

Supporting Object-oriented Design Learning Outcome Using an Android Development Project

Asjia Marion-Bethany Gilder, Alabama A&M University

Asjia Gilder, a native of Millbrook, AL received a B.A in Chemistry, with a minor in Computer Science from the Alabama Agricultural and Mechanical University in May of 2018. Currently, she is a Graduate Student at Alabama Agricultural and Mechanical University, and is continuing her studies to receive a M.S. in Computer Science.

Mr. Wichien Choosilp,

Wichien Choosilp is a graduate student of the Department of Electrical Engineering & Computer Science at Alabama A&M University. Currently, Wichien is working on the thesis of Privacy Protection and Malware Detection of an Android System towards a Master degree of Science.

Mr. Michael Allen Jr.

Hello my name is Michael Allen, I am a graduate student of the department of Electrical Engineering & Computer Science at Alabama A&M University. Currently I am working on my thesis of Mobile security towards a Master degree of science.

Mr. Quantarius Jerome Dean

My name is Quantarius Dean and I am a graduate student of the Department of Electrical Engineering & Computer Science at Alabama A&M University. I am currently working on the thesis of Machine Learning and Deep Learning towards a Master degree of Science.

Dr. Yujian Fu P.E., Alabama A&M University

Dr. Yujian Fu is a professor of computer science department at Alabama A&M University. Her research interests fall in formal verification of cyber-physical systems, behavioral analysis of mobile security, software architecture and design analysis of safety-critical and mission-critical systems. Her projects are supported by NSF, Air Force, and DoD. She has several publications regarding research and educational projects.

Supporting Object-Oriented Design Learning Outcome Using Android Development Project

Abstract

In the past decades, object-oriented programming and design concepts have become the most prominent programming paradigm used by programmers and application developers in today's time. Object oriented programming is a programming language model that is organized as objects which integrate data and actions to describe system logics. The idea of object-oriented programming gained momentum in the 1970s and Java has been widely used in many applications and domains. The concepts of object orientation have been adopted in many other languages such as C#, Python, Swift, and so forth. With the fast development of mobile applications, Android programming became one of the most popular platforms due to the broad adoption of mobile devices in medical, civil, military and construction areas. Project based learning method have been proved to improve and boost students learning of concepts effectively. This paper present an experimental study that use Android development to help students master the object-oriented design and programming concepts. The implemented mobile app is a smart student registration system that will facilitate registration, class schedule, and academic advising, making it easier for students and administration. The study will demonstrate the importance of object-oriented design and how it guides the programming in mobile development, and on the other side, the mobile development will improve understanding of object orientation concepts.

Keywords

Object oriented design, Android programming, project-based learning, software testing, programming language.

Introduction

Android Inc. was founded in Palo Alto, California, in October 2003 by Andy Rubin, Rich Miner, Nick Sears, and Chris White. Rubin [24] described the Android project as "tremendous potential in developing smarter mobile devices that are more aware of its owner's location and preferences". Android is a mobile operating system developed by Google, based on a modified version of the Linux Kernel, and other open source software. It is designed primarily for touchscreen mobile devices such as smartphones and tablets. Applications, which extend the functionality of devices, are written using the Android software development kit and, the Java programming language. The software developing kit includes a comprehensive set of development tools, including a debugger, software libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. In December 2014, Google released Android Studio, based on Intelligence IDEA, as its primary IDE for Android application development. Other development tools are available, including a native development kit for applications or extensions in C++, Google App Inventor, a visual environment for novice programmers, and various cross platform mobile web applications frameworks.

The basis for OOP (Object-oriented Programming) started in the early 1960s [12, 13]. A breakthrough involving instances and objects was achieved at MIT with the PDP-1 [6], and the

first programming language to use objects was Simula 67 [6]. It was designed for the purpose of creating simulations and was developed by Kristen Nygaard and Ole-Johan Dahl in Norway. The term "object-oriented programming" was first used by Xerox PARC in their Smalltalk programming language [6]. The term was used to refer to the process of using objects as the foundation for computation. There are a number of other languages that have successfully combined the object-oriented approach with procedures that are useful to programmers. Python is one example, and Ruby uses a similar approach as well. The use of an object-oriented approach has led to advancements in modeling languages, design patterns, and a number of other areas. It is likely that OOP is a programming paradigm that will continue to evolve as we move forward into the future. It is a powerful language which has continued to improve over the years. It is the subject of debate within the programming community, as critics point out a number of problems with the structure. However, the popularity of programming languages such as Java demonstrate that it is a paradigm that is here to stay.

The objective of the program is present a strategy that will use Android app development to help students master the object-oriented design and programming concepts. The implemented mobile app is a smart student registration system that will facilitate registration, class schedule, and academic advising, making it easier for students and administration. This program will allow the user to open a portal and input their identification number and password to sign in. Then from that portal another window will open giving the user the option to enter a pin for registration or look up classes. Students cannot register if they do not have a pin. After, another window will open allowing registration, and the options to add or drop classes. Similar functions will be available for administration.

Related Work

The project is related to two main categories of the research. One is project-based learning in object-oriented design methodology. Another is the Android programming in object-oriented design domain. In this section, we will discuss this in these two groups.

Project-based learning in Object-oriented design

Object oriented techniques incorporate different software engineering techniques that can be used in development of all projects. Unified Modeling Language (UML) is a structured method which involves formal methodologies and has made significant contribution in defining sets of methodologies that can be applied to any software engineering effort. Our approach is to make students experience the real time application of Object-Oriented Modeling and Design methods using Project-Based Learning tools that help them to have a hands-on experience with the operational software design [11, 25]. Project-Based Learning (PBL) is an active learning approach that was developed by the physician Howard Barrows. [7,8] It is a teaching method used to educate students for development intellectually and emotionally. PBL uses real-world scenarios, challenges, and problems so that students can gain useful knowledge and skills while performing their projects [30, 10, 9]. Object-oriented programming (OOP) is central to software development in modern software industry, and it requires more than the mere concept understanding. Instead, it requires mastering high-level skills of analyzing, designing, implementing and validating software. All which can be done in computer science. PBL [31, 30] is a very import learning tool in computer science. It helps students increase their understanding

of computer science materials while working together on complex real-world projects. Most computer science departments incorporate project work into their degrees.

Android programming in Object-oriented design domain

Several research works were done in this area in recent years. In [26], a small UML profile is developed and UML activity diagrams are presented to translate to a state machine to achieve an executable Android application. In [27], GenCode uses UML sequence diagrams to represent application behavior where UML standard notations are adopted. Min et al. [28] proposed an extended metamodel for Windows Phone application modeling based on the Model View Controller framework. Model View Controller framework is a CAD design tool focuses on the structural aspects of the system. Son et al. [29] proposed an approach for modeling Windows Phone applications based on class and sequence diagrams. The limitation of [29] is that the authors firstly proposed to model application using new stereotypes and then transform this model in a target specific model.

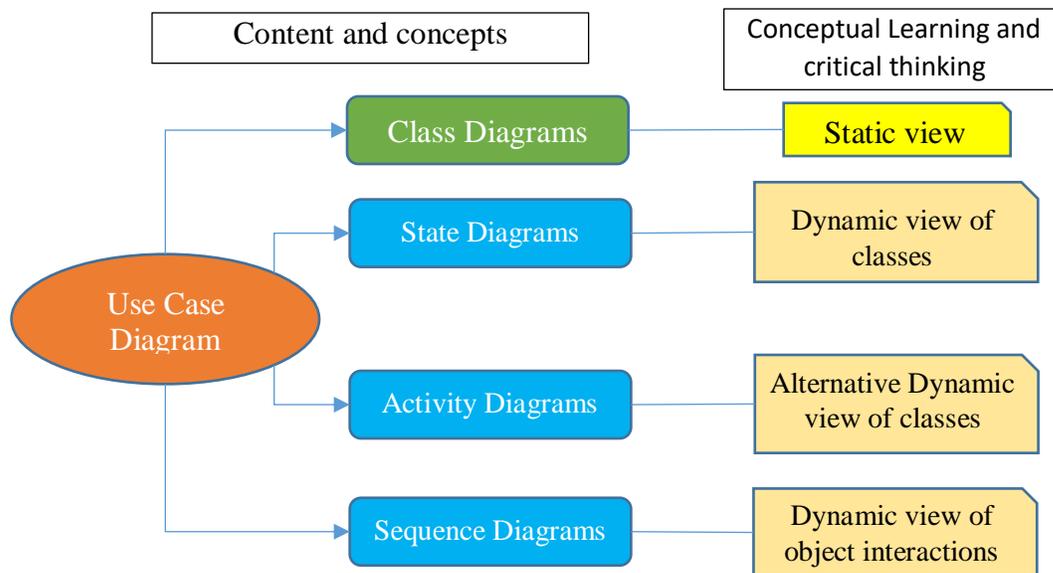


Figure 1. Overview of the system design architecture.

Student Registration App

Student registration app aims at supporting and facilitating students to register classes from anywhere using mobile device. The benefit of this design will be reducing the work load of registration staffs and reduce the budget and errors caused by humankind.

The student registration app is designed in three components – interface (graphic design, web design and user input), functionalities, and database (storage of classes, student, professors and administrative). This app was designed based on simplicity. We intended for it to be easy to use but also efficient. When designing the app, we wanted it to allow the user to input their log in information to enter the system. From the information entered at the log in screen the system will be able to determine if the user is a student or a professor. If the user is a student, they will have to option to search classes. After searching, the user will be promoted to enter a pin which is required for registration. After entering the pin, the user will have the option to add classes.

When adding the class, the system will then check the availability of the class. If the class is available, it will be added to the user's schedule, if it is unavailable a message will occur stating the class is unavailable. The user will then have the option to view their class schedule. If the user is a professor, upon logging into the system they will be able to view the classes they are teaching, see their students in each class, and view the students schedule.

System Design Architecture

Use case centric design is one of the major object-oriented software development (OOSD) approaches. In here, we will find out how to apply this design approach to our project as a case study. In this section, we will present use case diagram, after that, class diagram and state diagram are introduced. To elaborate the dynamic interaction, we have sequence diagrams developed. Finally, in order to fine the state diagram from the operational point of view, we will introduce activity diagram. The overall design architecture is shown in Figure 1.

System Design Models

From the above section, we will introduce our design models in this section in four subsections.

Use case diagram

A use case diagram is a representation of a user's interaction with the system and shows the relationship between the user and the different use cases in which the user is involved [12]. The purpose of use case diagram is to capture the dynamic aspect of a system. Dynamic aspect means the behavior of the system when it is operating.

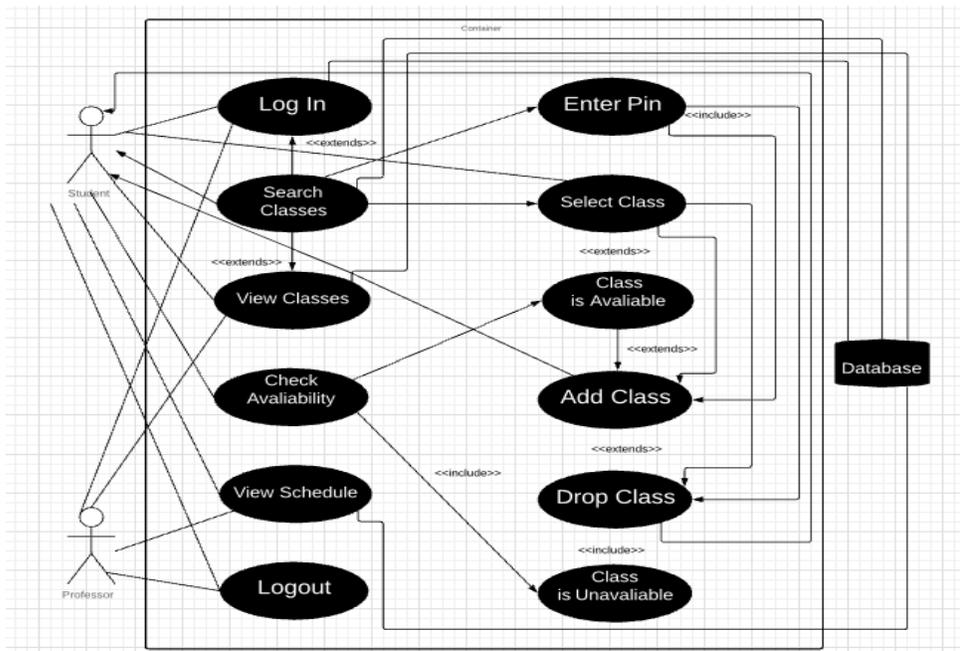


Figure 2. Use case diagram of the system design

Use case diagrams are used to gather the requirements of a system including internal and external influences. Therefore, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified. In Fig. 2, you will see how the two actors (student and professor) work throughout the system. Also, you will see which cases are intended for the students and which cases are intended for the professor. Each actor has line extending from them to a case which they relate to. The cases are the functions that the actors are allowed to do. For example, the “Student” actor has a line that goes to the log in case, meaning the student is able to log in. The log in case extends to the case search classes. Extend is a directed relationship that specifies how and when the behavior defined in usually supplementary extending use case can be inserted into the behavior defined in the extended use case [18]. Another relationship used in Fig. 2 is include. Include was used between the enter pin and add class use cases. Include is a directed relationship between two use cases which is used to show that behavior of the included use case, the addition, is inserted into the behavior of the including the base use case [18]. This display gives a clear flow, making implementation easier.

Class diagram

A class diagram is a static diagram. It represents the static view of an application. Class diagrams are not only used for visualizing, describing, and documenting different aspects of a system, but they are also for constructing executable code of the software application. A class diagram describes the attributes and operations of a class, and also the constraints imposed on the system. Class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Our class diagram is shown in Fig 3.

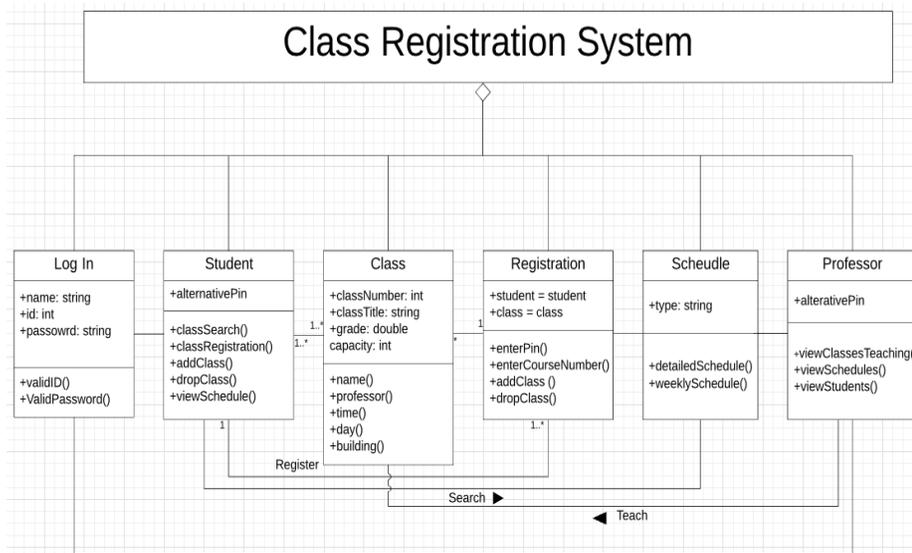


Figure 3. Class diagram of the system design

The diagram in Fig. 3 is a representation of what functions we wanted the app to perform. This diagram provides analysis and design of the static view of our system. With each class stated, you will find the responsibilities it will have in the system. Also, you will see the relationships the classes have to one another. Each class created has an aggregation relationship

to the main class (Class Registration System). Aggregation is a relationship between two classes that is best described as a "whole/part" relationship [20]. The aggregate class contains a reference to another class and is said to have ownership of that class [15]. A common relationship in the class diagram is association. Association defines a relationship between classes of objects that allows one object instance to cause another to perform an action on its behalf [19]. In our design, association relationships can be seen between the "Student" class and the "Class" class, the "Schedule" class and the "Professor" class, etc. This class diagram will start off as a model and translate into implementation code. For example, take the "Student" class and the "Registration" class. We created the "alternative pin" attribute because in order for the student to register for class the need a pin. When that pin is entered the student will be able to search classes, add classes, drop classes, and view their schedule. These are the operations of the student class. It associates with the registration class because in the registration class you can perform the same operations. These two classes have a "1 – 1..*" multiplicity. This means one student can register for one or more classes. This diagram makes implementation easier because it gives you a visual layout for implementation.

Sequence Diagram

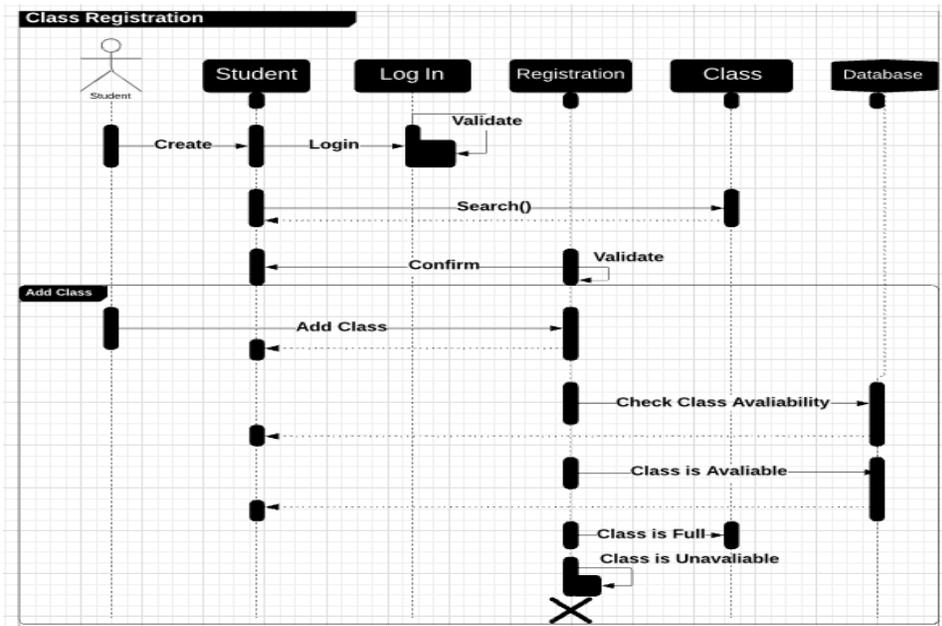


Figure 4. Sequence diagram of Registration system

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario [15]. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. A sequence diagram show is displayed as parallel vertical lines. These lines are considered lifelines. Lifelines are different processes or objects that live simultaneously, and, as horizontal arrows. Messages are exchanged between them in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

As shown in Fig. 4 you will see how the user interacts with objects that belong to the GUI. At the start of the sequence diagram the user enters the system by logging in. After logging in, the log in information will be validated and if validation goes through the user will have successfully entered the system. After entering the system, the user will then be given the option to search for a class. When searching, the interaction will be sent to the class object, and returned back to the student object. The search will then need to be confirmed and validated, and again returned. After validation is completed, the user will have the option to add the class. When adding the class to the system, the system will do a check to see if the class is full, if the class isn't full it will be added to the student schedule, otherwise, a notice of the class is unavailable will be issued. The diagram in Fig. 4 give a clear sequence when it is time for implementation.

Activity Diagram

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram [16]. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes, as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data store.

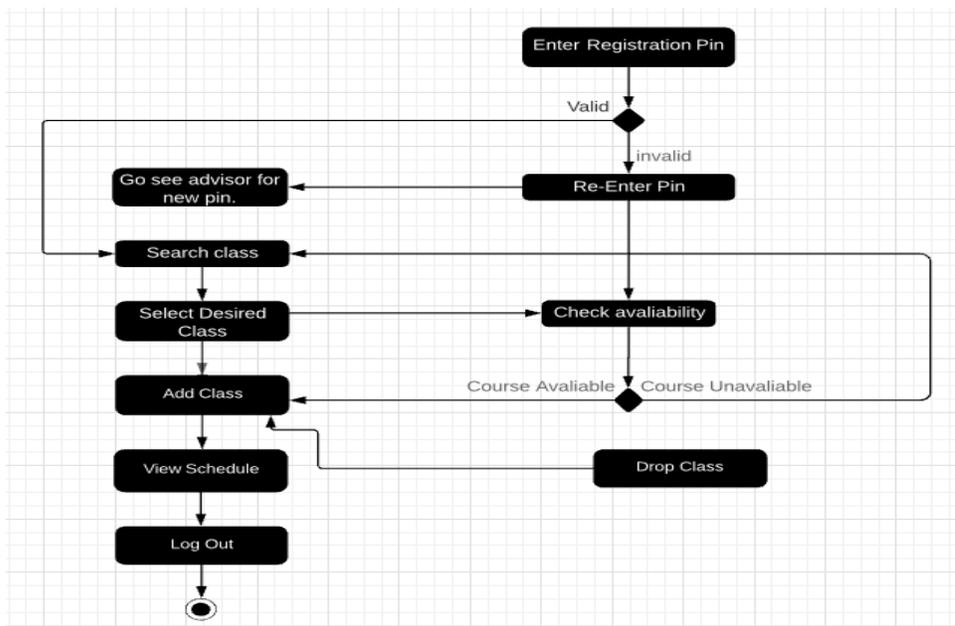


Figure 5. Activity diagram of registration system

The diagram in Fig. 5 shows the message flow from one activity to another. The diagram in Fig. 4 shows the process of the user entering the system to register for classes. They will begin by entering a pin which will then need to be validated, if validation goes through, the message will be sent to allow the user to be directed to search for a class. If validation isn't complete a message will show telling the user to re-enter the pin. This is important because if the message flow doesn't work, the implementation will not work.

System Implementation and Results and Discussion

With this program implementation getting the design of what we wanted was less difficult with the android studio. It's difficult with programming certain items within the application but overall the design was the easiest of them all. On the create login section of our program we wanted the look of an Alabama A&M sign up screen and we wanted it to run almost the same way. For example, in the code we had to set the background style and design with the image view. Then the next thing on this registration page was the two buttons. The first button is the sign-up button and the second button is the sign-up button. The sign-up button is supposed to take you to the page of the application where you get to sign up for your user name and password. And the sign in button takes you to the part of the application where you get to use your user name and password and decide whether you're an administrator. With the sign up button it takes you to the page where you get to create an account, and it allows you to sign in and register.

Android API

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ's. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Instant Run to push changes to your running app without building a new APK
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK support

Android Configurations

Creating custom build configurations requires you to make changes to one or more build configuration files, or build.gradle files. These plain text files use Domain Specific Language to describe and manipulate the build logic using Groovy which is a dynamic language for the Java Virtual Machine.

- Image View – Displays image resources, for example Bitmap or Drawable resources. ImageView is also commonly used to apply tint to an image and handle image scaling
- Button – A user interface element the user can tap or click to perform an action.

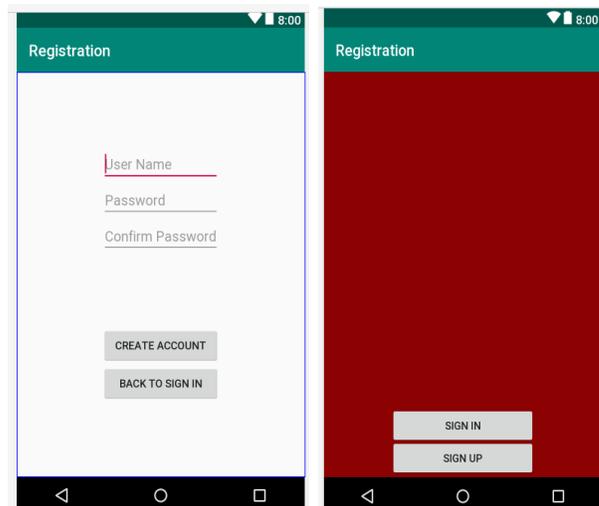


Figure 6. Two snapshots of the implementation

Our current implementation can be described in the following two main categories – interface design and function development. Our interface design is shown in Fig. 6, where are supported by Android gridle and xml design. A snapshot of button implementation is shown in Fig 7.

```

<ImageView          android:id="@+id/imageView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/buttonSignIN"
    android:layout_alignParentEnd="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:layout_gravity="center_horizontal"
    android:layout_weight="0.69"
    android:background="@drawable/aamu" />

<Button
    android:id="@+id/buttonSignIN"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:layout_alignStart="@+id/buttonSignUP"
    android:layout_alignLeft="@+id/buttonSignUP"
    android:layout_alignParentBottom="true"
    android:layout_marginBottom="41dp"
    android:onClick="signIn"
    android:text="Sign In" />

```

Figure 7. SignIn button in mainfest

Several critical functions are implemented such as sign in, add course, drop course, search courses. With the design models, we found out several interesting things during development. First, design models help us to understand the system wide structure, interaction and outline the development from static view to dynamic view. Second, design models provide guidance of the program from a high level point of view. It takes a lot of time for us to elaborate the design models in order to find out how to correctly implement. Once design models are completed, it will help to improve development and reduce the programming tasks. Thirdly, the implementation can help improve the design models by identifying some unrevealed issues or errors (partially).

System Validation

It is very important for the developer to know how to validate the program and ensure the quality of software (intensive) systems. It has been not easy job to express the concepts of software testing in the current computer science curriculum. Even though right now, many computer science departments or programs does not have software testing courses offered for undergraduate curriculum, where Alabama A&M University computer science program is one of them.

By saying these, testing mobile system is more challenge with such fast growing development in mobile devices. A testing program is a piece of software, which executes with the tested software. It validates that the code results in the expected state or executes the expected sequence of events. Evolved from WRESTT-CyLE[1], STEM-CyLE is new learning environment that helps students to learn software testing concepts outside the textbook and classroom by integrating the content with the cyber enabled learning environment. This platform has several components which are combined as a unified learning units. Each learning unit presented with content (including general information and pages), exercises questions, assessment (quiz), and comments. These learning units wraps the high level concepts of software testing to small individual items to help learners and students to master the concepts in an easy way. The STEM-CyLE is a good supplementary learning platform to help us to master the concepts that are not included in the textbook.

Unit Testing

Software unit tests help the developer to verify that the logic of a piece of the program is correct [14]. Running tests automatically helps to identify software regressions introduced by changes in the source code. Having a high-test coverage of the code allows programmer to continue developing features without having to perform lots of manual tests. In this project as Java programming was used to create a program for student registration. For this particular project, JUnit testing was used to test the program application [22]. Junit [21] is a unit testing framework for the Java programming language, that plays a huge role in regression testing. Regression testing is a form of software testing that confirms or denies a software's functionality after the software undergoes changes, and it uses an open source framework. An open-source framework is used to write and run repeatable tests.

Feedbacks of STEM-CyLE.

When working on this project, we were rewarded with many educational tools to aid the process of creating our app. As mentioned before, STEM-CyLE is a learning and engaging cyberlearning environment that contain digital learning content in several areas of computer science program [1, 3, 33, 34, 35]. Each learning unit in STEM-CyLE is composed of learning content, practice, quiz and assessment. Currently, STEM-CyLE covers programming language learning units, cyber security, other than the major component of software testing[1, 2, 35]. STEM-CyLE is a cyber-learning environment that uses assorted learning and engagement strategies including collaborative learning [32], problem-based learning, gamification [3, 35, 36], and social interaction. The fusion of gamification, social interaction, and collaborative learning in STEM-CyLE helps students explore and discuss topics in software programming content [37], software tools, and a variety of other areas [1, 4, 34]. A number of testing tools were introduced during the course period to support understanding of the knowledge units in the textbook. For example integration testing, black box testing, equivalence testing, white box testing, and Junit testing. A short summary of some testing techniques are listed as following:

- **Black-Box Testing:** tests the internal structure, design, and implementation of the system. These three factors will remain unknow to the tester.
- **Integration Testing:** is software testing where individual units are combined and tested as a group.

- Equivalence Testing: are a variation of hypothesis test used to draw statistical inferences from observed data.
- White-Box Testing: is a method of testing software that test internal structures or workings of an application as opposed to its functionality.
- Junit: is an open source unit testing framework for java that developers use to write and run repeatable tests.

These learning units in the STEM-CyLE provides a very convenient way to learn and master the concept of software validation and testing. Furthermore, it helps us to determine which form of testing we be the best to use for our project.

The use of STEM-CyLE helped us understand and master the software testing concepts and tool Junit. As stated by some students, this platform gave us a better understanding of the requirements needed when testing a program. For example, an understanding of what testing is, unit testing, and what test case is and what it consists of all played a role in testing our program. In addition, stem cycle gave us a better understanding on the requirements needed for our solutions to solve our problems within our program. Our group think that STEM-CyLE was a big help with introducing us to the purpose of JUnit as well along with the features of JUnit. Therefore, STEM-CyLE helped us master the total concept of software testing along with the understanding of Junit testing.

Testing Steps

In our project we used Junit testing to test our android program. We began with creating a few java classes. The classes have their own responsibilities within the program along with their specified coding. We began with creating a test class called TestRegistration.java with the addition of test method, Annotation, and implementation to the test conditions. The test method testPrintMessage () was used to test the java class along with the annotation @test. We then implemented the test conditions and checked the conditions using assertEquals API of Junit to test the program. Once we were done with the first java class, we then moved on to creating the test runner class for Junit testing called RegistrationTestRunner.java. This class was responsible for executing the test cases. It imported the JunitCore class and we used the runClasses() method to take the test class name as its parameter. Once we finished to coding, we compiled the test case and test runner classes using java and verified our output.

Assessment of Student Learning Outcomes

To assess the learning outcomes with the project based learning, the regular assessment such as quiz, tests, homework are adopted. In addition to this course, a control group that PBL was not adopted was used as compared data. A class wide overview of the conceptual learning from two groups of data were collected. The experimental group has shown clear improvement in the concepts of design models.

In addition, students' programming skill was dramatically improved. Some students did not learn Java before attending this course. However, at the end of the class, most of students can complete Java program properly without much help.

Conclusion and Recommendations

The objective of this program was to present a strategy that will use Android app development to help students master the object-oriented design and programming concepts. The implemented mobile app is a smart student registration system that will facilitate registration, class schedule, and academic advising, making it easier for students and administration. Although the purpose of this project was to create an app, this group was also able to get a better understanding of Object-Oriented Programming, and the role it played in creating this app. Object oriented programming is a programming language model that is organized as objects which integrate data and actions to describe system logics, which is basically what this app was. Many different objects for registration were integrated to form an app that makes registration easier for students. When working on this app, the Android studio software was used. Neither myself, nor my group partners had used this software before, so that proposed a challenge.

The early stages of creating this app were strictly fundamentals and learning how to work the software. Once that was mastered, implementation came in to affect. When performing the implementation for the project we struggled because we kept receiving errors in the code. We weren't sure if we were receiving errors because our code was incorrect, or it was the android studio software. We were having issues with downloading the android studio software and getting "test" code to run, so this caused us to delete the software and start again. After doing this and still receiving errors we decided to redo some of the code. To do so, we needed to go back and review the design of the system with the diagrams we created. We discovered the implementation and the diagrams did were not correlating. Because of this we re-worked all of the diagrams adding and removing features to better suit the app. After all this was done, we had fewer errors, but still the program wasn't error free.

Unit testing is able to reveal many bugs embedded in the program. With the support of the formal models, the assertions specified in the Junit can be properly encoded and inserted into the program. Many critical properties, such safety and security of the system, can be validated. For instance, this app was supposed to allow the user to open a portal and input their identification number and password to sign in. Then from that portal another window was supposed to open, giving the user the option to enter a pin for registration or look up classes. From there, the user will then be able to search courses and either add what was needed or drop what wasn't needed. These sequence of operations can be specified in the program as assertions. Once the program is running, Junit will check if these assertions are satisfied or not. Furthermore, after the necessary classes were added or dropped, the user would then be able to view their classes for the semester. These are critical functions of the registration system where correctness must be ensured and testing is the best validation method. After testing, the program was successful in opening a portal and allowing the user to enter their user name and password. Also, it was successful in letting the user search the classes he or she needed.

Acknowledgments

We would like to thank the reviewers for their valuable comments. This work was supported in part by the National Science Foundation under grants DUE-1525120, DUE-1225654, and DGE-1419295.

References

- [1] Peter J. Clarke, Debra L. Davis, Raymond Chang-lau and Tariq M. King. 2016. Impact of Using Tools in an Undergraduate Software Testing Course Supported by WReSTT. ACM Transactions on Computing Education (TOCE). 17, 4 Article 18 (August 2017).
- [2] Yujian Fu, Nelson Barnes and Peter J. Clarke. Integrating Software Testing into Computer Science Curriculum Using WReSTT-CyLE. The 123rd ASEE Annual Conference & Exposition. - Computing and Information Technologies (CIT) , June 2016. Paper Id: 15885.
- [3] Yujian Fu and Peter J. Clarke. Gamification based Cyber Enabled Learning Environment of Software Testing. The 123rd ASEE Annual Conference & Exposition. - Computing and Information Technologies (CIT) , June 2016. Paper ID: 15359.
- [4] Yujian Fu, Nelson Barnes and Peter J. Clarke. Integrating Software Testing into Computer Science Curriculum Using WReSTT-CyLE. The 123rd ASEE Annual Conference & Exposition. - Computing and Information Technologies (CIT) , June 2016. Paper Id: 15885.
- [5] STEM-CyLE. From: <https://stem-cyle.cis.fiu.edu/instances>. Dec 18, 2018.
- [6] Exforsys. The History of Object-Oriented Programming, from: <http://www.exforsys.com/tutorials/oops-concepts/the-history-of-object-oriented-programming.html> September 13, 2006.
- [7] A PBL-Based, Integrated Learning Experience of Object-Oriented Programming, Data Structures and Software Design. In proceedings of the 48th Annual Frontiers In Education Conference, 2018, San Jose, California, USA. July 2018.
- [8] Project based learning. Educators of America. Dec 17, 2018, from: <https://educatorsusa.org/our-programs/professional-development/>.
- [9] Eugenia Fernandez and David M. Williamson. Using Project-based learning to Teach Object Oriented Application Development. In Proceedings of the 4th Conference on Information Technology Curriculum. Pp 37-40. Lafayette, Indiana, USA. October 16-18, 2003.
- [10] Daniel Ewert, Daniel Schilberg, Sabina Jeschke. Problem based Learning of Object-oriented Programming with LEGO Mindstorms and leJOS. In Proceedings of Automation, Communication and Cybernetics in Science and Engineering 2011/2012. Springer, Berlin, Heidelberg. Pp 315-323.
- [11] Hong Huang. The Incremental Teaching Project Design for Project-based Learning and Its Application in Java Programming Course. Science Journal of Education. Vol. 4, Issue 6, December 2016, pages: 191-197.
- [12] Object-Oriented Programming. Microservices. August 2008, from: <https://searchmicroservices.techtarget.com/definition/object-oriented-programming-OOP>.
- [13] A Brief History of Object-Oriented Programming, from: <http://web.eecs.utk.edu/~huangj/CS302S04/notes/oo-intro.html>.

- [14] Roger S Pressman, “Software Engineering: A Practitioner's Approach,” 7th edition, McGraw-Hill Science, ISBN-13: 978-007337597-7.
- [15] UML Class and Object Diagrams Overview, from: <https://www.uml-diagrams.org/class-diagrams-overview.html>.
- [16] Activity Diagrams, from: <https://www.uml-diagrams.org/activity-diagrams.html>.
- [17] UML Sequence Diagrams, from: <https://www.uml-diagrams.org/sequence-diagrams.html>.
- [18] UML Use Case Diagrams, from: <https://www.uml-diagrams.org/use-case-diagrams.html>.
- [19] UML Association, from: <https://www.uml-diagrams.org/association.html>.
- [20] Aggregation in Java: Definition and Examples. Thought Co., from: <https://www.thoughtco.com/aggregation-2033995>.
- [21] JUnit – Using Assertion, from: https://www.tutorialspoint.com/junit/junit_using_assertion.htm.
- [22] JUnit Tutorial: Setting Up, Writing, and Running Java Unit Tests. Parasoft. August 2, 2018, from: <https://blog.parasoft.com/junit-tutorial-setting-up-writing-and-running-java-unit-tests>.
- [23] Unit Testing with JUnit – Tutorial, from: <http://www.vogella.com/tutorials/JUnit/article.html>.
- [24] Khamlichi. The History of Android, from: <https://www.unixmen.com/the-history-of-android/> August 31, 2015.
- [25] Margaret R.E., Vinod A. M., Tejonidhi M. R. 2018. A Project Based Approach to Teach Object Oriented Modeling and Design Patterns, Journal of Engineering Education Transformations, Special Issue, eISSN 2394-1707.
- [26] Frank Alexander Kraemer et al. Engineering android applications based on UML activities. In the 14th international conference on Model driven engineering languages and systems (MODELS), 2011 Jon Whittle, Tony Clark, and Thomas Kühne (Eds.). Springer-Verlag, Berlin, Heidelberg, pages 183–197, 2011.
- [27] A.G. Parada and L.B. de Brisolará. A model driven approach for android applications development. In Brazilian Symposium on Computing System Engineering (SBESC), pages 192–197, Nov 2012.
- [28] Bup-Ki Min, Minhyuk Ko, Yongjin Seo, Seunghak Kuk, and Hyeon Soo Kim. A UML metamodel for smart device application modeling based on windows phone 7 platform. In TENCON 2011-2011 IEEE Region 10 Conference, pages 201–205. IEEE, 2011.

- [29] Hyun Seung Son, Woo Yeol Kim, Jae Seung Kim, and Robert YoungChul Kim. Concretization of uml models based on model transformation for windows phone application. *Information Science and Technology (IST)*, pages 288–291, 2012.
- [30] Nurul Azmi, Dwi Maryono, Rosihan Ari Yuana. Development of an Android-based Learning Media Application for Visually Impaired Students. *Indonesian Journal of Informatics Education*. Vol. 1, No. 1, pp. 61-68, June 2017.
- [31] N. Nurwahidah, Insih Wilujeng, J. Jumadi, Senam. The Effects of Project Based Learning Model with Android on Scientific Literacy and Digital Equipedness/ICT Literacy. *International Journal of Sciences: Basic and Applied Research*. Vol. 36, No. 7, 2017. pp. 190-205. ISSN 2307-4531.
- [32] Mourya Reddy Narasareddygari, Gursimran S. Walia, Debra M. Duke, Vijayalakshmi Ramasamy, James Kiper, Debra Davis, Andrew Allen, and Geoffrey Potvin. Evaluating the Impact of Combination of Engagement Strategies in SEP - CyLE on Improve Student Learning of Programming Concepts, The 50th ACM Technical Symposium on Computer Science Education, SIGCSE 2019. February 27 - March 2, 2019, Minneapolis, MN, USA.
- [33] Ingrid A. Buckley and Peter J. Clarke. An Approach to Teaching Software Testing Supported by two Different Online Content Delivery Methods. 2018 LACCEI International Multi-Conference of Engineering, Education and Technology. Accepted 04/11/18.
- [34] Vijayalakshmi Ramasamy, Urvashi Desai, Hakam W. Alomari, and James D. Kiper. TP-GraphMiner: A Clustering Framework for Task-Based Information Networks. *International Conference on Systems, Computation, Automation, Networking (ICSCAN '18)*, Manakula Vinayagar Institute of Technology, Kalitheerthalkuppam, Puducherry, India, July 2018.
- [35] Mourya Reddy Narasareddygari, Gursimran S. Walia, and Alex Radermacher. Using Gamification and Cyber Learning Environment to Improve Students' Learning in an Introductory Computer Programming Course: An Empirical Case Study. *The 125th ASEE Annual Conference & Exposition. – Computers in Education Division*, June 2018. Paper ID: 22813
- [36] Mourya Reddy Narasareddygari, Gursimran S. Walia, and Alex Radermacher. Gamification in Computer Science Education: a Systematic Literature Review. *The 125th ASEE Annual Conference & Exposition. – Computers in Education Division*, June 2018. Paper ID: 22808
- [37] Vijayalakshmi Ramasamy, Hakam W. Alomari, James D Kiper, and Geoffrey Potvin. A Minimally Disruptive Approach of Integrating Testing into Computer Programming Courses. In *The 2nd IEEE/ACM International Workshop of Software Engineering Education for Millennials (SEEM'18)* (p. 8). 2 June 2018, Gothenburg, Sweden.