WINTER 2013 • Volume 34, Number 2

CUR Focus

Samuel B. Fee, Christian Griffith, *Washington & Jefferson College*

Custom Mobile Applications for Collection of Field Data

To enhance the research experience of undergraduate students at Washington & Jefferson College (W&J), we have developed a mobile application to collect field data regarding salamanders and myriapods from the Abernathy Field Station near our campus in southwestern Pennsylvania. The goal of the project was not to simply recreate on a mobile device the paper method of collecting data, but also to create a process that would assist students in learning about the proper methods for collecting data and the importance of collecting and analyzing valid data.

One element that makes this mobile app, named Appernathy, particularly interesting from a technical perspective (and extremely useful in the field) is that it is has been written to work without wireless connectivity. The Abernathy Field Station is located outside the range of most wireless service providers. So instead of students writing data continuously to an online database, all the data are collected and stored locally on the device. Later when the students or faculty have access to Internet connectivity, the data are uploaded to an online repository.

As developers, were also able to create an application that guides students through the data-collection process and provides instructional content regarding the field research activity. This replaces any need to have a printed guide for collecting data in the field, since all of that information resides inside the application itself, and can be accessed by students at any time—perhaps most importantly while students are actually collecting data in the field.

Figure 1. Sample Screens on the Appernathy app

To develop the mobile application to collect field data for courses in biology and environmental studies, three academic departments— the Departments of Biology, Environmental Studies, and Computing & Information Studies—worked together. The resulting app assists students in learning the proper methodology for data collection and the importance of ensuring the validity of data (Figure 1).

Because it operates independently of Internet connectivity, this app provides an inventive model for researchers who need to collect data in remote locales or under conditions in which connectivity may be unreliable. In addition, we have implemented data validation through a number of different mechanisms to ensure the validity of data entered into the central database.

The design of the application arose directly from specific research processes at the Abernathy Field Station. Using their knowledge of these processes, the student and faculty developers were able not only to write an application geared to those processes, but also to create an application that guides students through the data-collection process and provides instructional content regarding the field research activity. This collaborative work between both faculty and students in various departments at W&J was the central element in the successful creation of Appernathy.

Applying Appernathy

W&J's Departments of Biology and of Environment Studies regularly collect information about salamanders and myriapods (millipedes and centipedes) at the Abernathy Field Station. There, two areas containing multiple data-collection points, called transects, are set up in which researchers can track the species and numbers of salamanders and myriapods over time. For years, these data have been collected in paper form as researchers walked each transect, and then the data were re-entered manually into a database. Over the last two years, we have prototyped, tested, and revised Appernathy to handle the data collection and its uploading to a server accessible via the Internet. We have also written a web interface for the data so that our students (as well as researchers anywhere in the world) can query the database for their own analytical needs.

Navigating the terrain at the field station can be arduous and sometimes confusing, especially for undergraduates who are new to field research and who may need guidance

CUR Quarterly
COUNCIL ON UNDERGRADUATE RESEARCH

to ensure that their data collection has been systematic and complete. Appernathy is designed to help student researchers understand how to move through transects, and it guides data collection through a process called Full Transect, which is the main function of the application. The two transects at the Abernathy Field Station are identified as Red or Blue. Each transect contains 20 "board" locations. Each board is lifted to find the resident salamanders and myriapods. A full transect is completed when data from each of the 40 board locations is collected. The first step of this process asks users to enter their name and the transect in which they are collecting data (red or blue), followed by their starting location (either 1 or 20). After entering their name, course, and starting location, users can proceed to enter data (Figure 2). Users cannot proceed further in the application without this identifying information, which also makes it possible to revisit the personnel, date, and time should the database administrator ever find any errors in the data.

Figure 2. Screen for "Transect" Data Entry

After entering the identifying information, users are taken to the Salamander Selection page (Figure 3). In the top corner of this page, as well as on every following page, users can refer to their current location and transect, and match that with the transect location they are observing in the field. Pictures of common salamander species are available to users so that they can observe any salamander that they are seeing in a transect and compare it with the pictures in the application. If more information about a species is needed, users can click on the "Information" button, and receive a concise description of that species. If they are able to identify a salamander, they then click on the picture in the application that selects that species and prompts them to enter both the snout-vent length (the length from the tip of the nose to the beginning of the tail) and total length of the salamander, as well as any appropriate comments. The application also accommodates situations in which multiple salamanders are present in the transect, there are no salamanders present, or there were salamanders but they ran away.

CUR Sidebar

Social Media "Crowdfunding" for Undergraduate Research

Kande Jones and Qiang Xu, *Abilene Christian University, qxu@acu.edu*

With federal and state funding for research on a downward trend, scientific-research facilities across the country are experiencing budget cuts and losing grants. In this atmosphere, social media have been filled with novel ideas for garnering funds from sources beyond the conventional funders of scientific research. We first read about "crowdfunding" in a Nature.com article on Twitter. It described the practice as a means of building relationships between scientists and non-scientists to promote public involvement in research.

After researching several of the more prominent sites, we decided that fundageek.com best fit the scope of our research on cattle parasites. Unlike many other sites, the Fundageek website allows project owners to keep any funds raised, even if they don't reach their original financial goal. We also chose their premium marketing offer, in which they take nine percent of the project's overhead expenses in exchange for marketing guidance. After our project profile, "Population Control of Cattle Parasites," was established on Fundageek, we also created a video using Camtasia and Garageband software to describe the impact of our research and our lab techniques in everyday language. Once the profile was created, we posted it on Facebook, YouTube, and Twitter, and encouraged friends and contacts to forward it through their web pages.

We also offered incentives for donors, including email updates concerning our progress and Skype conferences with the research team. We have received a good deal of positive feedback and reached our original financial goal in two weeks. Several people with no ties to the scientific community were interested enough to donate in order to learn more as our research progresses.

Figure 3. Salamander-Selection Screen

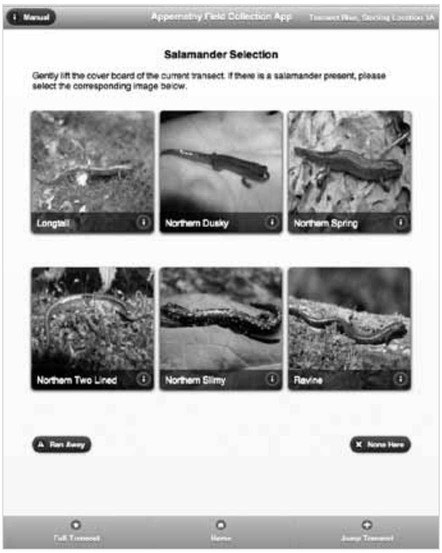
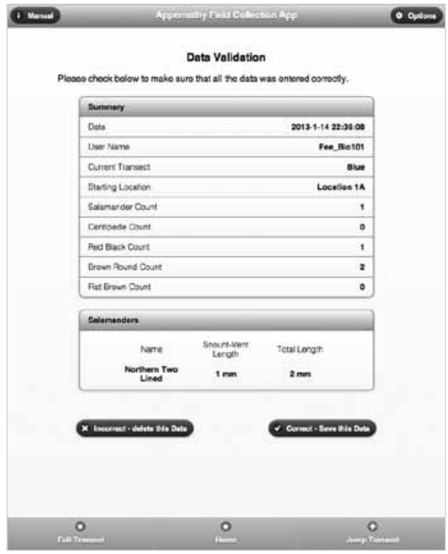


Figure 4. Validate Data Screen



In addition, the application incorporates a user manual, which is available on every page of the application as a modal dialogue—a page that pops up over the current content rather than changing to a new page. This design allows users to refer to the manual at any time without having to worry about losing their current data. Along with the manual, the Options page of the application includes a section, Fieldwork Information, which provides more-detailed instructions on how to collect data in the field.

The next step of the process prompts users to provide information about myriapods (millipedes and centipedes). As on the salamander-selection screen, pictures of different myriapod species are provided, but rather than individually entering each organism, users simply enter the number of each relevant species they see. Once the myriapods section is complete, the next screen asks users to validate their data for the transect (Figure 4). If the data are accurate, they are saved to the database, and feedback is provided in the form of an alert that the data were successfully added. Users are then taken to the salamander-selection page for the next transect. If the data entered are not accurate, they are dropped from the database, and users are taken back to the salamander-selection page for the current transect. This process continues until all of the data points have been observed and data have been entered, promoting consistent and conscientious practice for retrieving data from the field.

Besides the Full Transect feature, Appernathy offers another process for collecting data, called Jump Transect. This process allows data from a single location to be collected, rather than from every data point in the transect. The Jump Transect process exists only in the event of an error in recording the data. Thus, if a field walker needs to return and re-enter data for a particular data point, it can be done without having to run the entire transect.

After collecting information from the field (and when Internet access becomes available), the data saved locally on the device can be uploaded to the main database located on campus and the data on the devices can be deleted. Once the data have been uploaded to the central database, they can be viewed, as well as downloaded, through any desktop web browser

Technical Details

We developed the Appernathy application using HTML5. To develop the structure and design of the application, we also utilized jQuery's mobile framework, jQuery Mobile. In many places, we had to modify the scripts and style sheets, so that the application would represent the college with a custom color scheme, and also so that the application would be able to run the number of custom scripts we use to validate data and write it to the database. Our custom Javascript coding enabled these modifications.

The most unique aspect of this application is its use of web technologies without the need for Internet access. This method of application development became available with the introduction of HTML5 and local storage on mobile devices. Local storage provides the opportunity for us to store information in the browser without the need for a database or an Internet connection. The application makes use of two types of storage, a SQLite database and the browser's own local storage. Compared to a regular SQL database, SQLite is much smaller, which means it is a better choice for a mobile application (which needs to be quite small for the fastest possible download). The most important difference is that this database can be manipulated on the client's side without the need for a server-side scripting language.

In most dynamic web applications, an SQL database located on the server is the backbone of the system. This database allows for information to be entered by the user and for feedback to be provided based on that input. A simple example of a database interaction in a web application is logging into an application. As a user, you typically enter your username, followed by your password. This information is then passed to the server side of the application and is used to query the database, searching first to see if that username exists in the database. If that username is found, then the password you entered is compared to the password that exists in the database in which that username is located. If the passwords match, you are notified that you have successfully logged in. To interact with the database in this way, a scripting language is needed on the server side of the web application.

The database for a web application is typically located on a remote server rather than on the device you are using to access it. To communicate with this database, an Internet connection is required. With our implementation, the SQLite database is actually located on the mobile device. Because of this feature, we were able to eliminate the need for an Internet connection to communicate with the database. We are also able to communicate with the database with our client-side scripting language, Javascript, rather than a server-side scripting language. Once an Internet connection is available, the data that are saved in the database on the device can be transferred to a central database on a remote server.

To support the use of multiple devices in the field, we needed a way to successfully generate unique identifiers for the database, to prevent errors in the central database. A row in a database represents one physical entity. In our application, an entity is a single transect. Each entity in a database needs a unique identifier, known as a primary key, and this key can be any combination of characters or numbers. Each entity needs a unique identifier to ensure that duplicate entries are not saved in the database.

In our application, if any device creates a primary key for an entity that is identical to a primary key of an entity on another device or in the central database, the central database would not allow any of the data on the uploading device to be written to it because that could result in duplicate entities. To solve this dilemma, we used Javascript to generate timestamps. These timestamps are accurate to the millisecond to ensure that they are unique among devices. Once a user begins either Full Transect or Jump Transect data entry, a timestamp is generated and stored in the browser's local storage, where it is readily available until either the Full Transect or Jump Transect is saved to the device's database.

Lessons Learned

The key to our success with this project was the close collaboration between the software developers in the Department of Computing & Information Studies (CIS) and the content developers and users in the Departments of Biology and of Environmental Studies (EVS). Students in CIS served as the software developers, and students in biology served as the content experts and software users. The project began with an independent study by three CIS students who tackled the problem of identifying a suitable development platform for the project. The CIS Department on our campus employs a Problem-based Learning (PBL) pedagogical approach, so this project worked quite nicely as an opportunity to solve an authentic problem (Fee and Holland-Minkley 2010). PBL is a pedagogical approach in which student learning is centered around authentic, real-world problems. For an in-depth discussion, the inaugural issue of *The Interdisciplinary Journal of Problem-Based Learning* contains a particularly useful introduction that lays out the essential elements of PBL (Savery 2006).

Following that independent study, the students began their app-development tasks during the project-management course in the CIS curriculum, which is the capstone experience for CIS majors at W&J. During that semester, they designed the mobile app, programmed, and tested it; then they employed it at Abernathy to determine whether it actually worked for the collection of field data. At this stage in development, the application was used in conjunction with a set of paper instructions for students.

Following the completion of the app, it was introduced into biology courses during a specific lab exercise. Concurrently, other biology and environmental-studies students were working with the software weekly to collect data regarding salamander and myriapod populations at the Abernathy Field Station (cover photo). While this work was happening, we continued to track these data via traditional paper forms, so that in the event of any technical difficulties, there would be no loss of data. This foresight turned out to be useful, as we did experience some difficulties uploading the data upon returning from the field. This was the result of a problematic configuration of the campus network rather than a problem with the software; however, it became an issue that we wanted to resolve in the next iteration of the app.

While this initial work with Appernathy was taking place, we were simultaneously conducting a usability study of the software, using two groups of biology students. The first group worked through the data-collection process as it had been done traditionally with pencil and paper. The second collected their data with the mobile app. The student population for the study included both biology and CIS students. Our results are quite preliminary since we are still analyzing the data; however, a few obvious results quickly emerged during our observations of student usage. First, and not surprisingly, student motivation was quite high. Students preferred using the mobile app to working through the process using paper and pencil. We were also able to observe and measure the advances in students' efficiency while using the app.

Our initial observations point to an increase in student efficiency when using the mobile app, but not necessarily an increase in learning. We have quantitative data that give us some insight regarding data-collection efficiency. The following table presents the average and median time per student observation.

Table 1. Time Spent in Data Collection

Average time with iPad	0:10 minutes	Median with iPad	0:10 minutes
Average time with paper	0:12 minutes	Median with paper	0:12 minutes

Our reading of these data is that students using the iPads are, as expected, faster at executing the protocol than those using paper and pencil. This evidence may play a role in shaping later implementation of the tool in science labs and coursework since use of the app saves time in data collection that can be used for data analysis. However, at this early point in our work, it is enough to simply recognize that our hypothesis was correct: The data-collection process is more efficient through electronic means.

The initial assessment of students' learning indicates near equality of results whether paper or the app is used for data collection. So the primary lesson, as we await the results of more detailed analysis, appears to be that the app increases student motivation and efficiency in field work, but results in only limited differences in student learning. We base this understanding upon the number of correct questions answered from a pre-test/post-test protocol.

As a result of the ongoing analysis, we have made some revisions in the mobile app. Specifically, since we now know that the app made the process more efficient, we made some design changes to incorporate the usage instructions directly into the app, further improving the efficiency of data collection. We also made a number of user-interface modifications, based upon student feedback and our own observations of students' use of the app. We suspect these may result in improved learning, but we will need to conduct further usability studies to test that hypothesis.

Conclusion

Once we complete our analysis of the data gathered in our usability study, we will run another set of students through the process to see whether the changes we've already made in the app make any appreciable difference in student learning. Following this formative evaluation, we will place the app in its final form and will refrain from any future modifications. Meanwhile, we continue to use the app at the Abernathy Field Station to monitor and collect data on salamander populations. The app is also being further integrated into lab experiences and classes within the biology and environmental-studies curricula.

It should be noted that while the development of Appernathy has been challenging, it is much easier now to develop a customized mobile app than it was two years ago when we began this project. In part, this is because the tools for such development projects have matured, and because more groups and individuals have developed projects using the tools and written of their experiences, allowing others to benefit. For our project, we are in the process of creating an open-source repository of the codebase we used, on GitHub, so that others might consider developing their own customized apps for collecting field data. While our approach certainly requires some technical ability, anyone with a proclivity for software development should be able to replicate the process.

Acknowledgment

The Motorola Mobility Foundation funded a portion of this software-development work.

References

Fee, Samuel B, and Amanda M. Holland-Minkley. 2010. "Teaching Computer Science through Problems, not Solutions." *Computer Science Education* 20(2) 129-144.

Savery, John R. 2006. "Overview of Problem-based Learning: Definitions and Distinctions." *The Interdisciplinary Journal of Problem-Based Learning* 1(1) 9–20.

Samuel B. Fee

Washington & Jefferson College, sam@washjeff.edu

Samuel B. Fee is a professor in the Computing and Information Studies Department at Washington & Jefferson College. His courses focus on digital-media production, web design, and mobile app development. He has co-edited a volume on computing and education, The Role of Criticism in Understanding Problem Solving. *For more information, see http://samfee.net/.*

Christian Griffith is a student majoring in computing and information studies at Washington & Jefferson College (W&J), where he has focused mainly on digital-media production and graphic design. He has completed several projects for the campus biology and languages departments. His work is featured on his portfolio website at http://www.chrisgmedia.net.