Lab 1 – TPOT Product Description

Team Black

Aaron Walden

CS411W

Professors Janet Brunelle and Hill Price

April 2, 2014

Version 2.1

Table of Contents

1 INTRODUCTION 3
2 PRODUCT DESCRIPTION 5
2.1 Key Product Features and Capabilities 5
2.2 Major Components (Hardware/Software) 6
3 IDENTIFICATION OF CASE STUDY 8
4 PRODUCT PROTOTYPE DESCRIPTION 13
(This space intentionally left blank.) 14
4.1 Prototype Architecture (Hardware/Software) 15
4.2 Prototype Features and Capabilities 18
4.3 Prototype Developmental Challenges 19
REFERENCES 25

List of Figures

Figure 1. TPOT Major Functional Components 7
Figure 2. Current Process Flow: Deliverable Assessment 9
Figure 3. Current Process Flow: Status Determination 10
Figure 4. Improved Process Flow: Deliverable Assessment and Feedback 11
Figure 5. Improved Process Flow: Status Determination 12
Figure 6. TPOT Prototype Major Components. 17

List of Tables

Table 1. Feature Comparison Between Real World Product and Prototype. 15

1 INTRODUCTION

The post-secondary educational environment has been transformed by this century's proliferation of powerful and inexpensive electronic communication technologies. College student enrollment in online courses has increased from 10% in 2002 to 30% in 2013 (Allen & Seaman, 2013). This enrollment boom places a special burden upon instructors of online, project-based courses, such as those of graduate-level Engineering Management programs. These instructors are regularly managing 15 teams or more as part of a single course, often asynchronously (P. Pazos, personal communication, October, 2013). There exist software solutions which facilitate collaboration between team members, but these programs give no consideration to the instructors who must manage the development of these teams. As a result, already overburdened instructors face even heavier workloads as the number of students enrolling in online courses grows (Little, 2012).

Virtual teams, like these student teams, are known to be highly flexible and efficient, which results in increased productivity (Ebrahim et al., 2009). They face a special set of difficulties, however, which stem from their non-local nature. Virtual teammates must deal with a lack of information, such as the absence of non-verbal cues during communication, relative to their local counterparts. This deficit leads to slow development of trust, differences in expectations, less effective measures of accountability, and more difficulty in the resolution of conflicts (Ebrahim et al., 2009). The instructor, alone, is tasked with mitigating these issues in furtherance of the development of the team skills of their students.

TPOT, the Team Project Organizational Tool, was developed to address the laborious nature of virtual student team management and the lower levels of information available to virtual teammates. TPOT addresses the challenges of asynchronous team management by providing a dashboard through which instructors can manage all of their teams. The TPOT Instructor Dashboard, written by Old Dominion University (ODU) CS411 Team Black, draws information from an existing collaborative tool developed at ODU for use by Engineering Management students. The TPOT system will create an interface between that tool and the dashboard, giving instructors a powerful way to interact with their students. TPOT is an open source project and available free to any non-profit educational institution.

(This space intentionally left blank.)

2 PRODUCT DESCRIPTION

TPOT is a web-based software system through which students collaborate on group projects and instructors mentor those projects. At the user level, TPOT consists of two major modules: a team site tool, which students use to share their work, and an Instructor Dashboard, which reads data from the team sites and organizes it for instructors. The team site tool is a highly-customized Google Site containing scripts which are able to record user's interactions with the site in a database. Actions recorded include visits to pages, modifications of tasks, and the sending of messages. The Instructor Dashboard is a separate website that reads from said database and allows the instructor to browse student data without having to organize it manually. The dashboard also provides instructors with an efficient means of administering courses and interacting with students. By combining these two modules, TPOT aims to provide students and educators with a comprehensive tool for maximizing the effectiveness of distance learning.

2.1 Key Product Features and Capabilities

The most important and innovative feature of TPOT is its Instructor Dashboard. While a myriad of different project management software suites exist, TPOT is the only solution that offers a feature designed specifically to aid instructors of virtual teams. The dashboard gives instructors the ability to administer teams as a batch process.

The Instructor Dashboard presents information via a tabbed web interface. Each tab addresses a different element of team administration. Team data is displayed in customizable tables which allow sorting and filtering. An Overview tab presents the instructor with a summary of the latest team site activity, new messages, and items in need of grading. A Tasks tab contains information regarding all tasks assigned by the instructor and their status. A Files tab gives the instructor access to all files uploaded to the each team site. An Activity tab displays a record of all site activity. This tab is significant in that it provides instructors with a powerful way of holding students accountable for their work. If one team member claims to have participated actively in a group assignment, the instructor can see clearly whether or not that is true. A Messages tab provides a system by which instructors can communicate with students without having to leave the site. A Grades tab allows instructors to assess deliverables without having to search for them and provide feedback in a single operation. Finally, an Admin tab allows instructors to manage the creation and deletion of their own courses, teams, and Teacher's Assistants.

In support of the Instructor Dashboard are the Google Apps Scripts developed by the TPOT team and embedded within Google Site template pages. These scripts create interfaces which allow students to manage their projects by creating and tracking their own tasks. They also allow students to message each other without needing to run another application. The function of the Google Apps Scripts is to ensure that when students engage in these activities, the data produced can be read and interpreted successfully by the Instructor Dashboard.

2.2 Major Components (Hardware/Software)

TPOT is software as a service. As Figure 1 illustrates, TPOT consists of three major functional components: the Instructor Dashboard website, a series of team sites, and a database. There are also supporting hardware components. The Instructor Dashboard site is hosted independently by the developers of TPOT but may also be installed and maintained on a private server by a customer. Thus, the web server hosting the dashboard site is a hardware requirement. The site is written in PHP and JavaScript, which allow it to display content dynamically.

Team sites are created from a previously developed Google Site template when an instructor creates a new course. The templates may be customized by the instructor to fit his or her course's needs. These sites allow students to collaborate on their projects by creating web pages and by sharing files, comments, and tasks. Team activity takes place on these sites, and embedded in them by the TPOT developers are a number of Google Apps Scripts which record site activity in the TPOT database. Team sites require no hardware, as they are hosted and maintained by Google.

The final software component of TPOT is its database. The TPOT database is an instance of MySQL server running on a 3rd party database server. The database is a critical component of TPOT because it facilitates communication between the team sites and the Instructor Dashboard. Team data is recorded in the database by the team site scripts and read by the Instructor Dashboard.

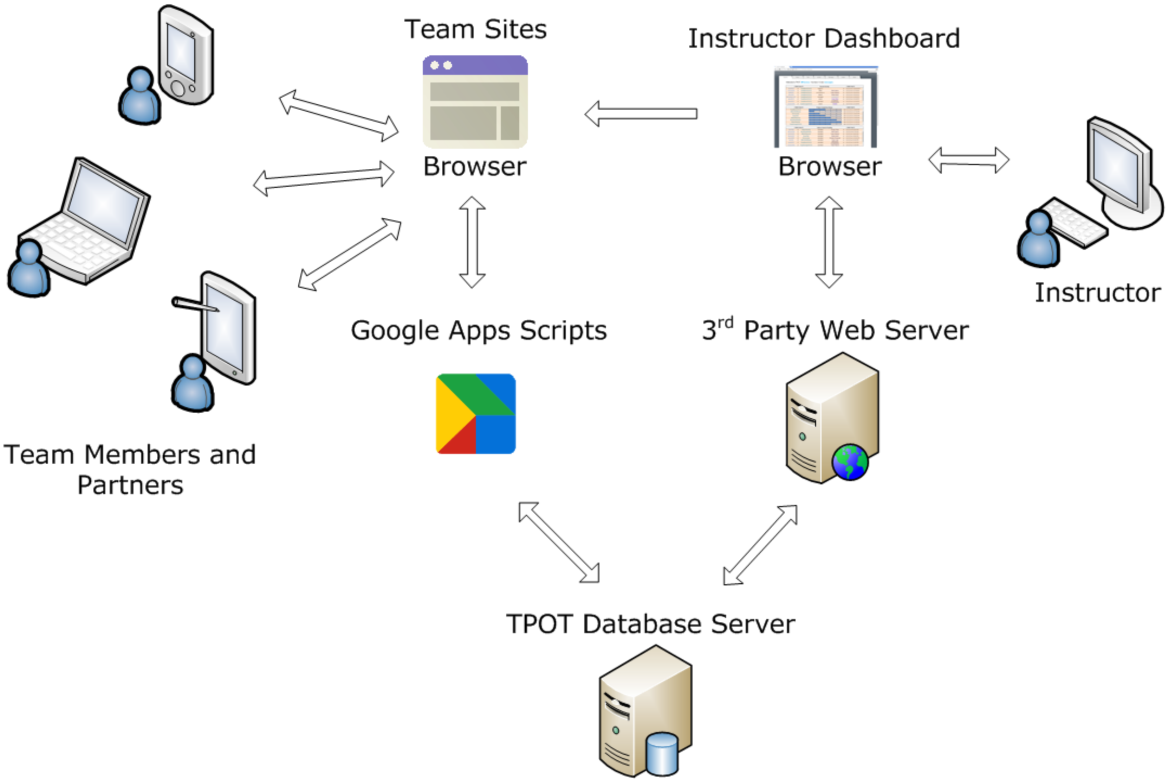

Figure 1. TPOT Major Functional Components.

3 IDENTIFICATION OF CASE STUDY

The project which became TPOT was originally proposed by Dr. Pilar Pazos, Assistant Professor in ODU's department of Engineering Management and Systems Engineering. Dr. Pazos, an avid researcher of virtual teams, developed the Google Site tool for her own use as an instructor of such teams. The tool proved successful in enhancing student collaboration, but was limited in its support of instructor activities. Dr. Pazos identified team status determination and deliverable assessment as two duties which she found to be problematically cumbersome for instructors (P. Pazos, personal communication, October, 2013).

(This space intentionally left blank.)

Figure 2 details the laboriousness of deliverable assessment. Lacking a centralized solution for deliverable submission, instructors and teacher's assistants must search through multiple file repositories to find the correct files. These repositories may contain different versions of files, introducing uncertainty about intended submissions. After assessing items, graders must provide feedback to students as a separate process, involving yet another program.

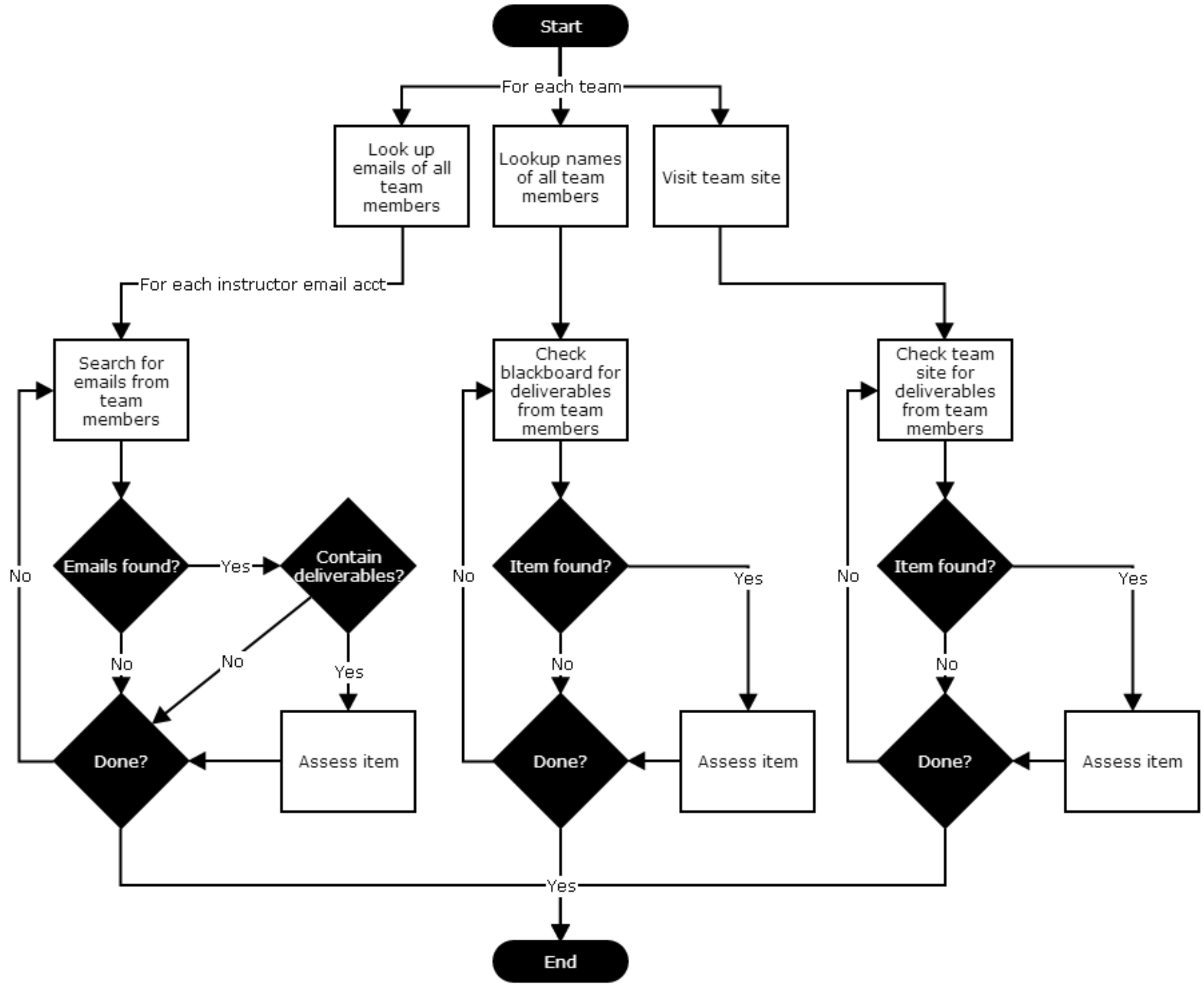
Start
For each team
Look up emails of all team members
Lookup names of all team members
Visit team site
For each instructor email acct
Search for emails from team members
Check blackboard for deliverables from team members
Check team site for deliverables from team members
Emails found?
Yes
Contain deliverables?
No
No
Yes
No
Done?
Assess item
Item found?
Yes
No
No
Done?
Assess item
Item found?
Yes
No
No
Done?
Assess item
Yes
End

Figure 2. Current Process Flow: Deliverable Assessment

As Figure 3 reveals, team status determination is fraught with repetitive actions. To determine status, instructors must navigate to multiple web pages for every team under their tutelage, recording status as they go. Minimally, instructors must also search through their email accounts for messages from students which may convey some notion of team status.

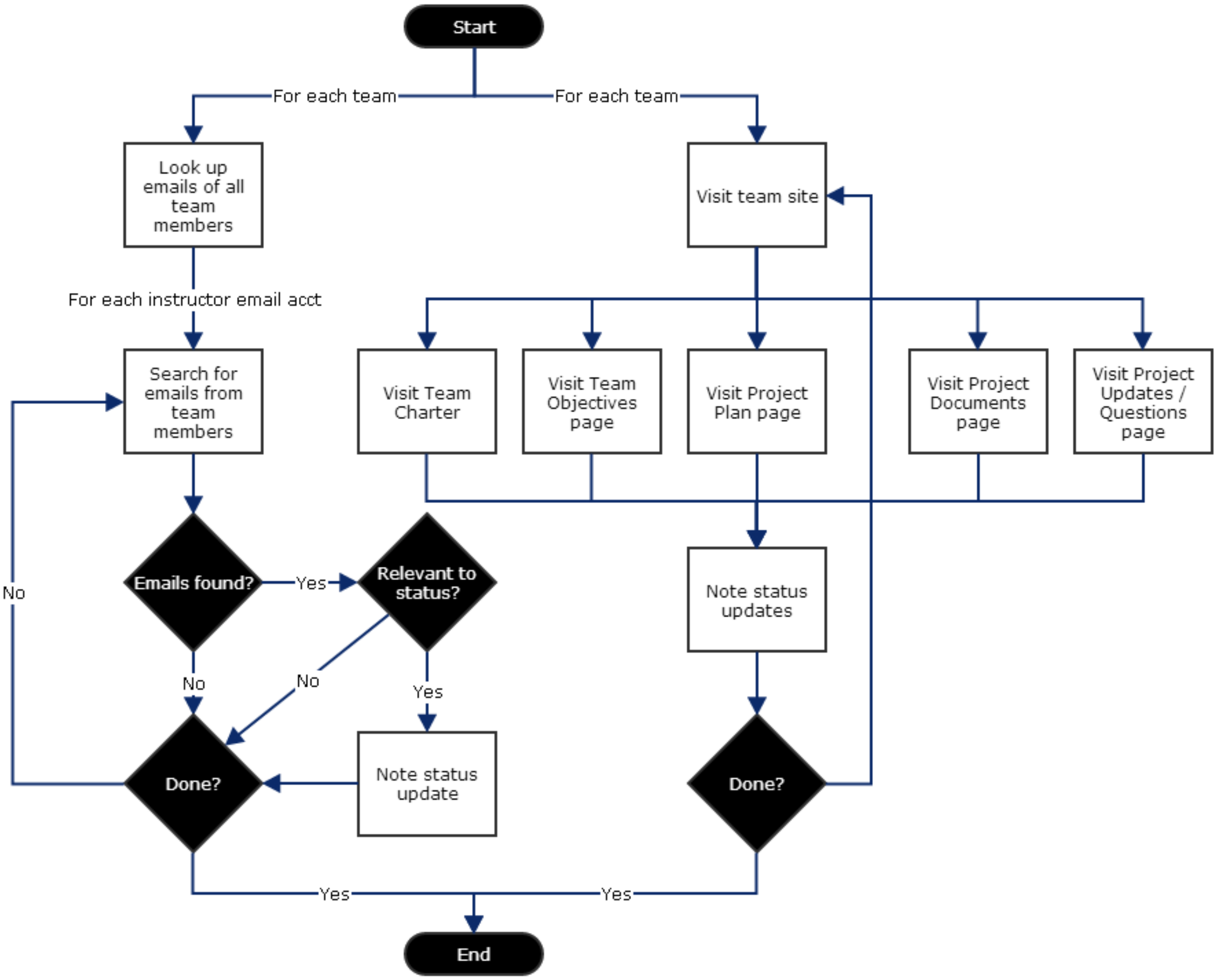
Start
For each team
For each team
Look up emails of all team members
Visit team site
For each instructor email acct
Search for emails from team members
Visit Team Charter
Visit Team Objectives page
Visit Project Plan page
Visit Project Documents page
Visit Project Updates / Questions page
Emails found?
Yes
Relevant to status?
No
Note status updates
No
No
Yes
Done?
Note status update
Done?
Yes
Yes
End

Figure 3. Current Process Flow: Status Determination

(This space intentionally left blank.)

To solve the problem, Dr. Pazos proposed to ODU's Computer Science senior undergraduates an Instructor Dashboard which could read and aggregate data from her team sites. The implementation of this idea is TPOT. By making all team data available in a single interface, TPOT reduces the more complex task shown in Figure 2 to the simpler process of Figure 4. Additionally, the process of providing feedback is folded into assessment to further reduce instructor and teacher assistant workload.

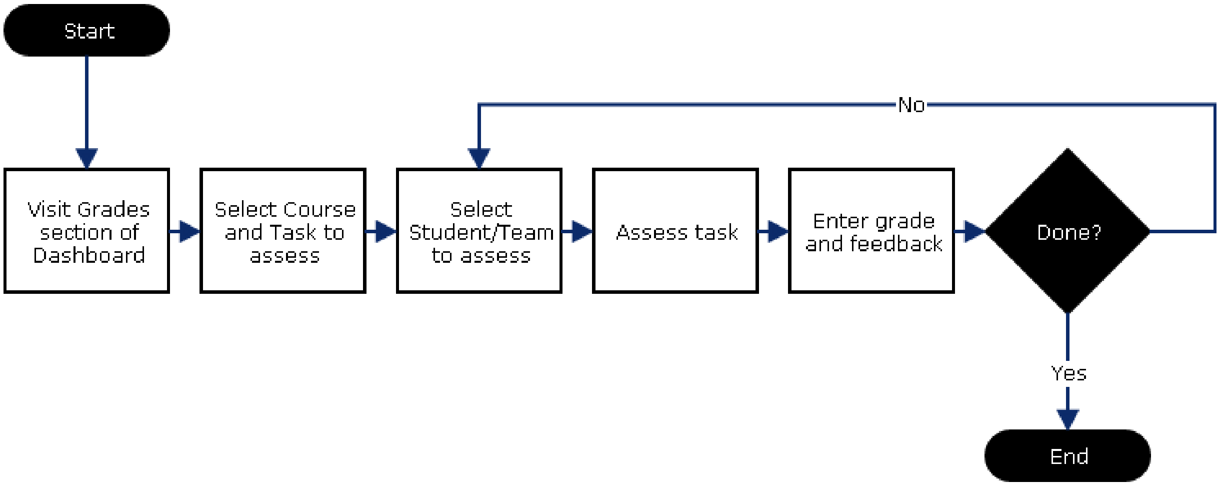
Start
Visit Grades section of Dashboard
Select Course and Task to assess
Select Student/Team to assess
Assess task
Enter grade and feedback
Done?
No
Yes
End

Figure 4. Improved Process Flow: Deliverable Assessment and Feedback

(This space intentionally left blank.)

Similarly to the condensing of assessment, the process of status determination is reduced to visitation of a single page on the Instructor Dashboard, as Figure 5 shows. Instructors and teacher's assistants will no longer need to visit every page of a team site in order to determine what progress a team has made. Dr. Pazos agreed that the proposed improvements brought by TPOT would mitigate her problem to a great extent (P. Pazos, personal communication, December, 2013).

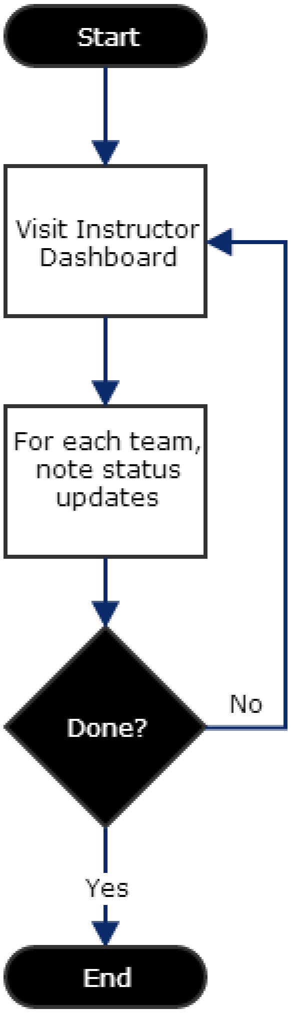
Start
Visit Instructor Dashboard
For each team, note status updates
Done?
No
Yes
End

Figure 5. Improved Process Flow: Status Determination

4 PRODUCT PROTOTYPE DESCRIPTION

The ultimate goal of the TPOT prototype is to demonstrate that an aggregative dashboard built atop team sites will augment an instructor's ability to develop students' virtual team skills. The prototype will achieve this augmentation by carefully structuring and monitoring team site data such that it may be collected algorithmically. Once collected, team site data will be presented to instructors and students in ways which provide optimal support for the achievement of their respective goals. This overarching goal may be understood as a series of more specific functional goals.

One functional goal of the TPOT prototype is a reduction in the amount of labor required to administer online project-based courses. Laborious activities include project status determination, deliverable assessment, feedback provision, and task assignment. The TPOT prototype will reduce the time required to determine project status by providing instructors with a single web page that displays a team's recent site activity, task progress, and questions to the instructor. Project status is then integrated, instead of being spread over a number of team pages and emails. The TPOT prototype will reduce the workload of deliverable assessment and feedback provision by combining the two, using an interface which presents all submitted deliverables alongside forms for entering both grades and feedback. The labor of task assignment will be reduced by endowing instructors with the ability to create a task with a simple form and simultaneously assign that task to any number of teams or individuals. Free to concentrate on more important work, instructors will be able to provide students with higher quality education.

Another functional goal of the TPOT prototype is to provide team members with a wide breadth of information regarding their projects. Students with a stronger understanding of their project's status may make more effective decisions regarding those projects. The TPOT prototype will provide concise status to all members of a project as discussed elsewhere in Section 4. Additionally, the task system will ensure that team members are aware of what assignments are to be completed, by when, and what a student's role in the completion of these assignments entails. Instructors and students may also be made aware of a project's status through notifications made possible by a system of database triggers. When a user-defined trigger condition is met, an email or other specified alert will inform the creator that the trigger condition has occurred. Additional information regarding the triggering event may be provided in the alert message. The TPOT prototype's activity tracking system will allow team members to be aware of every action taken that affects their projects. It will also allow them to know with confidence the contribution levels of other team members. A goal of the TPOT prototype is to provide team members with enough information to render difficult the development of an incorrect view of the status of their projects.

Another functional goal of the TPOT prototype is to allow team members to access their projects on as many different devices as possible. Virtual teams are productive due in part to their great flexibility (Ebrahim et al., 2009). It follows, then, that maximizing flexibility by removing device restrictions should be a priority.

(This space intentionally left blank.)

4.1 Prototype Architecture (Hardware/Software)

The TPOT prototype is planned to have nearly the same structure and feature set as the real world product. Table 1 shows a direct comparison between the two. The major differences are the prototype's reliance on ODU virtual machines for hosting and its use of simulated data.

Features/Components	Real World Product	Prototype
Student Team Members	Actual students participating in online project-based coursework	Actual student testers and TPOT team members using virtual test machines with past project data and randomly-generated data
Instructor	Instructors leading online project-bases courses	Professor Pazos as well as TPOT team members simulating the instructor role
Team Sites	The Google Site tool created by Dr. Pazos with embedded Google Apps Scripts which structure the data teams generate	The same tool and scripts
Instructor Dashboard	A tabbed, fully customizable dynamic web page which concisely presents data drawn from the team sites	A tabbed dynamic web page which concisely presents data drawn from the team sites
Google Apps Scripts	A series of JS-like scripts which collect, structure, and store in the TPOT database the data generated by teams which interact with the team sites	The same scripts
3rd Party Web Server	An independent web server hosting the Instructor Dashboard	An ODU-owned virtual web server hosting the Instructor Dashboard
TPOT Database Server	An independent database server hosting the TPOT database	An ODU-owned virtual database server hosting the TPOT database

Table 1. Feature Comparison Between Real World Product and Prototype

The Instructor Dashboard will be hosted on an ODU virtual web server instead of an independent one. The TPOT database will similarly reside on an ODU virtual database server. Students and instructors will test TPOT with their own machines. A virtual testing environment will also be created in order to test TPOT using an array of different operating system and browser combinations. The remainder of the prototype largely mirrors the planned real world product.

The TPOT database will take the form of a virtual instance of MySQL server. In addition to providing the schema which will structure the data drawn from the lower-level sites, the TPOT database will contain a series of triggers written in SQL which will alert users to changes in the status of their projects. These triggers may call stored database procedures which gather data from the database and email users. Such stored procedures will be written in PL/SQL.

The Instructor Dashboard front-end will be written in HTML, PHP, and JavaScript. It will present a tabbed interface of customizable table views and update forms. The dashboard back-end will consist of a series of SQL queries and updates embedded in PHP. These queries will populate the table views which provide aggregate information to instructors.

The existing team site page templates will be altered in such a way that they will provide structured data which records the work of team members. These alterations will include the embedding of Google Apps Scripts that connect to the TPOT database and update it with collected data. HTML and Google Apps Scripts will also be used to create GUI elements where necessary in the updated team site templates, such as forms for the creation of tasks.

(This space intentionally left blank.)

The TPOT database will be populated by a combination of automatically-generated data and data resulting from use by students and instructors. The data generator will be written in PHP and consist of automated SQL updates. A larger scale testing program must also be constructed which opens numerous connections with the database and makes randomly-generated queries in order to ensure that the database can handle concurrent use. Additionally, a Scenario Simulator will allow the TPOT developers to change the effective date recognized by the software, in order to simulate the passing of a semester for demonstration purposes. These features can be seen in Figure 6.

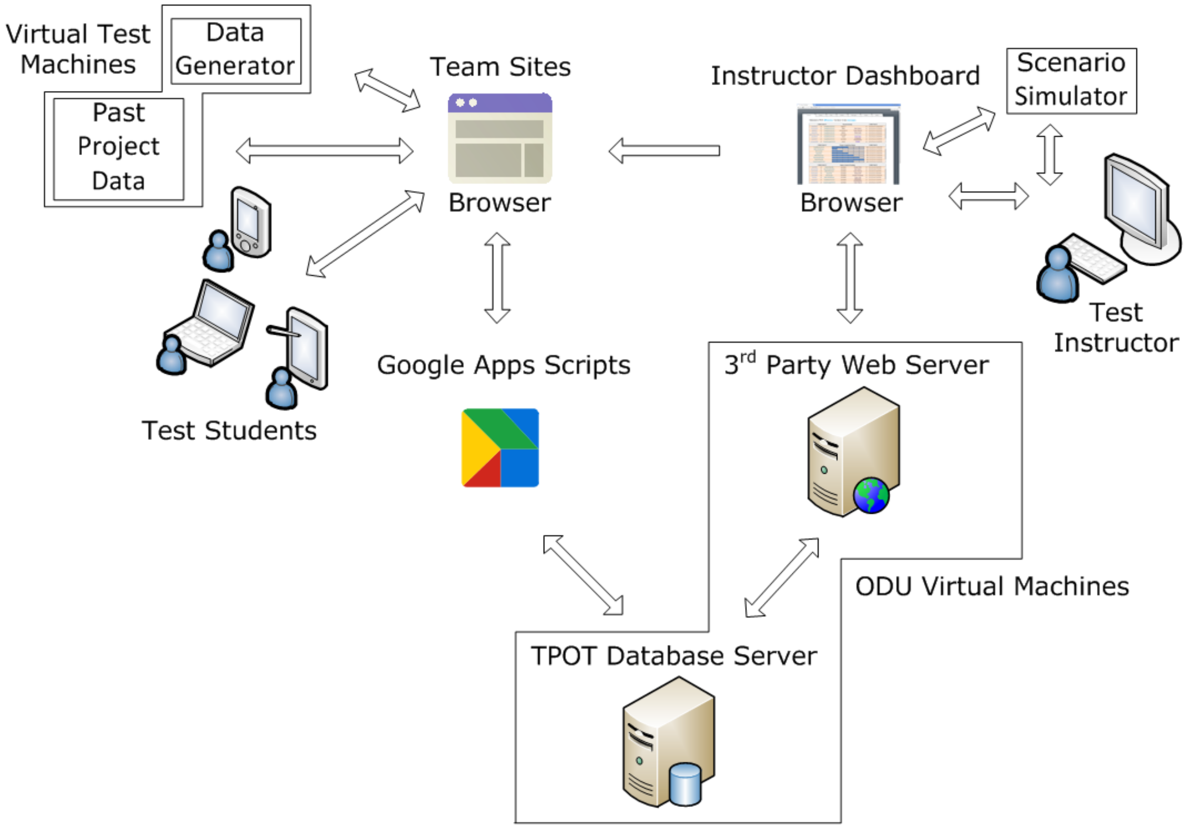

Figure 6. TPOT Prototype Major Components.

4.2 Prototype Features and Capabilities

The primary innovative feature that the TPOT prototype will demonstrate is the Instructor Dashboard and its ability to disencumber instructors of virtual project courses. The dashboard gives instructors the ability to determine team project status, assign tasks, assess deliverables, and provide feedback in a manner less labor-intensive than is currently practiced. With fewer administrative tasks to occupy them, instructors will be free to aid students in the development of their virtual team skills.

Another major feature that will be demonstrated by the TPOT prototype is improved access to project information. With concise status updates, comprehensive team activity tracking, and instant update alerts, student users of the TPOT prototype will have a clear picture of a project’s status and their role in that project. As the major obstacles inherent in virtual teamwork involve lack of information, the TPOT prototype’s abundance of information will mitigate this key problem.

Additionally, the TPOT prototype will demonstrate its flexibility. The prototype's team sites will be accessible by desktop computers, smartphones, tablets, and as many different browser and operating system combinations as possible. The benefits of virtual teams extend partially from their great flexibility. The ability to collaborate using any capable device will maximize the flexibility of users’ virtual teamwork and the benefits reaped from that flexibility.

In order to ensure that the TPOT prototype effectively improves collaboration, it will be tested by real students and instructors. Real project data will be used during testing whenever possible. Once it is established that the TPOT prototype functions correctly with real project data, large scale testing with randomly generated data will proceed in order to stress-test the software.

4.3 Prototype Developmental Challenges

One challenge which must be overcome in the development of the TPOT prototype is the issue of multiple logins. The Google Sites, upon which TPOT will be built, require either a Google account or an ODU account in order to access them. The issue is that users of TPOT must also be uniquely identified in the TPOT database. It would be preferable to avoid forcing users to create TPOT accounts in addition to Google or ODU accounts. The challenge to be overcome then, is to determine whether or not a unified login is achievable.

Another challenge in the development of the TPOT prototype is the issue of security. Because senior and graduate project courses may involve industry partners and have real economic stakes, securing the intellectual property developed during the course of a project is essential. Additionally, communication between instructors and students should be confidential, as the exposure of such missives may become a source of intra-team conflict. It is often stated that such risks will be handled by the application of best security practices during development. The challenge is to determine what exactly these best practices are and how to best apply them to the prototype.

An additional challenge in the development of the TPOT prototype is the question of how to structure the user-interface. Interfaces may be constructed in any number of different permutations. Care must be taken to develop an interface which is intuitive and attractive while retaining maximum effectiveness. If the interface is not attractive enough to users, TPOT will not be adopted and its other features, no matter how well-designed, will go unused. If the interface is not practically effective, TPOT may hinder instead of help users achieve their project goals, and the likelihood of adoption decreases. The challenge is to mitigate these risks and the approach will see TPOT developed iteratively with constant input from actual student and instructor users.

One more challenge in the development of the TPOT prototype is to ensure maximum accessibility to both the team sites and the Instructor Dashboard. As explained, maximizing the flexibility of virtual teamwork will maximize the benefits proven to come with such flexibility, such as increased productivity. The challenge is testing site accessibility on the wide array of devices that are in use today, and the strategy will be to test frequently using a suite of virtual machines which run the most popular operating systems and browsers available.

(This space intentionally left blank.)

GLOSSARY

Activity Tab: The tab on the TPOT Instructor Dashboard that tracks team member activity.

Admin Tab: The tab on the TPOT Instructor Dashboard that provides management of an instructor's courses, teams, and teacher's assistants.

API: Application programming interface that specifies how some software components should interact with each other.

Asynchronous Course: A course which does not hold regularly scheduled meetings.

Back-end: The portion of a computer program not visible to end-users.

Beta test: The final test of a software product, usually fully-featured, prior to commercial release.

Blackboard: Extremely popular course management software.

Cloud: A type of computing involving data and computation distributed over large numbers of dispersed machines which are connected via a network.

Collaboration Software: Software suites designed to organize the collaborative efforts of a professional team.

Dangling Pointer: Pointers that do not point to a valid object of the expected type. These are special cases of memory safety violations.

Deliverable: An item produced by a project team which is intended to be delivered to a person outside of the team.

File Cabinet: A default template page of a Google Site which facilitates the management of user-uploaded files.

Files Tab: The tab on the TPOT website that assembles all files uploaded by students to their team sites.

Front-end: Portion of application with which end users interact with directly.

Google Apps Scripts: A scripting language, based on JavaScript, developed by Google which runs exclusively on Google servers.

Google Sites: A tool offered by Google which allows users to create web pages and manage files and collaboration through a GUI without requiring any programming language skills.

Grades Tab: The tab on the TPOT website that allows for viewing and assigning of grades for individuals and teams.

GUI: Graphical user interface. Allows users to interact with electronic devices through graphical icons and visual indicators instead of a textual command line.

HTML: Hypertext markup language. A language designed to create web pages.

Industry Partner: A business or government professional working with or mentoring a school project team.

Instructor Dashboard: The module of the TPOT software suite which aggregates student data for display to instructors and teacher's assistants.

JavaScript: A scripting language commonly used by web browsers to dynamically alter content.

Messages Tab: The tab on the TPOT website that allows an instructor or teacher's assistant to send messages to the students they mentor.

Method: A function associated with a class of objects.

MySQL: An open-source relational database management system.

ODU: Old Dominion University, a public research university located in Norfolk, Virginia.

Overview Tab: The tab on the TPOT website that tracks recent activity and upcoming events.

Penetration Test: A deliberate attempt by a benign entity to find security flaws in a system.

PHP: PHP Hypertext Processor. A server-side scripting language designed to dynamically create web pages.

Platform: A hardware architecture and a software framework (including application frameworks), where the combination allows software to run.

PL/SQL: Procedural Language/Structured Query Language. A programming language which combines the declarative SQL with an imperative language to achieve more traditional functionality while retaining strong database support.

Pointer: A variable which contains a literal memory address.

Project Management Software: See Collaboration Software.

Scrape: A computer software technique of extracting information from websites.

SQL: Structured Query Language. A programming language designed to interact with relational database systems.

Tab: A navigational widget for switching between sets of documents.

Tasks Tab: The tab on the TPOT website that allows a user to view, create, assign, modify, and delete tasks.

Team: A collection of students working on the same project coordinated through TPOT.

Team Activity: The various ways in which Google Sites can be used. This may include visiting pages, creating and editing pages, uploading files, sending messages, or creating tasks.

TPOT: Team Project Organizational Tool. Software being developed in response to a team management problem posed at ODU.

User: A student, instructor, teacher's assistant, industry partner, or master administrator.

Virtual: Forms of interaction, lacking physical manifestation, which are mediated through electronic communication.

Virtual Machine: Software-based emulation of a computer system.

Virtual Team: A team dispersed through time and space which collaborates primarily electronically.

Web-based: Software which may be accessed through a web browser and need not be downloaded nor installed.

REFERENCES

Allen, I. E., & Seaman, J. (2013). *Changing Course: Ten Years of Tracking Online Education in the United States*. Retrieved from http://www.onlinelearningsurvey.com/reports/changingcourse.pdf.

Ebrahim, N. A., Ahmed, S., & Taha, Z. (2009). Virtual teams: A literature review. *Australian Journal of Basic and Applied Sciences*, *3*, 2653-2669.

Little, R. College Professors fearful of online education growth. *U. S. News & World Report*. Retrieved from http://www.usnews.com/education/online-education/articles/2012/07/06/college-professors-fearful-of-online-education-growth