Mobile Applications Development in Senior Design Capstone Courses

Dr. George Stefanek, Purdue University Northwest

Ph.D. Electrical Engineering, Illinois Institute of Technology

M.S. BioEngineering, University of Illinois at Chicago

B.S. Purdue University

Mobile Applications Development in Senior Design Capstone Courses

Introduction

The Computer Information Technology department had introduced a course in mobile application development in 2014 which originally taught mobile development using Android Studio and Java. In 2017, the technology used in the course was switched to Xamarin / C# using the Microsoft Visual Studio IDE. This mobile app development course enabled senior Capstone projects to include the development of mobile apps. Mobile app development projects began to be requested by participating organizations and other university research projects starting in 2016. The projects described in this paper span from 2016 through 2018 and show the wide range of applications where mobile app solutions were requested requiring both Android and iOS development. The description of the following projects include the technologies used, a brief design description, the approach the student teams used to solve the problem, the process for interacting with the client, the transition to the client, post transition status, and any lessons learned. A survey of client experiences with Capstone projects is summarized in Table 1.

Background

The development of mobile apps in Capstone projects using student teams has been studied by several researchers. Pinchot (2018) had focused on the incorporation of user-centered design, communications skills and teamwork for mobile development as a project model. It was found that focusing on these areas helped the teams complete the project with a working mobile app. User-centered design was included as a requirement. Using a pair programming model helped in development with the emphasis of having each team member contribute as equally as possible to the work. The split in work was difficult to monitor and it was found that some members in the team did not collaborate as well as others. It was also found that testing of the user-interface (UI) prior to the coding of the app would have helped. It was recommended that the UI development be done first followed by dedicated project coding.

Wong, Pepe, Stahl and Englander (2013) looked at a new collaborative Capstone model focusing on the development of a mobile hospital clinic app through a student team competition in a mobile programming course. The model consists of three elements of a Capstone course component, a business component, and an advanced technical course component focused towards enhancing the student learning experience. They found that the collaboration between the Capstone course and client enhanced the students' learning by relating the course material to a more complex real-world project which increased student motivation, performance, and problem-solving skills. The cross-course collaboration increased student preparation of professional skills required for working in industry. Finally, collaboration with an industrial client on a real-world project provided students with marketable computing skills.

Rover et al. (2014) present a case study on using an agile project management process in the development of an Android app for a client. The agile project management process used in the project was found to be beneficial to student, mentors and the clients which led to greater satisfaction and a higher quality in the senior design experience.

Wu, Hou and Tracey (2014) describe their experiences in four mobile app development Capstone projects in undergraduate and graduate courses. They found that it was important before starting the project that students equip themselves with knowledge in collaborative project development, research skills, and software engineering for design and development. They also found that it was critical for the teams to work with organizations to enhance their readiness for industry, enhance their networking with potential future employers, and get the skills in real-world mobile applications which are in high demand.

Totten (2013) describes the experience of creating mobile apps for game development for industrial clients. It was found that Serious Game client-based development projects added to the education of game designers by offering an alternative to typical entertainment-based games typically utilized in schools. The games apps were developed in collaborative group environments with several teams working near one another which enhanced student skills of analyzing problems and comparing their work with others. It was also recommended that educators implement mobile development into their existing courses that may require specific mobile app development technologies, content and approaches rather than creating general mobile development courses.

The importance and concepts regarding UX in mobile app design is described by Kuusinen and Mikkonen (2014). Enhancing collaboration on mobile app design through pair programming is described by Seyam, and McCrickard (2016) and the development of soft skills and team development is described by Brown, Lee and Alejandre (2016).

The projects described above in the background section primarily emphasized team collaboration skills for mobile app development that included team programming, teams working side by side, pair programming, and Agile project management as successful team and project methodologies that enhanced student collaborative skills that lead to successful development of the target applications. Additionally, student team interaction with clients was found to be very helpful to students as well as to the clients for the successful completion of projects and in the development of professional skills required for working in industry. Finally, the emphasis of user-centered design, the user experience and emphasis on implementing a user-interface first before the back-end programming was found to be important to the success of a mobile project. Based on this background, the mobile projects in our Capstone courses focused on using a team-based approach in doing the projects, interacting as closely as possible with the client, using the Agile project methodology where applicable, and focusing on the user experience and user-interface more heavily in the design of the apps.

Description of Projects

Project 1: Sales and Operation Planning Dashboard App

Description: The goal of this project was to create a mobile app that displayed data currently found on a Sales and Operations Planning Dashboard website of a manufacturer of compressors.

Technologies Used: The technologies used included Android studio, Java, C#, JSON, SQLite, and Android phones at version 4.0 (Ice Cream Sandwich) or higher.

Design: The app was designed using Android Studio using Java. The requirements had established that the app should be compatible with Android 4.0 version or higher. Even though the client preferred to have iOS and Android mobile versions, the students were not trained in making iOS apps so the scope of the project was limited to providing an Android app only. The mobile app displayed data received from the manufacturer's database that stored sales and operations planning data. A library of C# classes provided an API to the company's database that was used to retrieve information. A library of Java wrapper classes was written that could interface with the C# API to web services that was used to extract data needed for display in the sales dashboard application. Data was returned as JSON objects or as lists of information in CSV files. Data included information such as scheduled backlog, backlog of finished but unshipped goods, and shipped goods. Data was displayed by month and included search functions by product family and time period. Data was displayed in a grid format and in graphs.

Approach Team Used to Solve Problem: The project was lined up by a professor in charge of the Capstone senior design course at the start of the course. The professor had discussions with the manufacturing company regarding the content and scope of the project. Given, the scope and requirements of the project and the four-month timeline for the Capstone course, the professor assembled a team of four students, three of which had taken the mobile programming course and were familiar with Java Android programming. The fourth student had experience with the Java language and C# and would be able to focus on classes that interface with the web services API to the organization's database. The planning for the project was started at the start of the semester and was implemented in Microsoft Project. The student team met with the organization in a kick-off meeting to understand the project requirements better, refine the scope of work and try to get all the requirements as accurately defined as possible. The preliminary design and planning took about one month after the kickoff meeting. The student team self-selected a project manager for the project who was responsible for tracking progress and to provide status reports to the client and professor on a bi-weekly basis as well as do some programming on the project. The professor met weekly with the team to assess progress and discuss technical issues. The technical approach used was a traditional waterfall project development model that followed the project network in the sequence as planned. Each student was assigned to specific tasks and their progress tracked. There were many technologies new to the student team such as interfacing with an API to retrieve data, the use of an Android graphics package for graphing data and working with JSON files. Each of these tasks was a mini-project onto itself and required students to focus on their assigned tasks and integrate their individual contributions toward the end of the project.

Process for Interacting with Client: There was an initial kick-off meeting at the start of the project and bi-weekly on-site or telecon meetings with the client. The team and organization tried to meet on-site at the client operation once per month and the app progress demonstrated. An NDA was signed since students had access to some company proprietary data. An end-of-project meeting was held to demonstrate the fully integrated app and explain its final design.

Transition to Client: The app was not complete when delivered. The app was able to pull and display data in a grid format, had search capability, but did not include all the graphing capability as per the requirements of the project. The code and documentation were delivered to the clients' IT staff in a zip file. The client accepted the delivery of the app which their IT staff would debug, modify, add some additional features, and deploy within their IT operation.

Post Transition Status: The client was asked to fill out an anonymous survey regarding their experience with the project.   There was no other feedback and follow-up from the client.

Lessons Learned: The planning for the project should have been done earlier during the previous semester so that more time could be spent coding. This meant that the client needed to also be lined up by the end of October of the previous semester at the latest. Students would then be able to do a preliminary design and plan the project in their previous project management course and thereby start the coding of the project at least a month earlier during the next semester in their Capstone senior design course.

Additionally, the instructor should have been more careful in picking a better mix of students with better programming skill sets for Android mobile development. Even though the team had two students that had taken an Android programming course in a previous semester, both students did not have as solid a grasp of the subject matter as was necessary for the project. A more thorough interview with each of the students regarding their Android programming skills and examples of their work in the previous course may have provided additional insight into their skills. During the programming of the app, the students in this project had trouble with getting some of the fundamental app forms designed with standard controls and layouts. The back-end programming was done with a third student that had excellent programming skills recognized by the instructor and was able to successfully write code that could interface with the client's services through a class library that was provided. The instructor would need to look at all students in the Capstone course that had Android programming skills and evaluate their mobile programming abilities before-hand in order to assign a mix of students to each mobile project that would most likely be able to successfully complete the mobile project depending on its complexity.

The client had requested a user-interface design that looked similar to their website and included chart graphics. This required a third-party package to be included in the app for creating charts. The students struggled with trying to get the app to look like the website primarily because of the small phone screen size. The app had to split the view of data across different forms. The ease of use of the app and navigation through the various forms was a concern. Student familiarity with better user-interface (UI) design and the user-experience (UX) could be improved.

Finally, the client originally requested both Android and iOS versions of the app, but students were only trained on Android development. A need to explore alternate technologies or additional content for the mobile programming course was set as a priority.

Project 2: Police Department Crime Tip App

Description: The second project had the goal of creating an anonymous tip app for a mobile platform that allowed a user to send a tip of criminal or other activity to the local police department. The app would send the tip, location, and type-of-event information to the police department via email.

Technologies Used: The technologies used included Android studio, Java, SQLite, a JavaMail class library, and Android phones at version 6.0 (Marshmallow) or higher. Given the timeline of the project, an iOS version of the app was not included in the scope of the project.

Design: The system was implemented using Android Studio and Java to be compatible for Android version 6.0 or higher. A main splash screen and simple main Android form were created that included text boxes for entering a tip, location and dropdown menus for the type of tip. A button was used to send the tip. The tip was sent using email. A free Android Java mailing class library built on top of JavaMail was selected to provide an API for sending the tip as an email. All tips from the tip app were emailed from the app to an email account set up specifically for this application at the police department's mail server and managed by the police department's Information Technology services.

Approach Team Used to Solve Problem: The project was lined up by the professor in charge of the Capstone senior design course, scheduled for the spring semester, during the preceding fall semester. The professor had discussions with the police department regarding the content and scope of the project during the fall semester. Given, the scope and requirements of the project, the high visibility of the project, and the four-month timeline for the spring Capstone course, the professor assembled a team of two students, both whom had taken the mobile programming course, did well in the course, and were recognized by the professor to have the programming skill sets necessary to do the project. The planning for the project was started the previous semester in a project management course that each student had to take preceding the Capstone course. The student team met with the police department in a kick-off meeting during the fall project management course to refine the scope of work and try to get all the requirements as accurately defined as possible. At the start of the Capstone course during the spring semester the student team planned the project using Microsoft Project which had to be complete and approved by the professor during the first two weeks of the semester. The preliminary design and planning took a couple of weeks longer than originally expected as did unexpected technical issues regarding the best email Java class library to use in the project. The team self-selected a project manager for the project who was responsible to track the project and provide status reports to the professor on a weekly basis and bi-weekly to the client. The student project manager was responsible of managing two projects and therefore did not do coding on this project. The technical approach used was an agile development approach where the team created a quick

prototype within a few weeks and then continued to improve new versions of the software with each following iteration of the prototype. Also, additional features were added into newer iterations of the prototypes.

How Work was Split within each Team: During the planning of the project, a work breakdown structure was first created that specified all the deliverable and sub-deliverable items for the project. The project manager assigned individual students in the team to each of the deliverables. The student assigned the task was asked to provide an estimate for the task. The work was split approximately in half between the two students in the team.

Process for Interacting with Client: The project manager was the lead for contacting the client. Bi-weekly status updates were sent via email to the client. Any additional clarification and questions were also sent by the project manager to the client as needed. The goal was not to overburden the client with unnecessary questions and meetings. Therefore, during each status meeting with the professor, the team would inform the professor of what questions they needed answered and provided their status reports for review. The professor was copied on all client communications.

Transition to Client: At the end of the project, the student team met with the client and presented a live demo. The app was delivered to the police department to their IT department as a zip file containing all the code and documentation and as an APK file for distribution. The IT department for the police department was to install the app at their site for distribution to the public. The client was requested to sign-off on the acceptance of the system in accordance with the scope of work and requirements.

Post Transition Status: The client was asked to fill out an anonymous online survey regarding their experience with the project. The system was adopted by the police department within a half year and a press release was issued. The app was also ported to iOS by their IT staff and is currently actively used. The success of this application has led to another project with the police department for another information technology application.

Lessons Learned: The lessons learned included that the preliminary design and planning for the project and the implementation of the project tasks and project network in Microsoft Project should be moved to the preceding semester, if possible, to allow more time to be focused directly for software development. Also, the client would have liked to have both Android and iOS versions of the software which again highlighted the urgency to modify the mobile programming course with appropriate content and technologies to meet this need.

Project 3: An Android App for Inventory Tracking of Video Production Equipment

Description: This Android app was designed to track inventory of equipment used for video production by a local company. A barcode scanner checked equipment in and out of inventory using a tablet computer and synchronized the local inventory database with an online database hosted by Amazon AWS that was the central store of the inventory data.

Technologies Used: The technologies used in this system were Android Studio, Java, a barcode scanner, barcode software, AWS API to database services, and an Android tablet computer at Android version 6.0 (Marshmallow) or higher.

Design and Approach Team Used to Solve Problem: The client organization was a small company that setup video and audio equipment at music events that were being recorded for public television. The problem the company had was making sure all equipment needed for an event was sent out and then all the equipment shipped back. Equipment was often misplaced and not shipped back in an orderly manner, lost or broken. The project was lined up by the professor for the Capstone senior projects course during the preceding fall semester. The organization provided a very rough requirement of what they wanted to the professor. Based on what was understood the professor assembly a student team from the Project Management course which preceded the Capstone senior projects course. A team of three students was assembled all of which either had work experience in Android mobile development or had taken the mobile programming course. A project manager was self-selected by the team members as was a lead programmer. The lead programmer was recognized as an excellent programmer that could provide the lead on design of the application and get the team through the most difficult of the programming tasks. Planning of the project began during the preceding fall semester where a preliminary design was used to create a work breakdown structure of deliverable items and estimates assigned to each of the tasks. The tasks were implemented in Microsoft Project. The development of the project followed a traditional waterfall project management model where each task was assigned to a team member and then integration of all the pieces would occur at the end. This approach was chosen because of the complexity of the app. There were many pieces to the app such as the forms for entering products into inventory, checking equipment in and out of inventory, creation of database classes to read and write data to the local database, classes to sync local database and cloud storage using an API to AWS services, and reports. The architecture followed a Model-View-Controller (MVC) model. Each of these individual tasks required a lot of time which were not conducive to agile style development, but more traditional development where the project network was designed to sequence the tasks and each task assigned to a student. Integration would occur toward the end of the project. The lead programmer created the classes for reading and writing data to the database and using the API to connect to services for syncing the data to an AWS cloud database. The other programmers focused on form design and integrating the barcode reader into the mobile app.

Process for Interacting with Client: The client was very busy traveling to sites so onsite meetings could not be held during the course of development. Monthly telecons were setup by the team and the team sent bi-weekly status reports. The scope of work, requirements and features for the app were defined within the first month of the project. The preliminary design of the app during the preceding semester did not capture all the features desired so the plan for the project had to be re-done at the start of the Capstone course. This pushed programming back about a month at the start of the project. The student team had to do research how to use the AWS API and determine the tablet and barcode reader hardware requirements prior to purchase by the client. These tasks were done at the beginning of the project in parallel to re-planning of

the project. The students had communicated with the client via email to get feedback on their hardware choices and clarification of features. The client ordered the hardware during the second month of the project and it was received by the student team by the start of the third month of the project. In the meantime, the team was writing code as per their project plan. The client had provided the team with a full list of their equipment and all the detail for each equipment item which the team put into the database. The team also had met with the client toward the last month of the project to put barcode labels onto each piece of their equipment in inventory. The app was presented to the client about half-way through the project and also at the end of the project. The presentation was digital and consisted of sending an mpeg video file to the client so that the client could walk through the app when they had time. Also, the video presentation of the app provided a means for the client to be trained on how to use the app.

Transition to Client: The finished mobile app was installed on the tablet computer and delivered to the client. It was demonstrated to the client at the end of the project and training was administered to the client at the same time. The client signed-off and accepted the app. Also, the code was provided in a zip file along with the APK file on a flash drive.

Post Transition Status: The client was asked by the instructor to fill-out an anonymous online form for their feedback about the project. The client also began using the app after delivery. An evaluation of technologies that can more easily create both Android and iOS apps was also put as a priority.

Lessons Learned: This project required students to learn some new techniques with mobile apps that they have not learned in their mobile programming course. The database holding the inventory had to be synched to between the local SQLite database and a cloud-based database using a service from AWS. There were quite a few issues that had to be addressed to automate this syncing process which required significantly more time than originally estimated.   Students ended up implementing a manual synching process using a button click to invoke the synch when the user was sure there was an internet connection established. Additionally, the user-interface required training the user on its use and detailed instructional documentation because the user-interface was complex, and navigation was not very user-friendly and intuitive. Without training, the user had difficulty figuring out how to use the app.   Student familiarity with user-interface (UI) design and understanding of user-experience (UX) design could be improved.

Project 4: Home Health Advisor – Fitbit Interface

Description: This project was focused on implementing an interface between a mobile app and Fitbit cloud databases that could pull data captured by a Fitbit fitness tracker and a wireless scale and stored on Fitbit's cloud databases. The data was pulled from the cloud databases and transferred to the mobile device's local SQLite database using the Fitbit API. This project was part of a professor's larger research project for a smartphone app for monitoring and advising cardiac heart failure patients on their health status.

Technologies Used: The technologies used were: 1) the Visual Studio / Xamarin cross-platform development environment for programming mobile devices (MAC version), 2) the Fitbit API, 3) a Fitbit Surge fitness tracker watch, 4) a Fitbit Aria wireless scale, 5) OAuth 2.0 user authentication, 6) SQLite, 7) C#, 8) JSON, 9) XAML, and 10) Github as the code repository.

Design and Approach Team Used to Solve Problem: This project was done by a single student in collaboration with a professor for his research project. The professor initially provided the student with a scope of work which detailed the features and requirements that were required. The data that was to be collected included heart rate, steps taken, stairs climbed, weight, and body mass index. Additionally, some general mobile user-interface requirements were included to display the data, initiate a polling of the data, and login to the app. Also, the database table layout for the Fitbit data for the local mobile SQLite database was specified as well as the ability to sync the local database to a cloud database in a follow-up document several weeks after the start of the project. The mobile app was to be developed to Android version 6. Also, the code was to be implemented as class libraries that could be ported to other projects and easily used. The architecture used a Model-View View-Model (MVVM) design pattern. Security between the Fitbit cloud-based database and the application was handled using the Fitbit API's authentication requirements. Prior to accessing the information from the vendor cloud-based database, the user session had to be authenticated using the OAuth 2.0 user authentication service which is an open standard for token-based authentication and authorization on the Internet.

Additionally, technical and user documentation was to be prepared. The technologies to be used were also included along with the requirement of delivering a working app in a four-month timeframe. The student was first asked to plan the project using Microsoft Project. This required the student to do a preliminary design of the system and create estimates for each sub-deliverable item in a work breakdown structure. The work breakdown structure was implemented in Microsoft Project to determine if the work could be accomplished in a four-month timeframe. The scope of the project was slightly reduced by eliminating the syncing of the local SQLite database to cloud storage. Development followed an agile development process where small quick versions of the software were created and then refined in following versions as well as additional features added.

Process for Interacting with Client: The professor met with the student on a weekly basis during the course of the project to discuss technical issues and to track progress of the project as specified by the project plan. Additionally, progress was demonstrated by weekly code reviews with the functioning of the app demonstrated on an Android emulator in the Visual Studio / Xamarin environment. During the course of the weekly meetings, the implementation progress was also checked against the requirements.

Transition to Client: The student was required to prepare a test plan that exercised the app and checked that all the requirements were implemented and working. The app was demonstrated to the professor and verified against all the requirements to get sign-off on the project. The code and documentation were delivered by being posted to the Github repository, on a flash drive, and finally deployed on the professors' development platform and verified to be working.

Post Transition Status: The app was integrated into the professor's research project. The student had continued work on this project to create a formalized cloud-based interface based on the concept of micro-services as a layer between the app and wearable device vendor cloud services which would be used as a framework for designing future interfaces using APIs to other wearable and wireless devices. The research professor was also the instructor for the Capstone course and therefore did not fill out the survey regarding the client's experience with the project.

Lessons Learned: The synching of the local SQLite database to a cloud-based database using a service such as Azure and AWS continues to give students problems and needs to be planned carefully during the planning phase of the project. Some experimentation with the services would be beneficial to students before starting the project. It was also learned that it would save time in any future development if the interface to the vendor APIs be designed as an architecture of micro-services as a layer between the app and any vendor cloud services.

Project 5: Nursing Concept Map Mobile Application

Description: This mobile app was designed as an aid to help nursing students create graphical concept maps that represent a patient's assessment. A concept map is a useful tool for nursing students to help organize the material in a concise way. It helps a student to understand concepts since it requires a deep understanding of the material. The project was done for the Nursing department at the university.

Technologies Used: The technologies used were Xamarin / C#, Microsoft Visual Studio, SkiaSharp (a graphics package for Xamarin for doing 2D graphics), PCL Storage (provides a consistent, portable set of local file IO APIs for .NET, Windows Phone, etc.), Newtonsoft.Json (a high-performance JSON framework for .NET), and Microsoft Project.

Design: This application is compatible with Android and iOS. It allowed for the creation of concepts maps using graphical components. There were pre-defined main nodes that guided a student in the assessment process. Each node could be edited or deleted. To fill out any additional nodes, questinnaire content pages were used specific to each type of node. The questionnaire pages had a set of predefined items that students had to fill out or provide additional information. There were required and optional fields in the questionnaire. The Assessment form had categories such as neurological, endocrine, skin, hematologic/lymphatic, allergic/immunologic, and psychiatric that could be marked as assessed. Additional narrative fields were available to enter subjective and objective information, lab values to assess, and critical values. Other items such as chief complaint, diagnosis, planning, and intervention could be entered as other concept map nodes. Each concept map could be saved and accessed at a later time. The design of the system had modules with a set of classes to perform specific functions. Data for a student-created concept map was stored as a JSON file. Classes were designed to serialize objects into JSON format and parse JSON files into objects. The user-interface had Xamarin questionnaire content pages for entering node data. The content pages were implemented as a set of classes in Xamarin. There were other classes (content pages) designed that were used for the concept map graphics layout via SkiaSharp graphics and for placing

content stored in the JSON files into the graphical objects. There were a set of classes written for creating and manipulation the graphical objects. All coding was done in C# using the Xamarin framework. Finally, since the app is not intended to be used to track any specific medical information tied to a patient ID, the original prototype did not encrypt the stored data.

Approach Team Used to Solve Problem: The team was assembled based on the programming and project management skill sets required for this project which required Xamarin design experience and C# coding and familiarity with JSON. The team met with the Capstone senior design course professor on a weekly basis during the first half of the semester and with meetings with a Nursing professor once a month. The use of a Xamarin graphics package was new to the students and was not covered in the introductory mobile design course. The learning curve to create a usable and scaling user interface was underestimated in the Project planning. The project was originally planned in the previous semester in the project management course that preceded the Capstone course. Students met with the Nursing professor during the preceding semester, gathered the requirements for the app and picked a graphics package. However, it was underestimated by the students and the professor the amount of work that it would take to create a prototype application that would be usable by the nursing students. Therefore, in the second half of the semester, the project team met with the Capstone professor twice a week to carefully track progress and discuss technical issues.

Process for Interacting with Client: The student team met in-person with the nursing professor once a month during the course of the project. Students communicated their questions via email. Status reports on the project were sent via email to the nursing professor and course instructor bi-weekly.

Transition to Client: The student team delivered the app as an APK file to the nursing professor at the end of the project and provided documentation in a PDF file. The APK file was installed on the professor's phone and archived onto the professor's computer. A live demo was provided to the professor on how to use the app.

Post Transition Status: The client was asked to fill out an anonymous online survey regarding their experience with the project. Even though the main framework for the app was completed, it was deemed not to be ready for use by nursing students until some issues could be resolved. Some of these issues were: 1) text did not scale properly and could exceed the text-bound rectangles on the Android platform, 2) multi-line text could not be entered into the SkiaSharp graphics objects, 3) some objects such as a button could disappear after back button navigation in iOS, 4) there were panning problems, 5) touch to select was not working correctly in all instances, and a few other issues. Therefore, it was decided that another senior design team in the following year, if possible, would continue the project before it could be implemented for use. Also, encryption of the data stored on disk would need to be implemented as well as authentication into the app to address the problem if a student inadvertently entered medical information and tied it to a patient by their id or name.

Lessons Learned: The lessons learned were that there was a lot more risk to the timeline and in the amount of detail that had to be handled using graphics in a mobile application. Even though research on using the SkiaSharp graphics library was done in the preceding semester, the estimates for the tasks on the projects were underestimated by both the professor and student team. Future projects would need to be very careful in analyzing the complexity of the graphics needed in any future applications. Additionally, focus on the user interface and the user experience (UX) aspects of such an app need to be more strongly incorporated into such projects.

Analysis

The organizations involved with the senior projects were asked to respond to a survey regarding their experience with the projects. There were six organizations that responded between 2016 and 2018. The process of doing projects other than mobile projects followed the same processes as described in this paper. Table 1 summarizes the results of the survey.

Table 1: Client Satisfaction Survey Results

Question	Strongly Agree	Somewhat Agree	Neither Agree or Disagree	Somewhat Disagree	Strongly Disagree
Q1	3	3			
Q2	4	1		1	
Q3	2	2	1	1	
Q4	6				
Q5	5	1			
Q6	5	1			
Q7	5				1
Q8	4	1	1		
Q9	4	2			
Q10	3	1	2		
Q11	5	1			

Q1: Did the student team provide you with frequent enough updates regarding status of the project?
Q2: Were the updates detailed enough?
Q3: Were there enough face to face meetings between your organization and the student team?
Q4: Did the student team interact with your organization in a professional manner?
Q5: Did the students deliver the project application code and documentation to you?
Q6: Did the students install the application either at your site or via an online server in a format that is usable by you?
Q7: Did the students perform enough training with you to explain how to use the application?
Q8: Were the requirements for the application defined adequately?
Q9: Did the application meet the requirements that were defined?
Q10: Does the application run without errors?
Q11: Are you satisfied with your overall experience in working on a senior project with PNW students?

Limitations: The dataset was small and focused on the few projects that were completed and the limited feedback that was obtained from a subset of the clients. Also, the survey design could have been improved. The survey asked respondents for agreement or disagreement with the stated questions. It may have been more appropriate to ask for responses using a scale of "Definitely Yes" to "Definitely No" or something similar.

 "A total of seven mobile projects were completed over several years in the Capstone courses. The primary contact for each client organization was asked to complete the survey. A total of six clients completed the survey. The seventh mobile project which is the fourth project out of the five described in this paper did not complete the survey since the instructor for the course and the client were the same. Out of the other six mobile projects, four are described in this paper. The two other mobile projects that were not described in this paper were mobile website-oriented projects. Since the survey was anonymous, the specific projects described in this paper could not be correlated to the specific responses in the survey, therefore all six responses for all six surveyed projects were included. All projects focused on using a team-based approach in doing the projects, interacted as closely as possible with the client, and used the Agile project methodology where applicable. This approach was followed for all Capstone software projects regardless of the technology used."

Results Analysis: The results show that 6 out of 6 responses either strongly agree or somewhat agree to 6 of the 11 questions. Also, 5 out of 6 responses either strongly agree or somewhat agree to 9 out of the 11 questions. These responses support that the process of doing these information technology projects is mostly successful since the clients had favorable answers to the majority of questions. There were only 3 responses to questions 2, 3 and 8 where there was disagreement. Each of these questions had only one client responding with disagreement. The first two were related to not enough interaction with the client and question 8 was regarding the definition of the requirements.

Conclusion

Over the last few years where mobile app projects were undertaken in the Capstone senior design courses, improvements have been made in the process of doing such projects. Since as much coding time as possible is needed for these type of projects, the projects are now lined up in the preceding semester where students have meetings with the clients, gather requirements and determine the scope of work to help them do a preliminary design and plan their projects using Microsoft Project prior to the start of the Capstone senior projects course. This change in process has worked well because the preceding course in Project Management allowed for this early planning to be done. Additionally, more care is taken in selecting the right mix of students with appropriate programming, database, UI, and technical project management skill sets. Status reports to the client are first reviewed by the course instructor and monitored to go out on time. The mobile course was also modified from its use of Android Studio and Java development to the Xamarin technology for cross-platform mobile development where a single Xamarin / C# application is written that creates Android, iOS, and Universal Windows project code. This environment is a much more flexible platform that can be used to meet the needs of different

clients in future senior projects. Also, the use of technologies that were unfamiliar to students such as graphics packages would require careful planning and estimation of these tasks prior to beginning a project. Additionally, early planning and careful estimation of high-risk tasks requiring unfamiliar technologies may lead to a reduction in the scope of work for a project so that it can be fully completed and debugged prior to delivery. Finally, many of the apps could benefit from more emphasis on UX and UI design to provide a better user experience.

References

Brown, Q., Lee, F., & Alejandre, S. (2009, April). Emphasizing soft skills and team development in an educational digital game design course. In *Proceedings of the 4th international Conference on Foundations of Digital Games* (pp. 240-247). ACM.

Kuusinen, K., & Mikkonen, T. (2014, August). On designing UX for mobile enterprise apps. In *Software Engineering and Advanced Applications* (SEAA), 2014 40th EUROMICRO Conference on (pp. 221-228). IEEE.

Pinchot, J. (2018). A Mobile App Development Project Model Emphasizing Communication Skills, Teamwork, and User-Centered Design. *Issues in Information Systems*. 19(3), 22-32.

Rover, D., Ullerich, C., Scheel, R., Wegler, J. & Wipple, C. (2014). Advantages of agile methodologies for software and product development in a capstone design project. *IEEE Frontiers in Education Conference (FIE) Proceedings*. Madrid, Spain.

Seyam, M., & McCrickard, D. S. (2016, February). Teaching Mobile Development with Pair Programming. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (pp. 96-101). ACM.

Totten, C. (2013). Teaching Serious Game App Design Through Client-based Projects. In *DiGRA Conference*.

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

Wu, S., Hou, X. & Tracey, K. C. (2014). Use of Mobile Application Development Technologies in Capstone Projects. *Proceedings of The 2014 IAJC-ISAM International Conference*.