Swipe In, Tap Out:
Advancing Student Entrepreneurship
in the CIS Sandbox

Conner Charlebois
charleb_conn@bentley.edu

Nicholas Hentschel
hentsch_nich@bentley.edu

Mark Frydenberg
mfrydenberg@bentley.edu

Bentley University
Computer Information Systems Department
Waltham MA 02452

Abstract

The Computer Information Systems Learning and Technology Sandbox (CIS Sandbox) opened as a collaborative learning lab during the fall 2011 semester at a New England business university. The facility employs 24 student workers, who, in addition to providing core tutoring services, are encouraged to explore new technologies and take on special projects to support or enhance the day-to-day operations of the CIS Sandbox. Doing so creates a culture of entrepreneurship and encourages innovation among the student workers and their peers. This paper follows up on previous results by describing a student-initiated development project to rewrite a card-swipe desktop application that tracks student usage, as an integrated suite of web and mobile apps. The paper presents the architecture of the new system along with perceptions from the student developers around their motivations to contribute to the CIS Sandbox technology infrastructure. Initial use of the prototype suggests it will improve productivity among tutors and provide faculty with easier access to tutoring data.

Keywords: web development, mobile development, computer lab, tutoring, innovation, entrepreneurship

1. INTRODUCTION

The Computer Information Systems Learning and Information Technology Sandbox (CIS Sandbox) has become a valuable campus destination for learning about and with new technology, through the in-person peer tutoring services and online resources it provides. Staffed by 24 undergraduate and graduate student assistants, the facility has applied startup values to sustain its day-to-day operations and encourage innovation. In addition to providing tutoring to undergraduate and graduate students as a core service, student workers take on administrative responsibilities of managing, promoting, and planning activities for

the facility. On several occasions, they have also identified improvements to processes in place.

Previous research describes the results of transforming a computer lab into a collaborative environment (Frydenberg, 2013a) that supports both learning and technology. This redesigned physical space reflects the BYOD (Bring Your Own Device) paradigm of today's connected world, in which most students use their laptops and mobile devices to access the university digital resources and the Internet, and perform computing tasks.

Combining tutoring and technology in the same facility has been effective. Usage reports indicate that the number of students who came to the CIS Sandbox for tutoring increased as the role of the computer lab shifted. The lab was no longer a place where students simply could access technology. The CIS Sandbox has become a campus destination for students to learn to use the technology available to them to achieve their goals while simultaneously evoking further interest in technology.

The culture in the CIS Sandbox fosters a sense of ownership and innovation among student workers who are encouraged to explore new technologies and find ways to integrate them into the day-to-day operations of the facility (Frydenberg, 2013b). The CIS Sandbox relies on a variety of social and digital media tools (Frydenberg, 2013c) as part of its infrastructure.

This paper describes a web and mobile software development project proposed by two undergraduate CIS Majors to improve the process of managing student usage statistics and documenting tutoring cases in the CIS Sandbox. They began this project in their junior year as part of their work in the CIS Sandbox.

2. ENCOURAGING INNOVATION AND ENTREPRENEURSHIP IN THE CIS SANDBOX

The opportunity to learn about web and mobile application development tools enables students to develop marketable skills that the information technology industry seeks (Huang, Kvasny, Josh, Trauth, & Mahar, 2009). Providing students the opportunity to develop real-world software applications has been shown to be a valuable experience to Information Systems students (Abrahams & Singh, 2011; Wong, Pepe, Stahl, & Englander, 2013; Su, Jodis, & Zhang, 2007). Such projects enable students to "practice their content knowledge and workplace skills while working on authentic, contextualized projects." (Dunlap, 2005). Developing entrepreneurial skills involves giving students opportunities to take initiative and ownership of projects they want to work on. (Bilen, Kisenwether, Rzasa, & Wise, 2005) (Faltin, 2001)

The CIS Sandbox is a collaborative learning facility that supports university courses through tutoring and student exploration of technology through the availability of computing equipment and environments, career presentations, and incentives for innovation. It strives to provide students with learning opportunities that "enabl[e] students to perceive the world, and especially ... technology, as a learning opportunity space. This technology can be used for critical inquiry that allows them to develop as humans and as professionals" (Senges, Seely Brown, & Rheingold, 2008, p. 126)

Inspired by an employment model at Google, (Mediratta & Bick, 2007) student workers who take on expanded roles are given one hour per week apart from their scheduled tutoring responsibilities to work on special projects.

Student workers in the CIS Sandbox have identified and taken on a variety of projects indicative of innovation and entrepreneurship. These include creating instructional video tutorials on using software applications taught in CIS courses, planning extracurricular workshops with industry leaders who speak about career opportunities and technology topics, designing and managing the CIS Sandbox WordPress, blogging on technology topics, promoting the facility using social media, and creating engaging demonstrations for fellow students.

It is worthwhile to note that opportunities for student entrepreneurship can extend to learning centers in other disciplines. Tutors cannot just be students who have mastered the required material; they must be knowledgeable, personable and ambitious. By giving tutors both the opportunity to follow their own passions and access to resources with which they can succeed, the CIS Sandbox has demonstrated a model from which other labs can learn: to have students achieve great things, let them.

3. RECOGNIZING AREAS FOR IMPROVEMENT

One area where student workers saw a need to improve was the way that the CIS Sandbox

gathers usage statistics about its users. When the CIS Sandbox opened in fall 2011, university support staff implemented, as a side project, a Visual Basic desktop application to record usage metrics. Students, upon entering the facility, swipe their ID cards using a magnetic card reader. The swipe application records the date and time of their arrival along with student's name and email address to an Access database. The application obtains the user's identity information by passing the ID number read from the ID card to a student directory database. The swipe application also displays a personalized welcome message to the student, confirming the scanner reads the student's card properly. Students view the acknowledgment, along with photos of tutors on duty, on a front-facing monitor as they enter (Figure 1).

Figure 1. Swiping in at the CIS Sandbox.

As they finish working with students, tutors complete an online tutor form in which they record student's name, class, instructor, and comments about the nature of the help provided. Because tutors easily could hand the tablet to students to enter their names and class information before completing the session summary themselves, most tutors found tablet devices to be conducive in completing this form.

Each week, a CIS Sandbox student worker reviews both the swipe usage data and the tutor form data, to prepare reports informing faculty about students in their classes who come for help and the type of questions they asked, along with overall usage statistics. These reports provide a link between what students are learning in the classroom and what they are learning in the CIS Sandbox. The swipe data shows periods when the facility is at its busiest; these patterns also inform decisions around staffing and scheduling.

Faculty use student usage statistics to design their lessons, see the questions students are having, and recognize students that are "going the extra mile" for their classes. Reports of students who seek assistance and their questions inform the instructors of what students are learning in the lab, and enable faculty to better cater to students' individual needs.

Improving the Swipe System

Obtaining usage data relied on accessing two disparate systems that did not share data electronically. The swipe system lived as a desktop app whose data was only accessible from one computer in the room. The tutor form application existed as a Google Form which required asking students for personal information already known to the swipe system in order to complete.

The reporting capabilities also needed a user-interface overhaul and more automation. As it stood, a CIS Sandbox student worker imported the raw data into Excel manually each week in order to generate charts and reports, and then send them individually to each professor in the department. The process usually took over an hour. This task seemed like it could be automated easily.

The original system was not user-friendly to setup and maintain, often requiring numerous resets during a given shift. Each time the front-end swipe application was started, the application window had to be moved to the front-facing monitor, where a clicking a button on the app would maximize its window on the screen in which it was displayed. Once the app was running, there was no easy way to update the information displayed. For example, in addition to capturing student swipe information, the Welcome screen also displayed pictures of tutors who were currently on duty. These tutors' pictures updated only when the next tutor on duty swiped in, and his or her photo appeared over the previous tutor's photo. Updating tutor pictures and information was no easy task. The process could be simplified by allowing tutors to swipe in as workers (rather than as students), and having the app determine who is currently working by retrieving this information from a database.

Furthermore, tutors were required to carry an additional employee ID card with a different ID number to distinguish between times when they were working and times when they were using

the lab as a student. The original system was unable to differentiate between students who were tutors and students who were not, even though the data allowed making this distinction.

Improving the Tutor Form
The original tutor form was implemented as a Google Form intended to be viewed from a desktop or laptop computer despite the fact that most tutors found it easiest to complete it using a tablet or their own mobile devices. Filling out the form requires the tutor to enter information that he, or the students he tutors might not know (such as student's name, the section number of the classes they are taking), some of which could have been previously recorded from their ID card upon swiping in (such as the student's name).

4. PROPOSING A UNIFIED WEB AND MOBILE SOLUTION

Tutors recognized deficiencies in the current system, but were unable to make any improvements to it because the source code was unavailable. Two student tutors took it upon themselves to create a new, unified web and mobile app that combined the best features of both the original swipe system and the tutor form, and added new capabilities.

The new app would employ a touch interface making it simple for students swiping in, responsive pages which will adapt to the laptops, tablets, or smartphones on which tutors view or complete the tutor form, and a web interface for faculty checking their students' progress. The new version, necessarily rewritten from the ground up because of the lack of access to the original source code, is all open source, and can be easily modified for future expansion and development of new features. The student-developers encourage their successors in the CIS Sandbox to tinker with the source code to see what cool new features they can implement.

This design approach allows the tutors and students to make better use of some of the new technology the CIS Sandbox has to offer, such as an HP all-in-one touch screen computer running Windows 8, and Google Nexus and Asus tablets. It also enables tutors to work effectively from their smart phones.

Student workers identified key areas and new features to improve upon prior functionality in the new system. These included improved statistics and reporting, a better-integrated tutor form, and a wait queue to inform tutors on duty of students who swiped in and need help, in order to acknowledge them quickly and easily.

Statistics and Reporting
Because the new system records the times when students swipe in and out, as well as the time when the tutor submits the tutor form, deeper insights into how students are using the CIS Sandbox are available. It becomes possible to determine how much time tutors spend tutoring, and gather the most common questions they receive. For example, if students are spending a long time working on an Excel assignment with pivot tables, comments in the tutor forms will reflect these issues, and a faculty member could decide to spend more time going over the assignment in class. The new system also informs tutors how long a student has been waiting for help, and how much time tutors have been spending with individual students. These statistics are invaluable metrics to the CIS Sandbox in determining efficiency and effectiveness of the tutoring staff.

The new system also allows running a report of hours that all tutors worked during the previous pay period, and notifies individual student tutors via an automated, customized email message of the hours they swiped in during the previous pay period. This will reduce errors when students enter their hours worked on the university's time card reporting system in order to get paid.

Tutor Form
The new and improved tutor form requires only one input from the student being tutored: the time at which they take their IT class. Through working with students, tutors found that most students do not know the section numbers of their classes, but they do know when their classes are held. The form pre-populates the student's ID, name, and course tutored from data obtained when the student swipes in and specifies a course number for which they are seeking help. The only work left for the tutor is to enter a brief description of the help provided during the session. This is a marked improvement over the previous system, and allows tutors to spend more time tutoring, and less time performing administrative tasks.

Wait Queue
One of the most troublesome pieces of feedback received from visitors to the CIS Sandbox since

its opening is that the wait times for a tutor can be long, especially at peak times during the semester, or that students do not feel that tutors are seeking them out when they need help. There are usually between one and three tutors on duty at a given time, depending on the time of day and expected usage at that time. The original system had no way to track how long students were waiting, or inform the tutors which students were waiting for help.

To alleviate this, the new system makes a few key changes that shift the responsibility to seek out students needing assistance back to the tutors. Each time a student swipes in using the new system, it adds the student's name and the course for which the student would like help to a wait queue, along with the amount of time that the student has been waiting.

This queue is available to all of the tutors, accessible via a browser on any mobile device or laptop. Monitoring this screen, the tutors can acknowledge those students who need help and have been waiting the longest, allowing them to initiate the contact with students. This feature also demonstrates the power of the web as a development platform allowing the tutors to connect to a central location and view live-updating data simultaneously on their own devices.

5. DEVELOPMENT TECHNOLOGIES AND PROCESS

This application is built on the web, for the web. It makes heavy use of the traditional LAMP stack (Linux, Apache, MySQL, and PHP). Of those main components, Linux is the operating system of choice; the Apache web server handles the serving of the pages, MySQL powers database, and PHP is the server-side scripting language. The application also relies heavily on the Code-Igniter MVC framework for PHP. This framework keeps the application organized and coherent for future maintainers who may enhance it after the current developers graduate. On the front end, the jQuery JavaScript library assists in making numerous AJAX calls, allowing major sections of the app to be generated and updated without refreshing the page. Including the Twitter Bootstrap CSS allows for rapid app development with sleek and intuitive UI elements.

In keeping with the open nature of the CIS Sandbox, the application was developed to be entirely open source. One of the issues with the

original swipe system was that the code was inaccessible. Despite the need for changes, the tutors did not have the access to make them. Now, with all of the code managed with the git version control system and stored on Github[i], all of the tutors will have access to the code, enabling the system to grow and change with the CIS Sandbox.

Application development follows the standard tenets of the software development life cycle (SDLC). The student developers met to discuss the failings of the current swipe app, and ways to meet the needs of the tutors, students, and professors. One of the student developers describes their development process:

"We conducted interviews with current tutors, and talked extensively with the Sandbox director about the shortcomings of the old system and what features and functionality would return the most value to the users. This requirements gathering and legwork gave a solid foundation on which to build the new system. We designed the application based on the requirements generated and our respective technical backgrounds. Development was a series of sprints (features to develop within a given timeframe) with progress tracked in the web-based project management application Trello. At the end of each sprint, the development team would meet to debrief and select the next feature set to build. We had regularly scheduled meetings with the CIS Sandbox director to review our progress and the roadmap for upcoming development. We developed and tested using localized Apache servers while seeking hosting on a university server for production-level testing and deployment. We are also working on a full deployment plan and on how best to document the project for future maintainers."

6. ARCHITECTURE AND DATA FLOW

Appendix 1, Figure 1 shows the architecture and data flow of the Swipe In, Tap Out application.

Swipe In
A user swipes his or her ID card (1) using a magnetic card reader attached to a front-facing touch screen display running a web app displaying the Swipe In page shown in Appendix 1, Figure 2.

The app reads the user's ID from the card, and looks up the student's name from a student ID database (2) provided by the university's department of administrative computing.[ii] To ensure the security of student IDs, the database stores student ID numbers encrypted with the SHA-1 algorithm (Handschuh, Knudsen, & Robshaw, 2001) which the university's data security officer indicated would provide ample security. The app encrypts the student's ID obtained from the card using same algorithm, and matches that value with the one on file to retrieve the student's name. The student's name displays in a customized Welcome screen shown in Appendix 1, Figure 2.

The Welcome screen asks a user if he or she wants help. If the user taps Yes, the app displays a list of classes offered during the current semester, dynamically obtained by scraping the university registrar's website of course listings, as shown in Appendix 1, Figure 3. The user taps the class for which he or she is coming for help, and the app records this information into its database (3).

If a student is not here for help, the Welcome screen asks if he or she is an employee and signing in to work, in order to track the start or end of a tutor's shift.

Wait Queue Page
Tutors access a Wait Queue page (4) shown in Appendix 1, Figure 4 on their laptops or mobile devices to view the names of students waiting for help. Knowing who is waiting for help puts the onus on the tutors to reach out to students, rather than the other way around. After helping a student, the tutor taps or clicks the Tutor Form button near a student's name shown in Appendix 1, Figure 5. The tutor form loads, pre-populated with information already known. The only information required from the student is the time (or scheduling block) when the student takes the class in order to determine the student's instructor.[iii] That the tutor form is also responsive so tutors can fill it out from their own mobile devices encourages the BYOD culture in the CIS Sandbox.

Tap Out
When students are ready to leave the CIS Sandbox after tutoring or completing their work, they now will tap out on an iPad mounted on the wall near the door as they exit (5). Appendix 1, Figure 6 shows the Tap Out page, which displays

the student's first name, first initial of the student's last name, and an "I'm gone" button. This allows keeping a real-time list of everyone in the room, while also capturing the exact amount of time that each student was in the room, how long it took them to get the help they needed, and how long they stayed in the CIS Sandbox afterward.

Administrative Pages
Future development calls for administrative pages, where instructors may access a web app to view usage statistics, or detailed information from tutor forms describing tutoring cases about their students.

A CIS Sandbox assistant may access an administrative web app to perform tasks such as generating reports, downloading data, and entering tutor information.

API
Future plans also call for the development of application programming interfaces (APIs) to enable external apps to incorporate data stored in the Swipe App database. For example, efforts are underway to create a WordPress plugin to display on the CIS Sandbox home page the names of tutors who are currently on duty.

7. STUDENT MOTIVATION

Student workers recognize the culture of entrepreneurship that pervades the CIS Sandbox. They are encouraged constantly to suggest new ideas and experiment with new technologies. Because students and tutors use the swipe system every time they work, it was easy to pinpoint exactly what needed to be changed, and how to improve upon the original system.

Said one of the tutors who took on the development of this system:

> "We knew that the information systems of the swipe app and tutor form needed to be consolidated, and the entire system needed to be open, allowing easy modifications and for expansion down the road. So we paid close attention to succession planning throughout our planning and analysis process, and began the early stages of development. Once we had a proof of concept, we presented it to the CIS Sandbox Director, who commended our work so far, and encouraged us to continue on our own. He didn't try to steer us one way or another.

His style of 'management by enablement' really is what allowed us to do this. In addition to working on this project when the lab was quiet, he offered to pay us for part of the time we spent on this project outside of working hours, and offered to help us obtain computing resources we needed. It was as if we became the project managers, and he was working for us, offering to support us while not impeding our progress. In doing so, he evoked in us a great feeling of ownership and, ultimately, a sense of accomplishment in what we had built. That feeling of being able to do with this project whatever we wanted was a big factor in our continued efforts."

8. NEXT STEPS AND CONCLUSION

The initial prototype has implemented Steps 1 - 5 of the system's capabilities as described in Appendix 1, Figure 1. Development will continue to provide additional administrative capabilities.

Based on the promise this application has shown, and demonstrations to university administrators, the CIS Sandbox has been approached by several other learning labs on campus about using this system in the future. Although designed to meet the requirements of the CIS Sandbox, the application easily could be modified to suit the needs of facilities such as the Math Lab, the Academic Advising Center, and the Accounting and Economics Lab, each of which host students for tutoring, and could benefit from a student queuing and reporting system.

In the rapidly changing world of technology, there is always room for improvement. The students who created this system have identified potential areas for new development. For example, the app could benefit significantly from the development of a REST backend. Accessing the database through a REST API would enable further modularization of the application, easily separating the swipe system core from the administration backend. It would also allow for a simplification of the data model.

On the front-end, much of the interaction with the user is handled through AJAX requests routed through a controller to call methods in the application's models. This could be simplified significantly with a front-end Java-Script framework like Backbone.js or Angular.[iv] These frameworks would add more structure and coherence to the application, and allow it to operate from a single web address and appear much more like a standard desktop app.

The management style in the CIS Sandbox is effective in promoting innovation and exploration of new technologies. Creating the Swipe In, Tap Out app provided a practical development project and real world experience for two student workers who were empowered to recognize ways to improve efficiency in the CIS Sandbox, and build a system to accomplish their goals.

9. REFERENCES

Abrahams, A. S., & Singh, T. (2011). A 'Rainmaker' Process for Developing Internet-based Retail Businesses. *Information Systems Education Journal , 9* (2), 14-26.

Bilen, S., Kisenwether, E., Rzasa, S., & Wise, J. (2005, April). Developing and Assessing Students' Entrepreneurial Skills and Mind-Set. *Journal of Engineering Educatino* , 233-243.

Dunlap, J. C. (2005). Problem-Based Learning and Self-Efficacy: How a Capstone Course Prepares Students for a Profession. *Educational Technology Research and Development , 53* (1), 65-85.

Faltin, G. (2001). Creating a culture of innovative entrepreneurship. *Journal of International Business and Economy, 2*, 123.

Frydenberg, M. (2013a). Creating a collaborative learning community in the CIS Sandbox. *Interactive Technology and Smart Education, 10* (1), 49-62.

Frydenberg, M. (2013b). Fostering Entrep-reneurship in the CIS Sandbox. *Information Systems Education Journal , 11* (3), 35-41.

Frydenberg, M. (2013c). Aligning Open, Physical, and Virtual Spaces in the CIS Sandbox. In A. R. Tatnall, *IFIP Open & Social Technologies for Networked Learning* (pp. 121-130). New York: Springer.

Handschuh, H., Knudsen, L., & Robshaw, M. (2001). Analysis of SHA-1 in Encryption

Mode. In *Topics in Cryptology: CTRSA 2001* (pp. 70-73). Springer Berlin Heidelberg.

Huang, H., Kvasny, L., Josh, K. D., Trauth, E. M., & Mahar, J. (2009). Synthesizing IT Job Skills Identified in Academic Studies, Practitioner Publications, and Job Ads. *SIGMIS CPR '09: Proceedings of the special interest group on management information system's 47th annual conference on Computer personnel research* (pp. 121-128). New York: Association for Computing Machinery.

Janz, K., & Owen, S. (2004). Organizationally Supporting Innovation in TechnologyEnhanced Instruction and Research. *Proceedings of SIGUCCS '04* (pp. 202-208). Baltimore: Association for Computing Machinery.

Lennon, R. G. (2012). Bring your own device (BYOD) with Cloud 4 education. *SPLASH '12 Proceedings of the 3rd annual conference on Systems, programming, and applications: software for humanity* (pp. 171-180). New York: Association for Computing Machinery.

Mediratta, B., & Bick, J. (2007, October 21). *The Google Way: Give Engineers Room.* Retrieved June 1, 2012, from The New York Times: http://www.nytimes.com/2007/10/21/jobs/21pre.html

Senges, M., Seely Brown, J., & Rheingold, H. (2008). Entrepreneurial learning in the networked age: How new learning environments forster entrepreneurship and innovation. *Paradigmes Journal of the Catalan Ministry of Innovation, Universities and Enterprise , 1* (1), 125-140.

Su, H., Jodis, S., & Zhang, H. (2007). Providing an integrated software development environment for undergraduate software engineering courses. *Journal of Computing Sciences in Colleges , 23* (2), 143-149.

Wong, W., Pepe, J., Stahl, J., & Englander, I. (2013). A Collaborative Capstone to Develop a Mobile Hospital Clinic Application Through a Student Team Competition. *Information Systems Education Journal , 11* (4), 39-50.

Editor's Note:

This paper was selected for inclusion in the journal as an ISECON 2013 Meritorious Paper. The acceptance rate is typically 15% for this category of paper based on blind reviews from six or more peers including three or more former best papers authors who did not submit a paper in 2013.
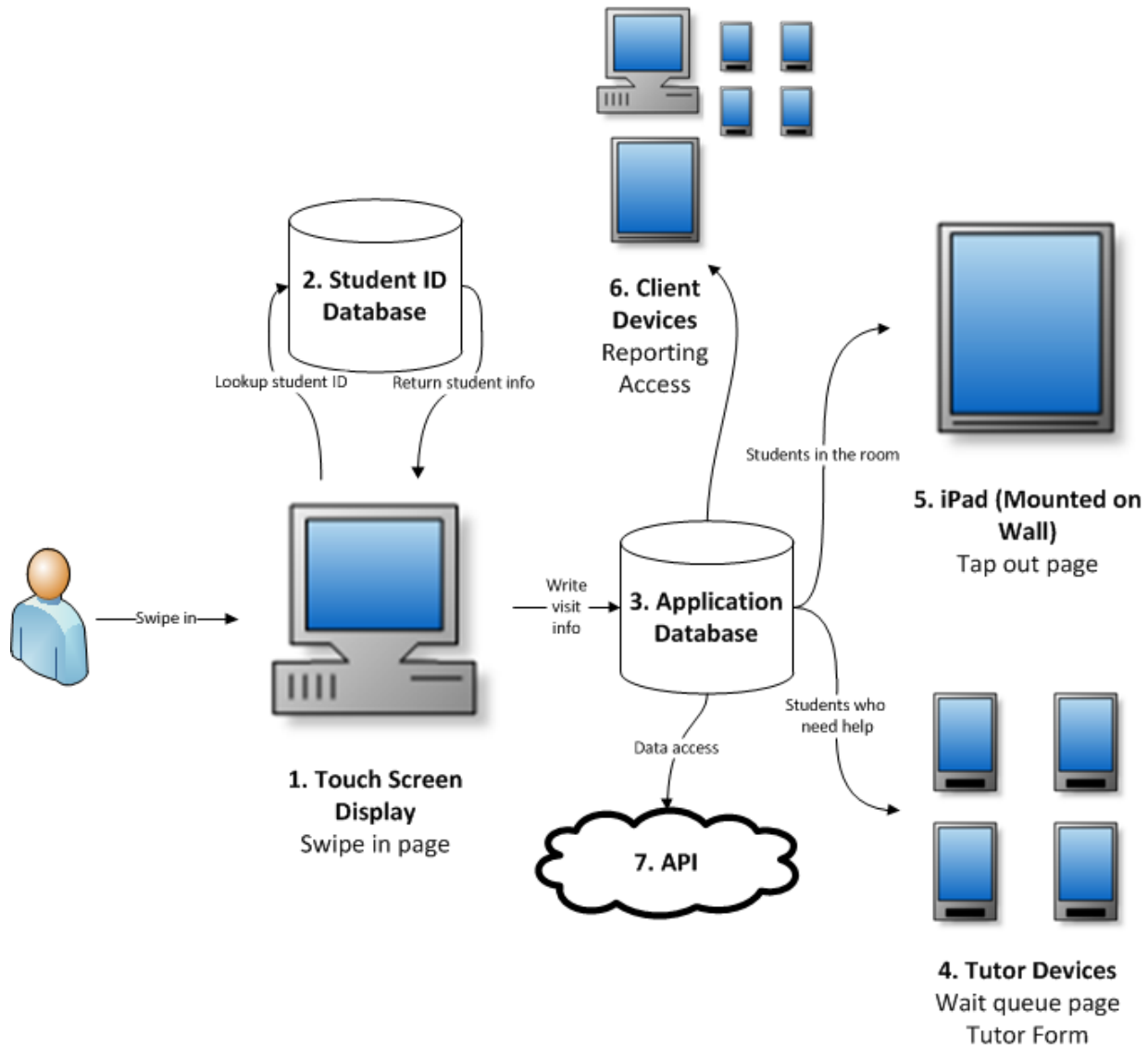
Appendix I

Figure 1. Architecture and data flow of the Swipe In / Tap Out Application.
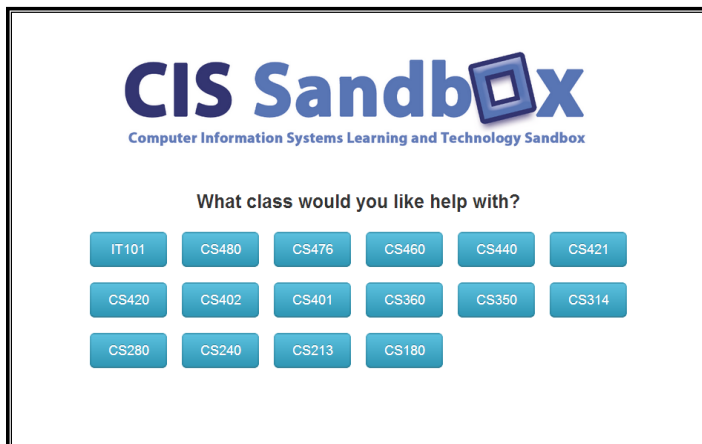
Figure 2. The Welcome screen.

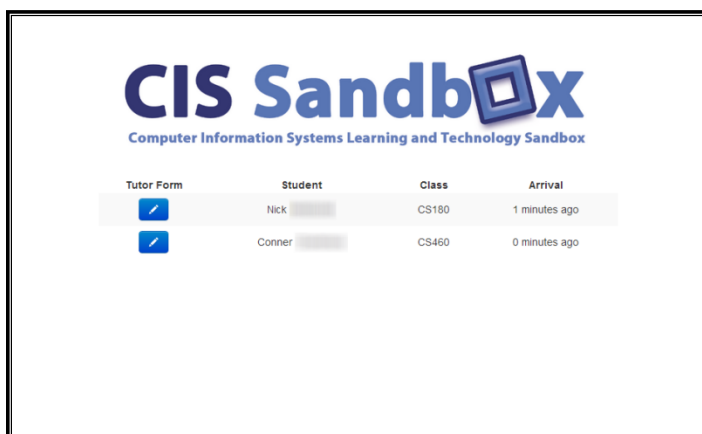Figure 3. The *What would you like help with?* screen.

Figure 4. The *Wait Queue* Page.

Figure 5. The Tutor Form, populated with already-available information.

Figure 6. The Tap Out screen as displayed on an iPad mounted by the door.

Endnotes

[i] The source code is maintained at https://github.com/cisSandbox/sandbox-swipe-system.

[ii] It would have been more elegant if the student developers could have access to an API from the administrative computing department's student database, but this was not available. The compromise was to have Administrative Computing automatically push a flat file containing updated encrypted student ID numbers to the server hosting this app on a monthly basis. When received, the app updates its Student ID database.

[iii] A current limitation is that the app cannot handle the case if two different sections of the same course are scheduled to meet in the same block. This occurrence is rare at the University, so this disambiguation capability is not included in the prototype.

[iv] Angular and Backbone.js are two examples of front-end JavaScript frameworks. Information about each is available at http://angularjs.org/ and http://backbonejs.org/ respectively.