International Journal *of* Designs for Learning
2024 | Volume 15, Issue 1 | Pages 1-11

DESIGNING A TOOL TO SUPPORT ONLINE PHYSICAL EDUCATION

James Castle, *University of Georgia*

This design case details a seven-year iterative design process to create an app for use in online physical education classes. Each iteration addresses the shortcomings of the previous version. The most recent iteration of the app allows students to use Fitbit™ devices to record heart rate data, which each student sees as progress toward course goals on the homepage of the course. The current version of the app has evolved to provide a seamless student experience using a web application programming interface (API) and data standards such as learning tools interoperability (LTI). The student experience of using the app is thoroughly documented, as are design processes and principles for engaging in similar design processes.

James Castle is Associate Director of Online Learning at the University of Georgia. He has worked in the field of education for 17 years in roles ranging from classroom teacher to instructional technologist & designer. As the associate director at the Office of Online Learning, he works to help develop online graduate degree programs, provide strategic direction via data analytics, and maintain high-quality course and program design practices within the instructional design team. He earned his Doctorate in Learning, Design, and Technology from the University of Georgia.

https://doi.org/10.14434/ijdl.v15i1.34531

INTRODUCTION

The very concept of online physical education seems paradoxical at first glance. However, advances in both consumer and web technology over the past decade have made it possible for students to participate in meaningful physical education courses while at a distance. Wearable technology, such as Fitbit™ or Apple Watch, makes it possible for people to collect physical activity data with the press of a button. These devices are typically wrist-worn and gather a range of biometric data like heart rate, steps, or sleep quality. Among the data these devices can collect, heart rate data during exercise are the most valuable for applied physical education. The activity data can then be shared in the context of a course to provide the basis for an authentic physical education experience to complement the conceptual components of the course. Cultivating this applied physical education experience has been challenging in a formal education context. An online physical education web application, called simply hereafter the app, was created for a fully online physical education course to begin to address this challenge. This design case presents the design iterations of this app developed over a seven-year period. The primary takeaways from this case for the design team include the importance of the relationship between instructional designers and faculty in higher education, the necessity to dedicate resources and attention to projects as technology evolves, and the importance of development capacity for realizing innovative instructional goals.

DESIGNING A TOOL TO SUPPORT ONLINE PHYSICAL EDUCATION.

Instructional Need

The design case presented here began as an online course development project between an instructional designer (the author) from the institution's Office of Online Learning and a faculty member from the kinesiology department in the institution's College of Education in 2013. At the institution where this project took place, all undergraduate students are required to complete at least one physical education course to graduate. The faculty member had a vision for an online physical education class (i.e., *Online Walking)* that students

ITERATION NAME (YEAR)	DESCRIPTION
Students Tracking Down Data (2013)	Students submit specific CSV from the GarminTM website for processing
Seamless Data Delivery (2017)	Data is accessed via FitbitTM API. Data reports are delivered to instructors via email. No student access to data within app.
Surfacing General Data (2018)	Students and instructors access basic performance data from Fitbit via an external LTI app.
Ubiquitous Access and Detailed Data (2019)	Students and instructors access detailed performance data using a widget on the course homepage.

TABLE 1. Physical Education App Iterations.

could complete while participating in internships away from campus or study abroad programs, thus enabling them to complete a graduation requirement while away from campus. A fundamental part of this vision was that students would track their heart rate while completing exercise to make progress toward the course goals of accumulating time in elevated heart rate zones. The core design challenge for this aspect of the course was to transform data collected by a heart rate monitoring device into data that were meaningful in the context of the course. This challenge is more complex than it appears on its face and is the central narrative of this design case.

DESIGN CONTEXT

The physical education app described in this design case has been in use in online physical education classes since the summer of 2013, and those classes have been the main venue for design feedback for the app. I was given the opportunity to talk with students, particularly in the early offerings of the course, to identify key areas where students struggled with using the app. These conversations and experiences were critical in identifying the weaknesses that were addressed in each successive design iteration. The four design iterations detailed in this design case are outlined in Table 1.

This project is technically complex. Its infrastructure includes Python, JavaScript, Google Apps Script, and AWS. However, our office has never had a dedicated app development or programming team. For this project to get off the ground, I had to serve as both the instructional designer and app developer. This required me to stretch and cultivate technical skills that I did not have when the project was conceived. At times we were able to bring in student developers to solve challenges and help me "skill up" to continue development and support of the project. However, this project has been primarily driven by a lead faculty member and a single instructional designer for the duration of its existence.

It is also worth noting that the opportunity to collaborate with a faculty member on a design project such as this one over the course of seven years is somewhat unusual. Instructional designers from our institution's Office of Online Learning typically move from one development project to the next, helping to launch several online courses each year. However, in the case of the online physical education course (to which I was initially assigned by sheer luck), I have been given the time and resources to complete several revisions of the project. We have also launched an additional online physical education class (*Online Jogging*) and scaled up our physical fitness app in response to remote learning necessitated by the COVID-19 pandemic. Over the years that this project has evolved, I have been promoted three times (first to *Lead Instructional Designer*, then to *Assistant Director for Instructional Design*, and most recently to *Associate Director*), but I am still afforded the opportunity to maintain my relationship with the people using the physical education app. This is, by far, the longest-running project our office has undertaken, and we could not have completed the design iterations described here without the time and resources allocated to it.

ITERATIVE DESIGN

Iteration One: Students Tracking Down Data

Development of the first iteration of the physical education app began in the spring of 2013. The major challenge we faced was how to take data from a heart rate monitoring device and transform it into data that were meaningful in the context of the course's goals and timeline. During this iteration of the app, students were required to wear Garmin™ chest strap heart rate monitors. After completing an exercise, the students would synchronize data from their chest strap with their Garmin™ account via the Garmin™ Connect App. This presented a challenge, as there was not a seamless way for students to share their activity data from Garmin™ Connect with their instructor to demonstrate the completion of course goals.

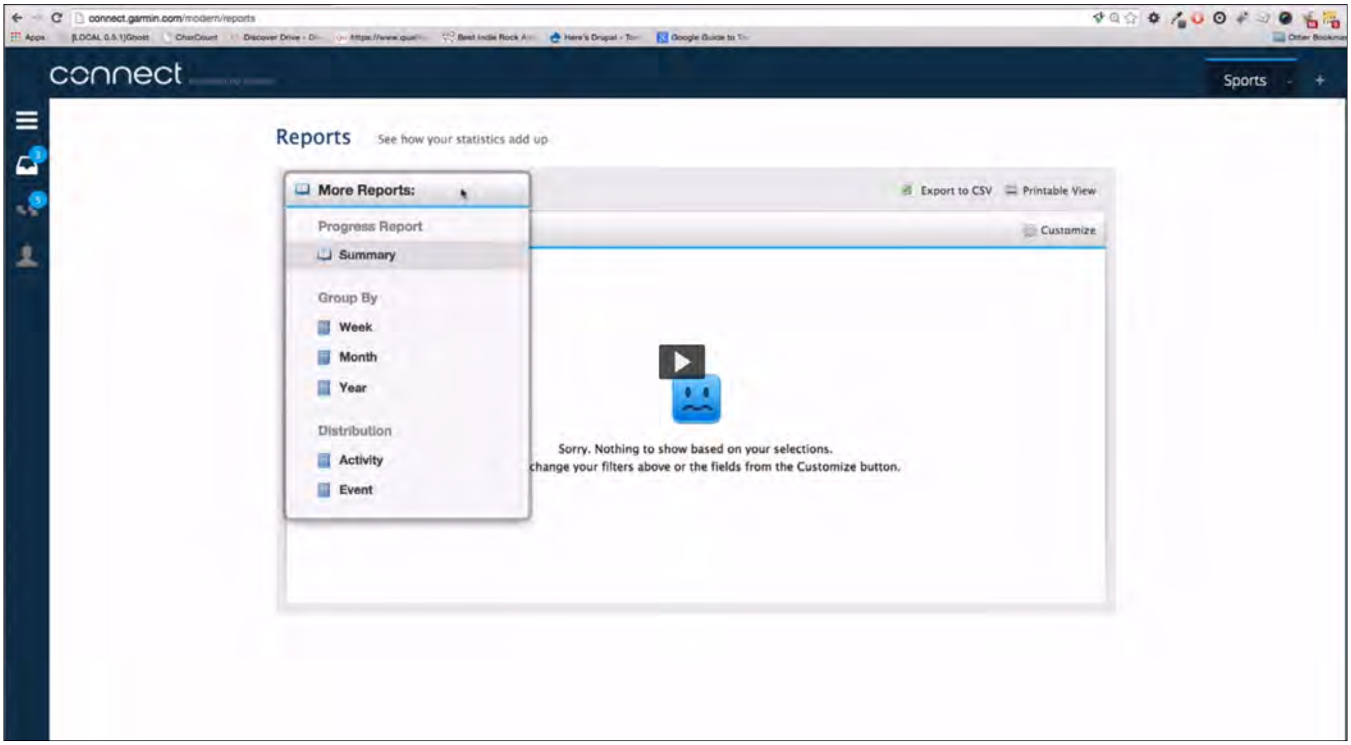

FIGURE 1. A screenshot of the Garmin™ Connect web interface at the point when the student generates a CSV formatted report.

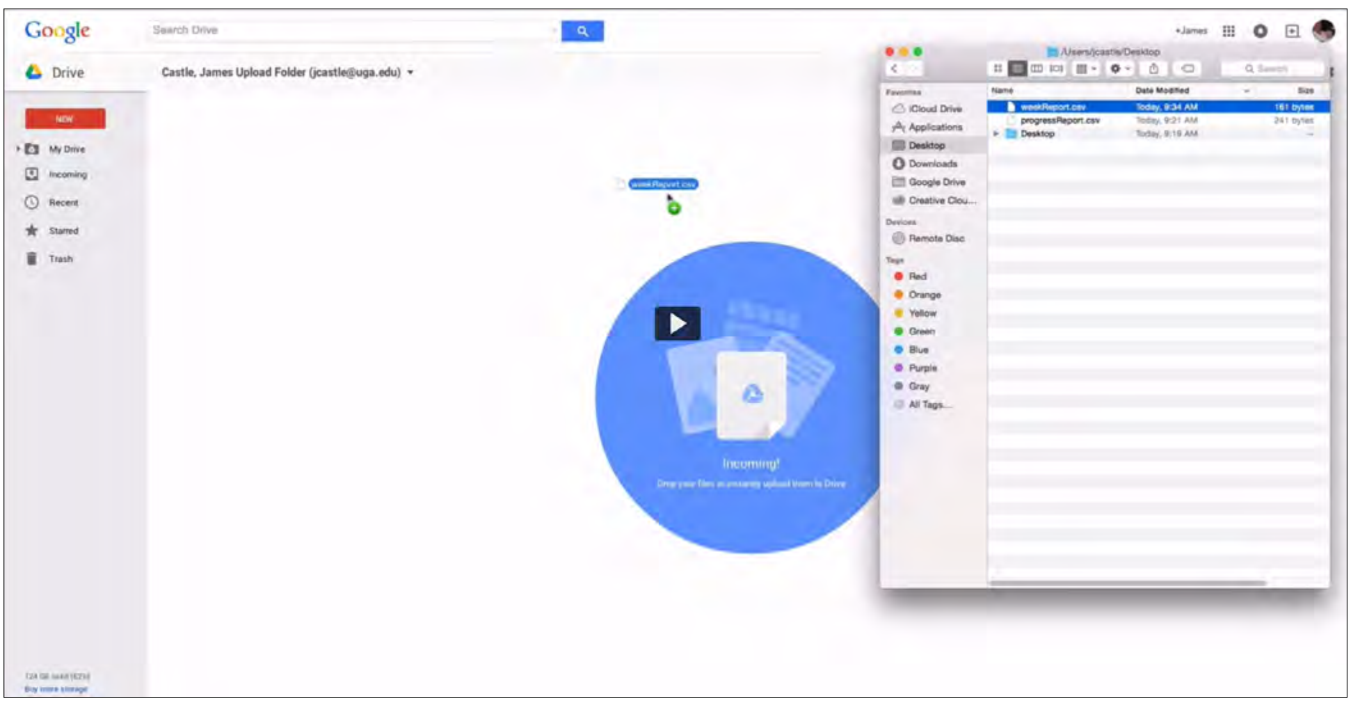

FIGURE 2. The interface for uploading a CSV into Google Drive for processing.

To address this challenge, the first version of the physical education app required students to download a specific report, formatted as a CSV, from Garmin™ Connect and then upload it to a Google Drive folder that was shared with each student individually (See Figures 1 and 2). We then used a Google Apps Script to monitor each student's upload folder for new files, and once a new file was detected it could be processed to deliver the student's activity results to the instructor of the course. This system allowed us to take general activity data from Garmin™ Connect and transform it into contextualized data for the online physical education class. However, this system introduced several usability challenges. After reviewing a tutorial video, we created in 2013 for the course that used this version of the tool, I noted that students had to follow a very specific set of steps to successfully submit their activity data. Students had to:

1. Configure their Garmin™ profile to set their week to start on Monday.
2. Navigate to the reporting interface in Garmin™ Connect.
3. Navigate to the Progress Summary area of the Garmin™ Reporting site.
4. Set the Progress Summary to Group By Week.
5. Set the Report Dates to correspond with the course Module Dates.
6. Export the report to a Comma Separated Values (CSV) format.
7. Navigate to their individual Google Drive Upload folder.
8. Upload the CSV that was exported from Garmin™ Connect to their Google Drive folder.

Every student was required to complete this process every week for their activity data to be submitted for the course. If a student made a mistake on any of these steps, their data would not be submitted properly. In practice, students would often not configure their progress summary report correctly or upload the incorrect file type for the system to process (e.g., uploading a PDF instead of a CSV). When this happened, students would inevitably need to reach out to the instructor to figure out which step they had missed. In many of these cases, I was called on to either check the Google Drive folders to see what had gone wrong or troubleshoot the system with students directly. This gave me great insight into how students used the system and what pain points needed to be addressed through further development.

While this iteration of the app provided a conceptual basis to build upon (i.e., connecting activity data with physical education course goals), the design of the system: (a) left too much room for error in submission processes; and (b) led to more effort and focus on data submission than on the learning outcomes of the course. After using this system for several semesters, we decided to undertake a major overhaul of the app to simplify the data submission process for students.

Iteration Two: Seamless Data Delivery

Development of the second iteration of the online physical education app began in the fall of 2016. We started by re-examining wearable devices that were available for use by students in the online physical education classes. In 2013 we had chosen the Garmin™ chest straps because wrist-based devices were not yet accurate enough at monitoring heart rate. However, by 2016 products developed by Fitbit™ had improved their heart rate monitoring technology to the point that they were accurate enough for our needs. This was a key decision in the design process, as Fitbit™ also provided a public application programming interface (API) that could be used to allow students to seamlessly share their activity data for use in the course. The API provides a mechanism to retrieve data from the Fitbit™ servers programmatically, which allowed us to build automatic heart rate data retrieval into our app. This allowed the data to be used in our application, which was developed specifically for the online physical education courses. This gave us the opportunity to replace the tedious process from the first iteration with a single link that students could click one time at the beginning of the course to share their data. By clicking the link, the student's data would become available for retrieval via the Fitbit™ API as needed for course assessment. While we have continued to improve this app since making the change to using an API to retrieve student data, this was the most impactful design decision we have made over the course of the development of this app. Changing from a manual data reporting process to an automated API-driven process facilitated every design improvement we have made since.

The implementation of the Fitbit™ API into our app was the first opportunity our student developer had to participate in the project. I collaborated with the student to scope the data we would need from Fitbit™ and identify the API endpoints from which we would request data. Once we understood our data needs, we requested API keys from Fitbit™ via their developer portal. API keys are values that identify our app to Fitbit™ when we make requests for data. Once our general API access was in place we requested enhanced access to heart rate data, as it was vital for demonstrating student performance toward course outcomes. This involved submitting a detailed request that had to be approved by FitbitTM. Once we explained the educational nature of the app we were creating, Fitbit™ granted the request. With access to the API in place, we had to set up a workflow for allowing students to grant access to their data. We ended up doing this with a link from our learning management system (Brightspace) in the introduction to the course.

Our next big design decision was where we would build the application. Up to this point, the app had been processed using Google Drive and Google Apps Script, but those were not great tools for redesigning the app to work with the Fitbit™ API. After a bit of consultation with engineers from Amazon Web Services (AWS), we settled on AWS Lambda1 as the primary technology that would power this iteration of the app. AWS Lambda provides access to computing power on demand without the need to manage server infrastructure. This allowed us to build the logic and data flows of our app without worrying about the more technical details of server deployment.

Once completed, this version of the app enabled course instructors to request a summary of their students' course activity at any time via a simple web form. Each time an instructor made a request, our Lambda script (which was written in Python) would query the Fitbit™ API for the necessary student data, put the data into a format meaningful for the context of the course, and deliver the CSV to a predetermined list of email addresses that included the course instructor and the designer of the app.

While this iteration of the app improved upon the previous version by greatly simplifying access to student activity data, a new set of challenges emerged. First, as the course rosters grew longer, the amount of time needed to gather all the students' data grew as well. Once enrollment in the online walking classes reached close to 60 students the Lambda script had to gather data for all those students each time the CSV was requested. Gathering activity data on that many students can take 3-4 minutes, which meant that after clicking the "request" button on the web form the requestor would need to wait until all the data were retrieved before closing their browser window.

For students, this version of the app presented a simplified activity loop. They simply had to walk and sync their data. However, in conversation with the course instructor, I learned that a challenge for students was that they could not see any of their data in the context of the course. The CSV reports only went to instructors, and students only knew if they had missed their goal once their grade had been entered into the course gradebook. They could see their activity data via the Fitbit™ app or website, but those data were decontextualized — it did not factor in how heart rate activity contributed to the overall grade or the date cutoff for the course modules. This resulted sometimes in students failing to meet a course goal that they thought they had met.

While the second version of the app was clearly an improvement over the first version, we knew soon after launching the second version that we needed to address these two major challenges. In the fall of 2017, we continued working on the app with the goal of (a) making it more performant and (b) allowing students better access to their data so that we could more quickly close the feedback loop (Norman, 2013).

Iteration Three: Surfacing General Data

The third iteration of the app was first used in classes during the summer of 2018. The most noticeable revision to the app in the third iteration was adding a user interface that was accessible by both instructors and students. We designed the app to use the learning tools interoperability (LTI) standard2 to securely pass data between the Learning Management System (LMS) and the app. Adding LTI to the app allowed us to detect the identity of the person accessing the app. It also allowed us to detect the specific course section in which the student was enrolled. These enhancements were key for providing more individualized, contextual access to data within the app.

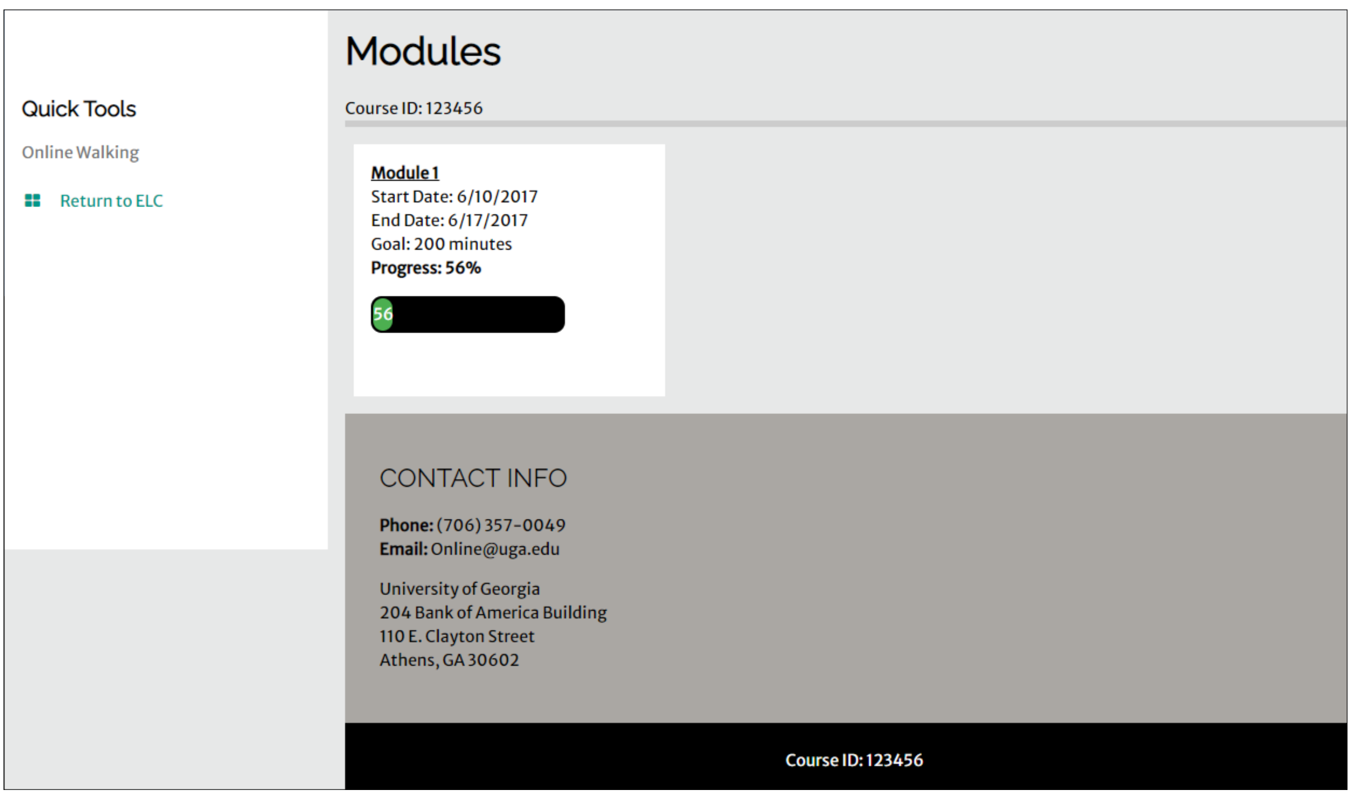

FIGURE 3. Screenshot of the student view of progress in iteration three of the app.

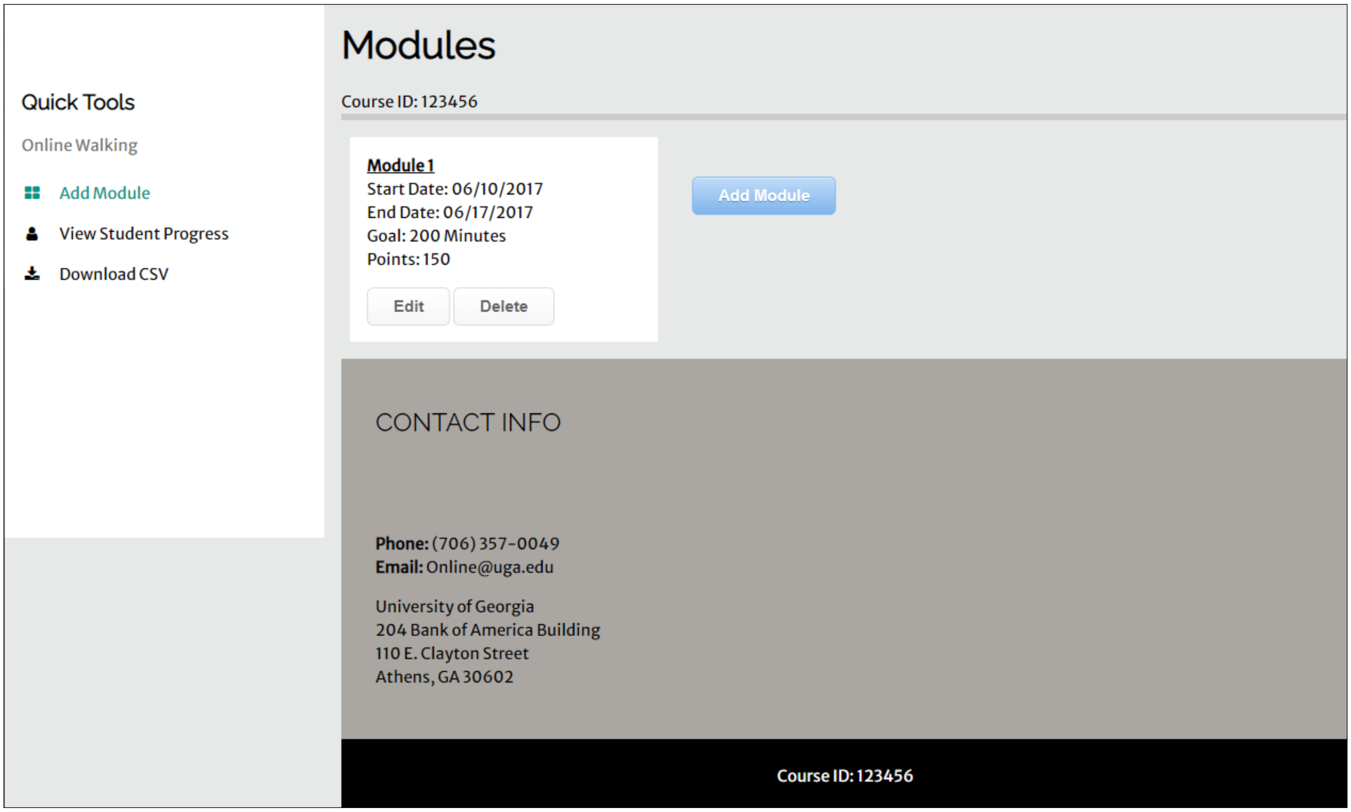

FIGURE 4. Screenshot of the instructor view for managing modules.

Another benefit of LTI was that we could detect the course role of the person using the app. Therefore, when someone clicked the link to launch the app, we could differentiate between students and instructors. This allowed us to present different data to people in each role (See Figures 3 and 4). When students visited the app, they would see an overview of each module in the course as well as their progress toward the goal for each module. The ability to access their data allowed more transparency for students to monitor their own progress in the context of the course. Each student was now able to view their activity data as it pertained to their progress toward class goals, an ability that was not available in either of the two previous versions of the app. This was another major milestone in the development of the app, as it helped to bring the physical fitness activity portion of the class into focus for students. Previously, students had to rely on the instructor to translate their activity data into a grade. Now, students can now keep track of their own progress in real-time.

The design of this third version of the app dealt with the challenge of performance. As described earlier, the second version of the app became inefficient once full enrollment in the online physical education classes was reached. This was because the app had to fetch and process data from the Fitbit™ API every time a request was made to the app. While this resulted in the app always using up-to-date data, it also made the app inefficient, often taking several minutes to load. To address this challenge, we developed a system where the app would fetch data for each course at set intervals and store the data in CSV files that were readily accessible to our app. These CSV files could be loaded in a fraction of a second resulting in improving the app's performance tremendously. The drawback to this approach was that the data shown in the app was not always up to date. We experimented with different intervals for refreshing data, from once a day to once every four hours, but this time lag in data availability presented a problem with the third version of the app. We quickly identified this as a challenge we would address if we were able to undertake another design iteration.

There were other challenges we wanted to address as well. First, the user interface for our application ran on a complicated technical infrastructure that came with a monthly cost of around $35 to run the entire system. While this is not a huge sum of money, we also did not have a revenue stream tied to the project. Initially, the user interface was set up with performance as the priority, but since we had solved the performance challenge, we wanted to re-evaluate the user interface to see if we could simplify it from a technical standpoint and eliminate the monthly cost to run it. Another challenge with the user interface was that the student had to click a link to access it. We wanted to try to get the user interface of the app embedded on the homepage of the course so that it was a ubiquitous part of the course. Finally, we needed the user interface to provide more contextual data than the original design allowed for. For example, we wanted to display the date that the data in the app was last updated so that students would know if there was a time lag on their activity data loading. We also wanted to provide students with more granular activity data so they would know how their activities contributed to their progress on the course. Finally, we wanted to address challenges with students in other time zones missing deadlines because all the course deadlines were shown in US Eastern time.

Iteration Four: Ubiquitous Access and Detailed Data

For the fourth iteration of the app, we completely redesigned the user interface. Our motivation for the redesign was to move away from the technologies we were using that were: (a) difficult to maintain; and (b) had a monthly cost. We also wanted to enhance the visibility of the app by putting it front and center in the class instead of sitting behind a link where it could be perceived as less important (Norman, 2013). While we had abandoned Google Drive as part of the app's infrastructure after iteration one, we reconsidered it for use in the fourth version, albeit in a very different way.

A lesser-known feature of Google Drive is that it can be used to host web apps via Google Apps Script. Setting up web apps using Google Apps Script is relatively easy, and there is no cost to build or host apps on Google Drive. These factors made Google Drive web app hosting an attractive option to host the user interface of the fourth iteration of the physical education app. Unlike our previous use of Google Drive, this iteration of the app did not require students to upload anything to Google Drive. Because of the way the web app was set up using Google Apps Script, students did not need a Google Drive account to use the app, nor did they even know that the finished product was running in Google Drive.

The new user interface running in Google Drive still used LTI so that the LMS could pass identity, role, and course offering details to the physical fitness app. However, with the new version of the app, we were able to embed the app in a widget on the homepage of each course in the LMS rather than requiring a link to launch the app in a new browser window. Situated in the homepage widget, the app is an ever-present part of the course. Students see it the first time they log into the course, and anytime they want to check their progress they can just go to their course homepage. For an overview of the data flow for this version of the app, see Figure 5.

The new widget provided a dropdown menu with the name of each module in the course. When a student chooses a module from the dropdown, the widget shows data for that module. In addition to showing the student their progress and requirements for the selected module, the widget showed the date that data was last updated, which helped set student expectations for when specific activities would show up. Additionally, the widget showed a summary of

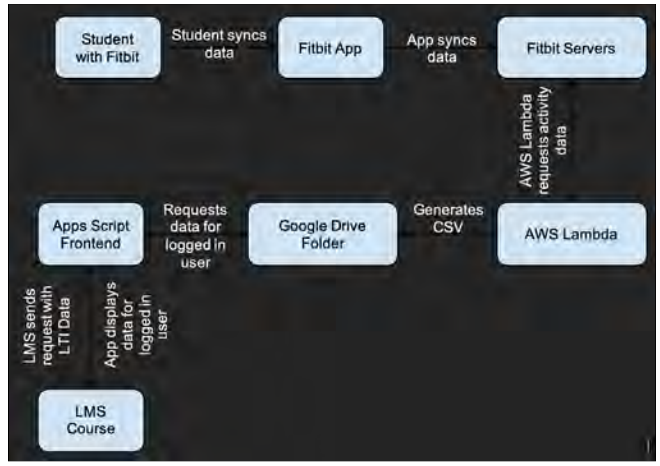

FIGURE 5. Illustration of the data flow in the current version of the physical education app.

all activities completed in the selected module, along with the time spent in each heart rate range during each activity. This activity breakdown made it clearer for students how the time they spent on their activities was contributing to their course progress as compared with earlier versions of the app. The widget would also show students activities that did not count toward their module progress and gave them the reason the activity did not count (e.g., the activity was auto-detected or the activity had no associated heart rate data). Last, we added a time zone interpreter that could detect the student's time zone and translate the module due date to their local time zone. These changes decreased student questions regarding their activity progress. This version of the app (as shown in Figure 7), which is currently in use in online physical education classes, provides students with all the information they need to succeed in meeting their activity goals in their physical education course.

Minor Iteration: Scaling the App for COVID-19

In March 2020 the COVID-19 pandemic forced all courses at our university to operate at a distance. For traditional physical education courses, this presented a challenge. However, the Fitbit™ app that was developed specifically for use in online physical education challenges was able to help fill the void created by social distancing with only a few minor "under-the-hood" changes. This is a minor iteration because from the student and instructor viewpoints, nothing changed. The app still functions for end users as it did before the pandemic. However, scaling the app from 60-100 students using it concurrently to potentially several thousand students using it concurrently did require some design changes.

Having the app serve a potentially unlimited number of students required changes with how the app pulls data from the Fitbit™ API. When we changed the app to fetch data at intervals and store the data in CSVs, we still had the app fetch data for every student in the system every time it ran. With around 60-100 students in the system, this is not a problem. It might take the data retrieval script five minutes to run, but it is running in the background (i.e., no one is waiting for it to finish for the app to load). However, as the number of students in the system increases, we run the risk of the data retrieval script timing out before it can finish. This poses a serious risk to the data integrity of the system, if the data retrieval script constantly times out, then the data in the system never updates, rendering the app useless.

To mitigate the effects of the system being flooded with new students, we rewrote the data retrieval script to pull data course-by-course rather than for all students in the system. This allowed us to specify a certain number of courses to pull data for each time the retrieval script ran, ensuring that we would not try to pull data for too many students at a time. After a bit of testing, we settled on pulling data for six classes every fifteen minutes on a rolling basis. So, every fifteen minutes our data retrieval script gathers data for six courses in the system, and once it reaches the end of the course list it simply starts back at the top of the list. This ensures that every course in the system is updated multiple times per day, but it also keeps the load of each data retrieval low enough that we are confident the script will finish.

I added this section to not only highlight how we adapted the app to function at scale for the COVID-19 pandemic but also to draw attention to the value of this sort of design project when such a situation arises. We did not set out to design the online physical education app to prepare for a pandemic. However, because we had put in years of work to execute a vision of quality online physical education, we had an infrastructure in place that could be adapted to serve the entire university community. This is a benefit of innovative learning design that should be highlighted and celebrated.

REVIEW OF CURRENT STUDENT EXPERIENCE

With the current version of the app, students must grant the app access to their Fitbit™ data. They do this by clicking a link that directs them to log in to their Fitbit™ account. They are then given a description of the data the app will access, and they must affirm the app's access to the data by clicking a confirmation button. This authorization flow is illustrated in Figure 6.

Students then complete exercise activities while wearing the Fitbit™ device. They sync the device like any other Fitbit™ user would (typically, but not exclusively, via a smartphone), and the data they synchronize then becomes available to the online fitness app via the Fitbit™ API. Once students begin completing exercise activities, they see their progress reflected in the online fitness app, which is embedded on the homepage of their course. When students visit their course in the LMS, the module loads the data

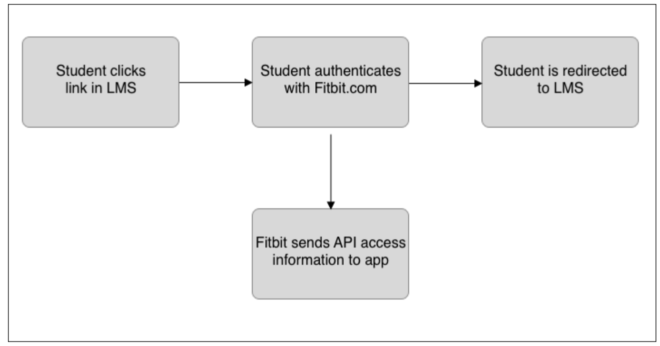

FIGURE 6. Student authorization flow for physical fitness app.

for the currently active module and displays the student's performance data. Specifically, the online fitness app gives students the following information regarding their progress toward course requirements:

- Overall progress toward the goal for the chosen module;
- Minutes required and points available in the currently selected module;
- Date that the data were last updated from the Fitbit™ servers;
- Start and end date of the module;
- Local time zone we believe the student is in, along with the due date of the selected module in that time zone;
- List of activities counted for the selected module, along with a breakdown of time spent in each of the four heart rate zones;
- List of activities that did not count for the selected module, along with the reason the activity did not count.

Students can use data presented by the app to monitor their progress in the course and improve their performance over the duration of the class. Figure 7 provides a screen snapshot from the LMS of a student's activity on the first module of the course. The widget shows that the student's first attempted activity was logged on January 8, 2020, but it did not count because it was auto-detected rather than intentionally recorded. The student then completed 12 more activities over the course of the module, each of which is shown along with the time in four different heart rate zones for each activity. The heart rate zones are listed as *OOR* (out of range), *FB* (fat burn), *C* (cardio), and *P* (peak). At the end of the course, students are instructed to revoke the app's access to their Fitbit™ data. Once access is removed, the app can no longer retrieve data from the Fitbit™ servers for the student.

REVIEW OF CURRENT INSTRUCTOR EXPERIENCE

The physical education app also loads on the homepage of the course for instructors. However, instructors are provided with additional data, such as allowing instructors to select any module in the course from a dropdown menu to see

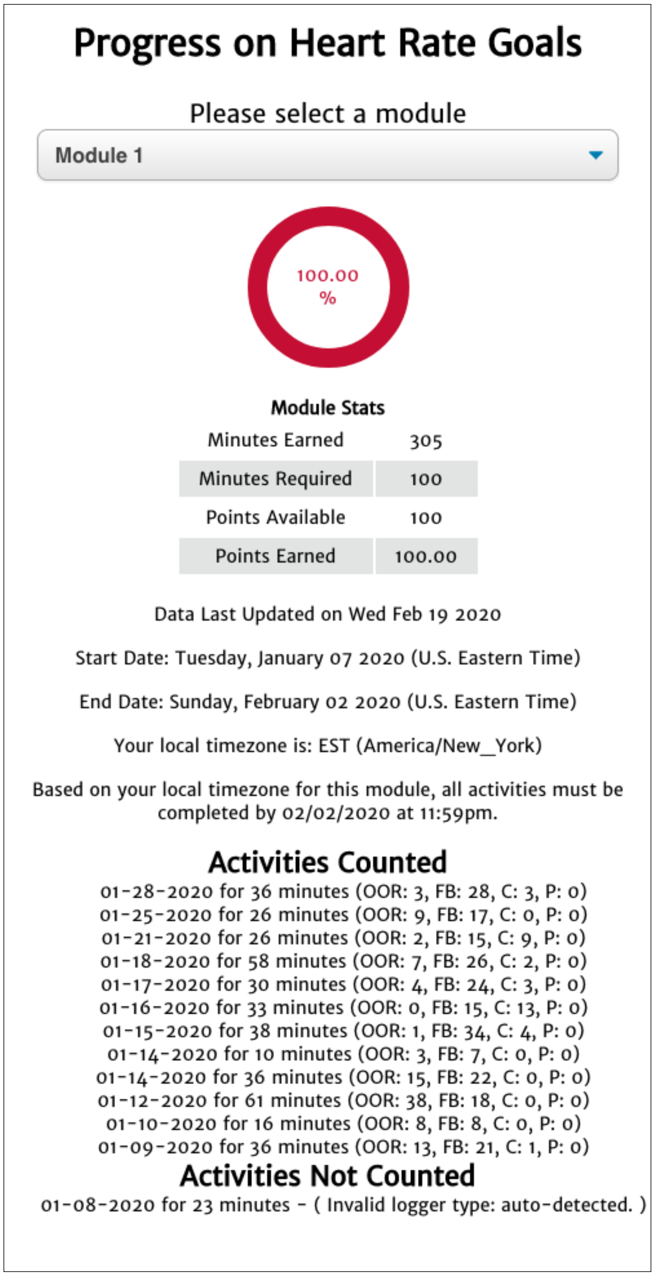

FIGURE 7. Screenshot of the student view of the current iteration of the app.

an overview of class data for that module. Upon selecting a module, they see a list of students who have accrued minutes for that module, along with their total minutes and current score for that module, as shown in Figure 8. Additionally, they are given links to either (a) make changes to the modules for the class (i.e., editing dates or goals) or (b) download a CSV file with all the detailed activity data for their course. The CSV download can be useful in cases where an instructor needs to closely examine each activity for a

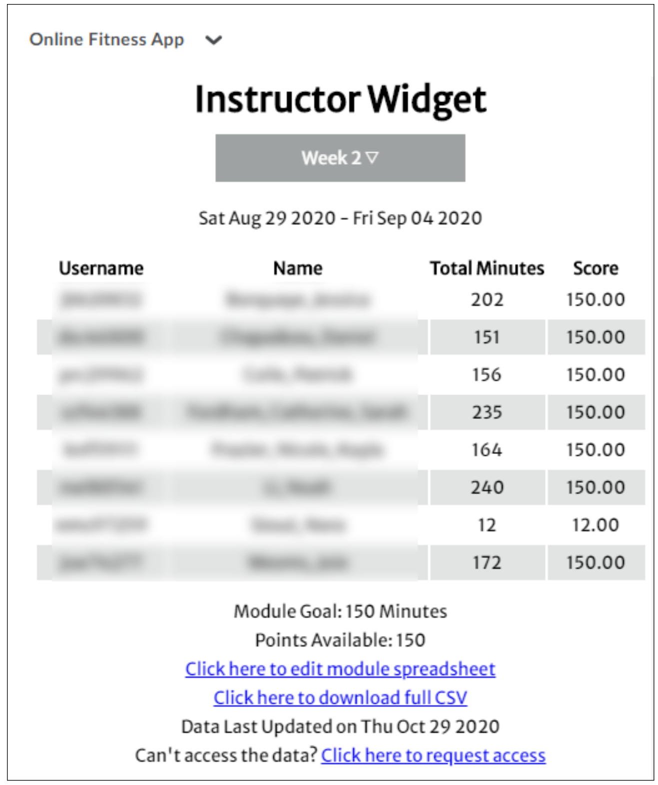

FIGURE 8. Screenshot of the current instructor view of the app showing all students' progress along with contextual course data and access to full activity data.

student in the course to provide clear feedback on activity for a module.

DESIGN REFLECTIONS

The design of the physical education app started with a novel idea (heart rate based online physical education) that has remained consistent in its vision and orientation. The design iterations were informed first and foremost by reflections on student and instructor usage of the app. Each of the iterations outlined in this design case takes steps to remedy a shortcoming of the previous version. However, another important factor in the evolution of this design was the availability of technology to meet the instructional vision. In 2013, there was no reliable heart rate monitoring device that had a well-documented, publicly available API. The closing of the gap between the course vision and technical possibilities enabled our design to close the gap between the user experiences in the early iterations and the current user experience. Throughout the design and development of the four iterations, we made many design decisions. Table 2 summarizes some of the major decisions made during each phase.

ITERATION NAME (YEAR)	MAJOR DESIGN DECISIONS
Students Tracking Down Data (2013)	• Value the systematic processing of data into the context of the course (e.g., meeting heart rate goals per module). • Use of Google Apps Script as main infrastructure for the project.
Seamless Data Delivery (2017)	• Switch from Garmin to Fitbit™ products, primarily for ease of API access. • Simplify submission process for students so they can focus on physical activity. • Integrate AWS and Python into the project to provide more robust and flexible solutions.
Surfacing General Data (2018)	• Provide on-demand access to data for both students and instructors using the tools adopted in iteration two.
Ubiquitous Access and Detailed Data (2019)	• Place the widget front and center on the course homepage. • Re-integrate Google Apps Script and JavaScript to simplify the app's frontend.

TABLE 2. Major Design Decisions.

While design cases do not exist to test a theory, a reflection on the iterations of this design process brings to mind the critical variables identified by transactional distance theory (Moore, 2019). Although the first iteration of the app did take substantial steps toward contextualizing heart rate data for the online physical education courses, it did not provide a good contribution to the design in terms of the course structure (sharing information with students), dialogue (encouraging constructive interaction), or learner autonomy (empowering students to make decisions about their learning). Instead, the submission process for the first iteration of the app was complex, causing much of the course dialogue to center on the technical details of submitting heart rate data rather than on the impact of physical exertion and heart rate on one's health.

Each successive iteration of the design improved the app in these aspects while responding to the shortcomings of the user experience. For example, providing students with an interface to view their data in the context of the course helped to improve the course structure so that students more readily knew what was expected of them. Providing more transparency and clarity to students by showing them their data helped to transform the interactions in the course from technical exchanges (e.g., "Why didn't my minutes count?") to constructive dialogue. Finally, by showing students their own course progress in terms of their exercise activities, the app empowers students to make decisions about how they will approach meeting their course goals, fostering learner autonomy. Throughout its evolution, the app has become a more integrated part of the course, moving from being a process completely outside of the course's LMS space to being embedded in the course homepage so that students see it every time they visit the course. While this design case does not test the app based on transactional distance theory, it does illustrate how a naturalistic design process can unfold along established theoretical lines.

With iteration four, the app is stable and in use year-round for our *Online Walking* and *Online Jogging* courses. We intend to support the app for as long as there is a need for online physical education at our institution. The app will need maintenance periodically as technology evolves. Inevitably, certain parts of the app will become deprecated and need to be upgraded. For example, we will need to upgrade the technology used to connect the app to our LMS. The connection was initially built with LTI 1.1, which has been deprecated in favor of LTI 1.3. Soon, we will need to decide whether we should upgrade to the more complex and technically demanding LTI 1.3 or change the integration to work with Brightspace's API.

When I started working on this project, I did not know that I would still be involved with it many years later. It is vitally important to the success of this project that the leadership in my unit has remained supportive of continuing to dedicate time and resources to support this project. I am also aware that without my technical skills, this project would not have been able to flourish. It is somewhat unusual to find instructional designers who can code. When I started this project, I did not know Python at all and had only a passing knowledge of JavaScript. I sought out opportunities to learn about development on sites like Codeacademy™ and CourseraTM. I attended conferences intended mainly for web and user experience professionals, such as *An Event Apart*. Most importantly, I had the opportunity to work with gifted student developers, which supercharged my own capacity because I was able to learn how they approached development problems. Through these experiences, I came to understand that if I had no development skills then I would never be able to see my designs through to completion.

SUMMARY

With wearable technology becoming more commonplace, possibilities for technology-enhanced online physical education should increase. However, to facilitate meaningful learning, it is important to consider how we design those devices (and the data they generate) into the learning experience. This design case presents the evolution of one such design, along with the most important decisions and iterations that have taken place over seven years of design and development work. The transformation of the app from the first iteration to the fourth iteration has informed other instructional design projects of mine over the past several years. Of all the ways this project has helped me improve as a designer, I think the most profound is that it has made me consider how best to expose students to data that reflects their own performance to encourage learner success.

1There is much more technical infrastructure underlying this part of the app's development, an explanation of which would be out of place in this conversation of design decisions. It is mentioned here only to acknowledge that as our design has evolved, our technical tools and skills have had to evolve along with it.

2This is a standard for sharing data between systems. For more information see: http://www.imsglobal.org/activity/learning-tools-interoperability

REFERENCES

Moore, M.G. (2019). The theory of transactional distance. In M.G. Moore & W.C. Diehl (Eds.), *Handbook of distance education* (4th ed., pp. 32-46). Routledge. https://doi.org/10.4324/9781315296135-4

Norman, D. (2013). *The design of everyday things*. Basic Books