact

This project aims to impact Educational Institutions, specifically Computer Science, Software Engineering and Computer Engineering instructors that have to manually evaluate code submitted by students. It will impact this community by providing an easy to use solution that will save them time when evaluating student's coding assignments. In the long-term, it will also increase the student's skills in programming areas by giving recommendations on their codes referencing well proven standards and patterns. Moreover by making instructors save time they can be more engaging in other educational activities involving the students. From a different perspective, this project will also decrease the chances of computer viruses being spread trough the former evaluations methods.

11.1 Commitments

The system:

- Will have a database in a remote server.
- Will have a web interface.
- Can manage accounts (create, edit and delete).
- Can receive code submissions.
- Will support C++, Java and Javascript code.
- Can evaluate submitted code.
- Can present results of evaluations.
- Will enable communication between the grader and the student through inline comments.
- Will display linter's results of submitted code in the web application.
- Will automatically evaluate submitted code.
- Will send emails to students and graders when code is evaluated.
- Will be visually appealing.

11.1.1 Limitations

- The web application is not guaranteed to work properly on non supported browsers.
- It will not be able to process programming languages other than the supported ones.
- The web application will not work properly when no internet connection is available.

11.1.2 Legal Commitments

The licenses for the tools that the team intends to use to build this system will be non propagated (i.e not Copylefted), permissive open source licenses e.g. MIT License.

12. Market Overview

The intention of the team is to market this product as a service. It is intended to charge educational institutions a yearly rate based on the amount of students they intend to support.

Income will come from subscriptions and these will be used to pay for hosting infrastructure to support the cloud service, it will also pay for any further work done by developers to update and maintain the cloud service.

12.1 Potential Customers

The product that the team intends to develop is aimed to educational institutions that require evaluation of student's coding assignments. The service can potentially be used by interviewers and peer to peer collaboration in a company.

12.2 Current or Potential Competition

There are some closed code review systems available in various university institutions like University of Maryland's Kassandra [6]. There are some open source products available as well, like Web-Cat [9] and Praktomat [**?**]. There are also many online judge systems like UVa [4] and SPOJ [5]. Additionally, many code linters and style guides exist. All of these competition products are outdated, they all require installation on an institution's own infrastructure, and they don't include any code quality analyzers at all.

12.3 Competitive Advantages

The main competitive advantage of this product is that it provides a service that is cloud and web-based. It is reliable, secure and easy to use. It can throughly analyze code, not just by comparing its result but by reviewing coding styles. Additionally, this system does not include any installation or configuration at all.

13. References

[1] Z. Cross, "Linkedin profile." http://www.linkedin.com/in/zachcross/.

[2] J. E. Luzón, "Linkedin profile." http://www.linkedin.com/pub/jesus-luzon/61/335/982/.

[3] A. Ildefonso, D. Santiago, N. Colón, and S. Rodríguez, "Wavesphere." http://www.ece.uprm.edu/micro2/index.php/WaveSphere, 2013.

[4] "Uva online judge." http://onlinejudge.org.

[5] "Sphere online judge." http://www.spoj.com/.

[6] urs von Matt, "Kassandra: the automatic grading system," in *ACM SIGCUE Outlook*, vol. 22, pp. 26–40, jan. 1994.

[7] "Curator." http://courses.cs.vt.edu/curator/.

[8] J. Santuche, "Linkedin profile." http://www.linkedin.com/in/joserisan/en.

[9] "Web-cat." http://web-cat.org/group/web-cat/.

[10] "Praktomat." http://sourceforge.net/projects/praktomat/.

Appendices

A. Team Aguacate Members Biographies

A.1 Nelián E. Colón Collazo

Nelián E. Colón Collazo was raised in Orocovis, Puerto Rico. She is a senior student currently pursuing her bachelor's degree in Computer Engineering at the University of Puerto Rico at Mayagüez. Nelián has a passion for programming, that is why she is currently specializing in Software and is always participating in programming competitions. She has been involved in undergraduate research both in her University and in the University of Texas at El Paso, and has done two summer internships at Honeywell Aerospace and Harris Corporation. Nelián also likes to be involved in leadership positions. She is currently the ACM-ECE's treasurer and the Tau Beta Pi - PR Alpha's Computer Engineering Representative. Appart from being a programmer, Nelián is also a musician. She plays the Puerto Rican Cuatro and Tiple with her well-known father, Edwin Colón Zayas.

A.2 Samuel A. Rodríguez Martínez

Samuel A. Rodríguez Martínez was born Chicago, IL, United States of America. He moved to San Juan, Puerto Rico when he was 2 years old. While at San Juan he developed his passion for software development, and enrolled for a Computer Engineering B.S. degree at the University of Puerto Rico at Mayagüez. While at the University he has been involved in undegraduate research, and was given the opportunity to perform an undegraduate research project at the University of California at Berkeley. He has also done two internships at Google Inc. in Mountain View, CA, and is currently finishing his final year at the University.

A.3 Daniel A. Santiago Rivera

Daniel Santiago is currently a computer engineering student at the University of Puerto Rico, Mayagüez. He is passionate about programming and interested in mobile, backend, and web development. Daniel likes working with emerging technologies such as Node.js and MongoDB. In his free time, he enjoys cooking, running and assembling gaming computers.

B. Documentation Standards

B.1 General Standards

Application must meet the following general standards:

- Must have a low learning curve for the targeted audience.
- Must use the same styling in every page of the web application.
- Must adhere to industry best practices.
  - Good logging systems for more maintainable code and error correction.
  - Must be secure and protect user's confidential information.
  - Submitted code must be properly sandboxed for protecting the system.
  - Must adhere to HTTP standards.

B.2 Minimum Browser Standards

Web application must function and display properly in the following browser versions:

- Internet Explorer 10
- Safari 4
- Chrome 7
- Firefox 4

B.3 Minimum Hardware Requirements

- Must be the same as the browser's minimum hardware requirements.
- In addition the user must be connected through and internet connection of at least 128kbps speed.

C. Required Skills

Table 6: Task and Skills Matching

Task Title	Skills Required	Lead	Experience
Project Management and Team Organization	People skills and leadership	Nelián	Leadership positions in student organizations, leader in previous class projects
Web Front-end	Knowledge of HTML, JavaScript, graphics and web standards	Samuel	Google internship experience in front-end web development
Back-end Server	Knowledge of HTTP and Restful API, MongoDB and Node.js	Daniel	Freelance projects, hackathons and past internship experiences exposure to emerging web technologies
Test Framework	Knowledge of test driven development with code standardization and Coverage, compilers, processes and threads	Daniel	Past internship experience exposure to code quality and performance analysis
Accounts & Repositories	Knowledge of repositories and Data Bases	Nelián	Database Systems course, and version control and source code management technologies exposure