Journal of Information Systems Education

Volume 32
Issue 1
Winter 2021

Teaching Tip
A Teaching Module of No-Code Business App Development

Shouhong Wang and Hai Wang

Recommended Citation: Wang, S. & Wang, H. (2021). Teaching Tip: A Teaching Module of No-Code Business App Development. *Journal of Information Systems Education,* 32(1), 1-8.

Article Link: http://jise.org/Volume32/n1/JISEv32n1p1.html

Initial Submission:	11 April 2020
Accepted:	28 July 2020
Abstract Posted Online:	10 December 2020
Published:	13 March 2021

ISSN: 2574-3872 (Online) 1055-3096 (Print)

Teaching Tip
A Teaching Module of No-Code Business App Development

Shouhong Wang
Charlton College of Business
University of Massachusetts Dartmouth
Dartmouth, MA 02747, USA
swang@umassd.edu

Hai Wang
Sobey School of Business
Saint Mary's University
Halifax, NS B3H 2W3, Canada
hwang@smu.ca

ABSTRACT

No-code / low-code app development is transforming traditional information system development paradigms. This paper presents a pedagogical design and teaching method for no-code business app development. The contents of the teaching module include vital concepts of business app development and assignments for students. The teaching module can be customized and demands a portion of a one-credit-hour workload, embeddable in a three-credit-hour IS course. Suitable courses include introductory computer programming courses, web application development courses, or e-commerce courses. A preliminary assessment of student learning resulted in positive outcomes for this module of instruction.

Keywords: Application development, Mobile computing, Programming, System development tools & methods

1. INTRODUCTION

The information systems (IS) education area is developing new IS topics or new IS courses to meet the job market (Topi et al., 2010; Mills, Chudoba, and Olsen, 2016; Babb, Waguespack, and Abdullat, 2019; Topi 2019). After years of development of large-scale, enterprise IS in business organizations, such as ERP, CRM, and SCM systems, app development has become mainstream in the software development sector (IBISWorld, 2019; Statista, 2019). An app is a small-scale computer program or software application designed to run on mobile or desktop devices. Apps are widely used for personal productivity assistance, games, and e-commerce. The IS education literature about apps has reported the use of apps for effective mobile education and mobile game app development to improve student engagement (Dekhane, Xu, and Tsoi, 2013; Yan et al., 2014), but has few reports about how an IS program can teach business app development to help IS students enhance technical skills as well as expand the knowledge set of IS development. On the technical side, no-code / low-code app development is transforming traditional information system development paradigms and has become a highly regarded paradigm in many leading organizations (Chang and Ko, 2017; Fryling, 2019).

Because of the wide-ranging IS subjects and a restricted number of courses in an IS program, an independent business app development course seems to be infeasible and unnecessary. Hence, an instructional improvement must be considered to include a teaching module of business app development that can be embedded and integrated into an appropriate IS course. This paper proposes a teaching module of no-code business app development for IS students and describes the pedagogical design and implementation of the teaching module. The rest of the paper is organized as follows. Section 2 provides an overview of the design of the teaching module of no-code business app development. Section 3 explains the implementation of the proposed teaching module. Section 4 presents the evidence of usefulness based on preliminary practices. Section 5 offers suggestions to explain why the no-code business app development module can be useful for IS students to expand their knowledge set and skill set. Finally, section 6 reflects upon how this paper contributes to IS education.

2. THE STRUCTURE OF THE TEACHING MODULE OF NO-CODE BUSINESS APP DEVELOPMENT

The teaching module of no-code business app development teaches how to use tools to develop business apps without using computer languages for coding. The teaching module delivers the following essential concepts of no-code business app development for IS students.

2.1 Basic Concepts of No-Code Software Development

A no-code development platform allows developers to create application software using graphical user interfaces and templates instead of traditional computer programming languages (Cypher et al., 2010; Liu and Downing, 2010; Frydenberg, Yates, and Kukesh, 2018; Lee, Ross, and Kramer, 2020). Clearly, in comparison with general tools of computer programming languages, a no-code app development platform would be application-specific and less flexible. Each no-code app development platform has its own configuration and specific templates, and the developed apps have limited expandability and maintainability. Nevertheless, a good no-code development platform allows end-users to develop apps to meet wide-ranging business needs. More importantly, IS professionals and entrepreneurs can use no-code app development platforms to design app prototypes for sophisticated apps.

The teaching module explains the key concepts of no-code software development by using two simple examples: Google Sites for "drag-and-drop user-interface design" and Macro recorder in Excel for "programming without coding." Google Sites allows developers to create simple web pages without using computer languages, such as HTML and JavaScript, directly. Developers are allowed to use the templates to drag-and-drop web page elements and arrange them on a grid layout. The web page is supported by embedded HTML and JavaScript behind those web page elements. In Microsoft Excel, developers are allowed to automate a repetitive task without coding by using the Macro recorder to record a Macro and then replay the Macro whenever needed. The Macro recorder records all the steps supported by Visual Basic for Applications (VBA).

2.2 Overview of the No-Code Business App Development Platform

The teaching module not only lays a foundation for students to learn business app development but also helps students develop a better understanding of the iterative nature of application software development process.

The no-code business app development platform used in the teaching module is Microsoft Power Apps (2020) because Power Apps is a part of the Microsoft Office 365 environment which is widely available for students and provides seamless connections to Microsoft Access and Excel for business analytics exercises. Microsoft Power Apps is one of many commonly used no-code development platforms on the software market. It is easy to integrate data from various sources and has mobile support. However, Microsoft Power Apps' capacity for web applications integration is weak. Also, many of its functionalities are not straightforward for beginners.

2.3 A Diagramming Tool for No-Code Business App Development

Diagramming tools are visual languages for communication and documentation. In the IS field, diagrams have been used as effective tools for systems analysis and design in all types of systems development activities. Structure diagrams, object-oriented diagrams, entity-relationship diagrams, network diagrams, and data flow diagrams are widely used in different areas of IS analysis and design. The Unified Modeling Language (UML) (Booch, Rumbaugh, and Jacobson, 2005) provides various types of diagramming tools for software development. The present teaching module raised an important issue: what diagramming tool can be used to support no-code business app development for effective collaboration and documentation. After a thorough review of all types of diagramming tools, the authors decided to use so-called scenario design diagrams to facilitate no-code business app development. Scenario design diagrams are a variation of use case diagrams and include explicit images and icons in the context of mobile business app development. Hand-drafting of scenario design diagrams can be used for discussion during the design process. A formal scenario design diagram includes screenshots of the user-interface, resources needed, and execution results of the app. Scenario design diagrams can describe the following properties of business apps (see Table 1).

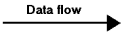
Property	Symbol
Actor	
The mobile user-interface	
Data used by the app	
Website accessed by the app	
Interaction between the mobile user-interface and data/websites	

Table 1. Scenario Design Diagram Properties

Accordingly, five types of notations are needed for scenario design diagrams to represent app scenarios: actor, mobile user-interface, data, website, and line with text which indicates the interaction between the user-interface and data/websites. As the mobile users are default actors of a business app, an actor notation is used only for an external entity other than the mobile users. The workflow of an app is represented by all linked notations in the diagram. Examples of scenario design diagrams for business no-code app development will be shown in the next section.

2.4 Three Major Categories of Business Apps

Generally, business apps can be designed for mobile users and/or for general desktop users. This teaching module focuses on mobile apps. For students to carry out a progressive learning process, students are asked to learn three major categories of mobile business apps and to develop a project which can

integrate the features of the three essential business app categories.

2.4.1 Business back-office applications – data emphasis. A back-office app allows administrative users to share, use, and manipulate organizational data for back-office business operations. In the context of personal productivity, a business back-office app supports a simple personal mobile office-automation. The data sources of back-office apps include organizational databases (e.g., Oracle, MySQL, SQL Server, Access), spreadsheets (e.g., Excel), shared text documents (e.g., SharePoint), industrial datasets in the cloud (e.g., Salesforce), or social media (e.g., Twitter). Typical back-office functions include administrative operations (accounting, human resource management, inventory control, etc.), customer services, and business analytics. After learning this aspect of the teaching module, students will be able to develop simple, mobile business apps that allow the users to retrieve business back-office data files, manipulate the data, and conduct simple business analytics operations.

2.4.2 M-commerce applications – personalization emphasis. E-commerce web portals are usually designed for desktop users and are not user-friendly for users with small size mobile devices. Mobile commerce (m-commerce) delivers e-commerce capabilities through mobile user-interfaces and special functionalities directly to customers' mobile devices. Business apps can provide personalized mobile user-interfaces for e-commerce to meet diversified consumers' needs. Common special functionalities of mobile user-interface design include mobile shopping, mobile banking, mobile ticketing, location-based services, and mobile auction. After learning this aspect of the teaching module, students will be able to develop simple apps of user-interface personalization for m-commerce.

2.4.3 Business process automation – workflow emphasis. A business app for business process automation defines a workflow that consists of a series of tasks triggered by an event. For example, when a student files a loan application, a workflow is triggered. The workflow includes selecting the responsible case manager depending upon the applicant's profile and sending application information to the case manager for approval electronically. An automated business process can be an operation of data processing across multiple data sources or integrated activities across multiple web services. Each app development platform has its unique way of building a workflow for automated business processes. Many no-code app development platforms provide visual, pre-defined, drag-and-drop logic elements for the developers to create their own workflows. However, complicated drag-and-drop logic elements are not easy to use. After learning this aspect of the teaching module, students will be able to develop simple business mobile apps for business process automation.

3. IMPLEMENTATION OF THE TEACHING MODULE

3.1 Teaching Materials

As no suitable teaching material for the teaching module can be found, a teaching note has been developed. The teaching note has its companion materials including a tutorial of Microsoft Power Apps, videos of lectures, and artifacts of business app examples.

The teaching note of no-code business app development demonstrates step-by-step processes of app development in the three categories (i.e., business back-office applications, personalized user-interface design for m-commerce, and business process automation). Several lecture video clips were developed to present the core of the teaching note. Students are able to follow the teaching note to fully understand the app development platform and no-code app development processes. The teaching module in the current form demands less than one-credit-hour workload and can be embedded and integrated into a three-credit-hour IS course. The rest of this sub-section presents illustrative descriptions of the teaching note. A detailed implementation document along with the example artifacts of the teaching module can be obtained from the authors upon request.

3.1.1 Example app of business back-office applications – data emphasis. The teaching note provides an example app of business back-office applications along with step-by-step development instructions. This category of app emphasizes data, and an example of this category is an app of back-office inventory analytics. It starts with data used in the app – a table in Excel which is stored on OneDrive for Business. The example introduces basic elements of the mobile user-interface, such as Screen, Label, Text, Button, Chart, and others, and it demonstrates the use of parameters for these objects. The scenario of this example is described as follows. An inventory data file is stored as an Excel spreadsheet in the cloud. The user, as an inventory control manager, is able to retrieve the Excel table to view/edit the inventory records and is allowed to click on a button on the mobile device to see a statistical graph of inventory analysis. Figure 1 is a scenario diagram that depicts the design of the app. Figure 2 shows the features of the mobile user-interface. The key learning points of this first example include:

- the environment of Microsoft Power Apps;
- basic elements of user-interface design, such as label, textbox, and button;
- use of Excel in the cloud for Power Apps;
- connection to One Drive for Business; and
- graph design for data visualization of business analytics in Power Apps.

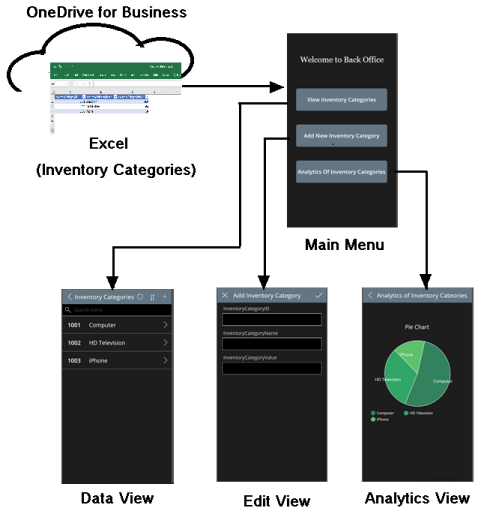


Figure 1. Scenario Design Diagram of "Back-Office Inventory Analytics" App

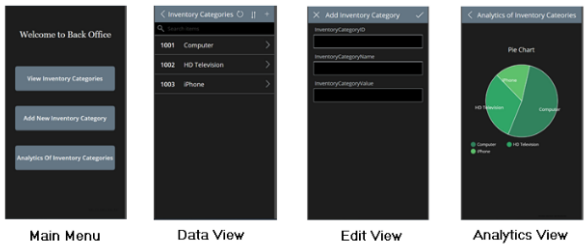


Figure 2. User-Interface Examples of "Back-Office Inventory Analytics" App

3.1.2 Example app of m-commerce – personalization emphasis. The second example is an app of a personalized user-interface for m-commerce along with step-by-step development instructions. Students will learn how to develop a user-interface for specific types of users to browse for interested products or services on mobile devices. The example app allows students to shop for certain titles of textbooks for their majors required by their departments. The scenario of this example is described as follows. The main page of the app displays several majors. The user is able to choose a major and browse all required textbooks for the corresponding major. The browsing screen displays information of each textbook, including a cover image, ISBN, book title, first author, and year. A shopping cart symbol is attached to each textbook record on the screen. The user is able to click on the shopping cart to open the bookstore website to purchase the desired textbook. Given Microsoft Power Apps' limitations, a table of textbook records is used to emulate a search function to construct the records of those required textbooks for the majors. Figure 3 is a scenario diagram that illustrates the steps of the personalized user-interface design for m-commerce. Figure 4 demonstrates screenshots of the m-commerce user-interface.

The key learning points of this example include personalization of mobile user-interface design, the concept of if-logic, and website access function in Power Apps. This example demonstrates two cases for students to learn if-logics which instruct the app to perform different actions depending on specific conditions. In the first case, if-logics are attached to the textbook's image objects so that the app loads the image objects depending upon the textbook ISBN. In the second case, if-logics are attached to the OnSelect function of the shopping cart objects so that the app loads the shopping cart with the user-selected textbook.

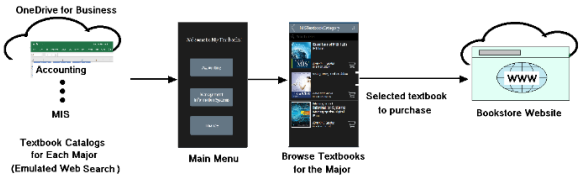


Figure 3. Scenario Design Diagram of "Books for My Major" App

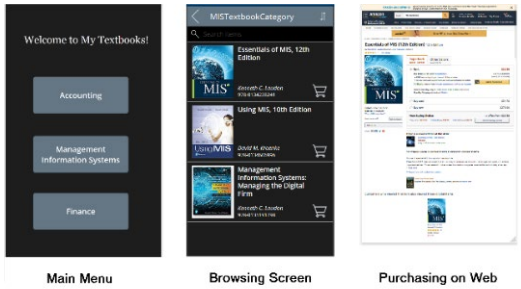


Figure 4. User-Interface Examples of "Books for My Major" App

3.1.3 Example app of business process automation – workflows emphasis. The third example is the development of an app to automate a business process along with step-by-step development instructions. This category of app emphasizes workflows. The scenario of this example is described as follows. A student loan application system has an application form for applicants to complete. The loan application form obtains application information including the applicant's name, type of loan (e.g., Federal, Private, Refinance), loan amount, and loan term. The business app allows the user to file a student loan application. The app selects the responsible loan case manager in accordance with the type of student loan application and generates an email message with the document directly to the responsible case manager for further loan application processing. Figure 5 is a scenario diagram of the design of the example app. As shown in the scenario diagram, the workflow of email transmission has two branches which are implemented by the if-logic attached to the SendEmail function in the app.

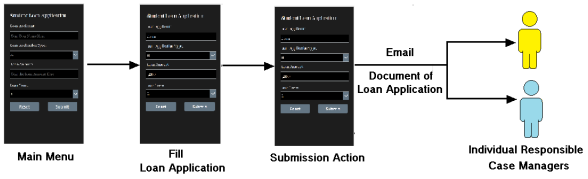


Figure 5. Scenario Design Diagram of "Student Loan Application Process" App

Figure 6 demonstrates the app that automates the business process for multiple actors and the output. The key learning

points of this example include the design of business process automation workflow, the digital documents transmission function (i.e., email in this case) in Power Apps, and the concept of if-logic in building a workflow with conditions.

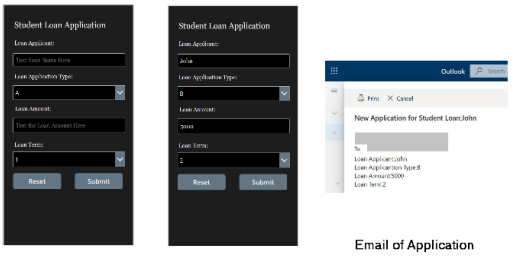


Figure 6. User-Interface Examples of "Student Loan Application Process" App

3.2 Core Competency Components of the Teaching Module

The learning goal of this teaching module is to expand the knowledge set of IS development and develop app development technical skills. The assessment instrument is an app development project that demonstrates the concepts and practical skills learned from this teaching module. The learning objectives of the teaching module should be consistent with the overall learning objectives of the host course. Nevertheless, the teaching module has its own assessment instruments and rubrics. Assessment criteria relate to the learning objectives in three aspects of competency: problem-solving, self-regulation, and creativity, as described below.

3.2.1 Problem identification and problem-solving skills. Problem identification and problem-solving skills are the ability of matching business problems and the app development tool. Students must go beyond understanding the teaching note to identify the right target business problem, factors of the problem, and prediction of the solutions.

Each student is asked to develop a business app in the context of e-commerce. Students are encouraged to integrate the features of the three essential business app categories for their projects. The problem to be solved by the app should be clearly defined by using a scenario design diagram. The scenario design diagram of a project must be checked by the instructor to ensure that the identified problem is business-oriented, meaningful, and achievable by using Microsoft Power Apps. The project documentation should be professional.

3.2.2 Self-regulation skills. Self-regulation skills are the ability to self-monitor and learn from positive experiences as well as mistakes. Learning no-code business app development highly depends upon individual learning styles and unique characteristics of cognition. Uniform lectures, general hints, and illustrative examples are the starting points of learning, but students are required to find the best way to learn no-code app development by fitting their own learning strategies and thinking structures. Each student has opportunities to use scenario diagrams to communicate with the instructor to discuss whether their project has a good scope and is achievable. The teaching module promotes the development of conscious and quick learning of new concepts and techniques.

3.2.3 Creativity skills. Creativity skills are the ability to be effortful and creative. Students are requested to set challenging, novel, and achievable goals on their own to demonstrate motivation. Creativity is measured by the uniqueness of the project as well as new elements of business applications beyond the teaching note's examples.

The competency components as the measurable learning objectives of the teaching module are summarized in the rubrics for assessment as shown in Table 2.

	4. Excellent	3. Good	2. Fair	1. Poor
Problem Identification and Problem Solving	° The problem to be solved is clearly defined ° The design of the app is optimal ° Excellent documentation	° The problem to be solved is fairly clear. ° The design of the app is correct but not optimal ° Fairly good documentation	° The problem to be solved is not clear ° The design of the app could be improved significantly ° Unclear documentation	° The problem to be solved does not match the tool ° The design of the app is incorrect ° Poor documentation
Self-Regulation (Self-Monitoring)	° Conscious learning strategies ° Quickly learning new concepts and techniques ° Counting on minimum assistance ° Thorough self-testing app	° Fairly good learning strategies ° Good learner of new concepts and techniques ° Occasional need for assistance ° Good self-testing app	° Lack of good learning strategies ° Slow learner of new concepts and techniques ° Depending on assistance ° Weak self-testing app	° No or poor learning strategy ° Poor ability to learn new concepts and techniques ° Poor utilization of assistance ° No self-testing app
Creativity (Effortful)	° Excellent scope of the app ° Creative ideas for the project ° Many new elements beyond the teaching note's examples	° Good scope of the app ° Good ideas for the project ° Some new elements beyond the teaching note's examples	° Weak scope of the app ° Lack of ideas for the project ° Lack of new elements beyond the teaching note's examples	° Poor scope of the app ° Poor ideas for the project ° Not better than teaching note's examples

Table 2. Rubrics for Assessment of the Teaching Module

4. DATA ANALYSIS, RESULTS, AND DISCUSSION

4.1 Data Analysis and Results

This teaching module was embedded in an e-commerce course, a required course at the senior level for the e-commerce concentration in the IS program of one of the authors' institutions. The 14-week e-commerce course used three weeks to teach this module. The assessment instrument used in this teaching module was a business app in the context of the e-commerce course. The app project was a small part of the course group project of an e-shop web portal development by using no-code web application development tools (e.g., WordPress) but was assessed separately on an individual basis. All IS students who completed this teaching module demonstrated their understanding of the key concepts of no-code business app development. Table 3 exhibits the learning objectives of this teaching module and the evidence of students' learning outcomes based on the performance of 17 students who completed this teaching module.

Learning Objectives		Student Learning Outcomes (Scores*)
Problem Identification and Problem Solving	Be able to identify the business problem for the app	Range: 80% - 100% Average: 87.1%
	Be able to design the app for the identified business problem	Range: 75% - 100% Average: 80.3%
	Be able to document the app	Range: 80% - 100% Average: 91.2%
Self-Regulation (Self-Monitoring)	Be able to conduct active and self-regulated learning	Range: 60% - 100% Average: 80.5%
Creativity (Effortful)	Be able to create a novel business app beyond the textbook examples	Range: 60% - 100% Average: 71.3%
	Be able to implement a large scope business app with integration of functionalities	Range: 60% - 100% Average: 75.4%

* The passing score of this teaching module was set to 65%.

Table 3. Evidence of Students' Learning Outcomes

At the end of the teaching module of no-code app development, students were asked to complete a short questionnaire anonymously. Table 4 summarizes the responses from 17 students.

Questions	Percentage of Students Who Agree or Strongly Agree
Knowledge of no-code business app development is useful for IS students	96%
The teaching module of no-code business app development enhances IS students' knowledge set and skill set	94%
The techniques introduced in this teaching module are not difficult to learn	81%
The workload of three weeks of this teaching module is appropriate	86%
The delivered teaching module of no-code app development meets your expectation of IS major study	85%

Table 4. Summary of Students' Feedback

The observations of evidence were preliminary. The opinion-based information collected by the authors might involve biases for any generalization. As no pedagogical report of teaching methods of no-code business app development in IS programs can be found at this point, any further comparative assessment of the teaching module has not been conducted. Potential subjects for future investigations include more analyses of the design of the materials in this subject and additional assessments of learning outcomes.

4.2 Discussion

4.2.1 Potential host IS courses for the teaching module. An ideal host IS course for this teaching module should be related to computer programming or web application development. An introductory computer programming course can be a candidate for the host course of this teaching module. Students can learn computer languages as well as no-code development for programming and problem solving. However, this option might make the introductory programming course overloaded. If an IS program has a higher-level programming course (e.g., web applications development with .NET or PHP), then this teaching module seems to be an ideal component of such a course. However, a higher-level programming course is usually an IS elective course with low enrollments. In the present case, the host e-commerce course is commonly taken by the majority of IS students. As the e-commerce course has a group project of an e-shop web portal development by using no-code web application development tools such as WordPress, the learning objectives of the present no-code business app development module are consistent with the overall learning objectives of the e-commerce course. Nevertheless, the rubrics of the teaching module in the current form are rather general and subjective. Specific measurable criteria (e.g., number of screens / buttons of the artifact) will be developed for improvement of the rubrics in the future.

4.2.2 Learning programming through no-code business app development. Upon the completion of this teaching module, students understand that no-code business app development does not mean "no-programming business app development." During the app development process, the developer is actually performing a programming task. The major programming tool in a no-code app development platform is an organized set of visual drag-and-drop user-interface elements instead of computer languages. Nevertheless, each of the user-interface elements has its properties that need to be described by the developer for the specific application. Many user-interface elements, especially Button, require the developer to define necessary parameters and action functions which possess features of coding such as syntax and if-logic. Thus, the distinction between no-code and low-code development is blurry. From the perspective of curriculum design, if an IS program covers no-code business app development and computer languages, it might be interesting to investigate a favorable teaching/learning sequence of both subjects.

4.2.3 The teaching module from the perspective of systems development. There have been many terminologies and interpretations of fundamental software development approaches in the literature. Systems development life cycle (SDLC) or waterfall, rapid application development (RAD), agile, extreme programming (XP), incremental development, and spiral development are commonly discussed in IS courses. This teaching module does not need to fully explore the debatable definitions of these approaches, but highlights the contrast between SDLC and RAD. Normally, an IS program has a systems analysis and design course which fully discusses the SDLC model or waterfall model of organizational systems development. Organizational systems development projects are plan-driven and involve both significant business requirements and architectural considerations. Knowledge of the traditional SDLC approach is essential at the organizational level regardless of whether the organization uses construction methods or acquisition methods. On the other hand, this teaching module allows students to have first-hand experiences in RAD for small-scale software applications. A substantive benefit of this teaching module for IS students is to learn app development using short iteration cycles of initiation, design, construction, testing, and revision. Students should be reminded that end-user development of apps is just one method of IS development with pros and cons. Nowadays, IT professionals in IT departments create no-code business apps for their organizations to bridge the gap between users' ever-changing needs and the general functionalities of the enterprise systems.

Currently, the teaching module uses the individual learning approach. It is possible to use a group project of business app development with a larger scope if the instructor chooses to do so. Such an expansion would be useful for students to learn the concept and practices of agile software development as an environment that encourages a team of people to work toward a common goal.

5. CONCLUSIONS

Systems development is an important topic in IS programs, and business app development is a topical subject for IS students. No-code / low-code system development promises to transform IS development. However, scant literature exists addressing this important industry trend; thus, a teaching module of no-code business app development has been designed and implemented in our IS programs. The teaching module explains why and what IS students need to know about no-code business app development and provides a tutorial of no-code app development with Microsoft Power Apps. The teaching module expects a proportion of one-credit-hour workload, and can be embedded in a three-credit-hour IS course.

The paper presents the original pedagogical design and contents of a teaching module of no-code business app development for IS programs. The preliminary study of students' performance and evaluations has indicated students' positive learning experiences and overall satisfaction with the teaching module. By exercising the no-code business app development projects, students had no difficulty in learning the basics of no-code business app development within a short timeframe. The progressive nature of the teaching module accommodates differing levels of preparation for learning business app development and sets the stage for students to progress to advanced levels on their own. The design and delivery of the teaching module have demonstrated that knowledge about no-code business app development is practicable and very useful for IS students. The paper can be valuable for IS instructors who wish to incorporate a practical teaching module of no-code business app development in their IS courses.

7. REFERENCES

Babb, J., Jr., Waguespack, L., & Abdullat, A. (2019). Subsumption of Information Systems Education Towards a Discipline of Design. *Journal of Information Systems Education*, 30(4), 311-320.

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *Unified Modeling Language User Guide* (2nd Ed.). Hoboken, New Jersey: Addison-Wesley Professional.

Chang, Y. H. & Ko, C. B. (2017). A Study on the Design of Low-Code and No Code Platform for Mobile Application Development. *International Journal of Advanced Smart Convergence*, 6(4), 50–55.

Cypher, A., Dontcheva, M., Lau, T., & Nichols, J. (2010). *No Code Required: Giving Users Tools to Transform the Web*. New York, New York: Morgan Kaufmann.

Dekhane, S., Xu, X., & Tsoi, M. Y. (2013). Mobile App Development to Increase Student Engagement and Problem Solving Skills. *Journal of Information Systems Education*, 24(4), 299-308.

Fryling, M. (2019). Low Code App Development. *Journal of Computing Sciences in Colleges*, 34(6), 119.

Frydenberg, M., Yates, D., & Kukesh, J. (2018). Sprint, then Fly: Teaching Agile Methodologies with Paper Airplanes. *Information Systems Education Journal*, 16(5), 22-36.

IBISWorld. (2019). Smartphone App Developers Industry in the US: Market Research Report. Retrieved January 29, 2021, from https://www.ibisworld.com/united-states/market-research-reports/smartphone-app-developers-industry/.

Lee, E., Ross, J., & Kramer, J. (2020). Teaching on the Front End: Gathering all Educators Interested in Web and Mobile Design and Development. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (*SIGCSE'20*).

Liu, C. & Downing, C. (2010). Using Cascading Style Sheets to Design a Fly-Out Menu with Microsoft Visual Studio. *Journal of Information Systems Education*, 21(3), 275-281.

Microsoft Power Apps. (2020). Microsoft Power Apps. https://powerapps.microsoft.com/en-us/.

Mills, R., Chudoba, K., & Olsen, D. (2016). IS Programs Responding to Industry Demands for Data Scientists: A Comparison between 2011-2016. *Journal of Information Systems Education*, 27(2), 131-140.

Statista. (2019). Mobile App Usage: Statistics & Facts. Retrieved January 29, 2021, from https://www.statista.com/topics/1002/mobile-app-usage/.

Topi, H., Valacich, J. S., Wright, R. T., Kaiser, K., Nunamaker, J. F., Jr., Sipior, J. C., & de Vreede, G. (2010). IS 2010: Curriculum Guidelines for Undergraduate Degree Programs in Information Systems. *Communications of the Association for Information Systems*, 26, Article 18.

Topi, H. (2019). Reflections on the Current State and Future of Information Systems Education. *Journal of Information Systems Education*, 30(1), 1-9.

Yan, G., Rawat, D. B., Shi, H., & Alnusair, A. (2014). Developing and Applying Smartphone App in Online Courses. *Journal of Information Systems Education*, 25(2), 149-159.

AUTHOR BIOGRAPHIES

Shouhong Wang is a professor of management information systems at University of Massachusetts – Dartmouth. He received his Ph.D. in information systems from McMaster University. His teaching and research interests include innovative teaching, business intelligence, semantic networks, and knowledge management. He has published over 130 papers in academic journals and several books in the MIS area.

Hai Wang is a professor in the department of finance, information systems, and management science at Sobey School of Business of Saint Mary's University. He received his B.Sc. in computer science from the University of New Brunswick and his M.Sc. and Ph.D. in computer science from the University of Toronto. His research interests are in the areas of software engineering, database management, knowledge management, and e-commerce. He has published more than 80 research articles.

Information Systems & Computing
Academic Professionals

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 2574-3872