PROJECT **3.1.10**

Social Networking App - Development

INTRODUCTION

Now that you have created a prototype for your social networking app, it is time to use the knowledge and skills that you have acquired thus far to develop it! You will build and test your app following the Agile software development methodology. As you build your app, you will:

- Consider how user interfaces can impact the programmer and the end-user experience.
- Relate the Android ecosystem to the model-view-presenter pattern.
- Document your work with the appropriate tools.
- Collaborate effectively with your partner.

Materials

- Computer with Android™ Studio
- Android™ tablet and USB cable, or a device emulator
- Free Backendless account per student
- Tools with which to create Unified Modeling Language (UML) class diagrams

RESOURCES

Project 3.1.10 Requirements
Resources available online

Lesson 3.1 Reference Card for Backendless
Resources available online

Procedure

1. Continue to work with your partner following the pair programming model, making sure to switch roles as directed by your instructor.
2. Recall the Agile software development process described in *3.1.9 Social Networking App Design*. In this project, you will be working on the Code and Test cycle (or *iteration execution*).

3. Review your sprint task list and refer back to your prototype.
4. Do iterative execution:
    a. Identify all your entities, and as directed by your instructor, use a tool to draw a UML class diagram to illustrate what the attributes and possible methods will be for each entity.
    b. Begin coding your first sprint, working on the tasks identified in your task list.
        - Make sure to comment your code as necessary.
        - Determine whether certain code needs to be original, reused, or found in a pre-defined library.
        - Determine what new classes you will need and what classes or interfaces this class should extend or implement.
        - Maintain encapsulation when designing your classes.
        - Decide which layer of the model-view-presenter pattern each new class belongs to and avoid mixing functionality.
    c. Test your code frequently and use any of the troubleshooting methods you have learned in this course to isolate bugs and fix them.
    d. As you work on the code-test cycle, you might find the need to go back and strategize again. Update your task list appropriately as you iterate through this cycle.
5. Write your iteration retrospective, by answering the following two questions:
    a. What did you do well as a team? Identify successes that can be repeated in the future.
    b. What did you NOT do well and how should you address that in the future? Identify failures and lessons learned.
6. If you complete the requirements identified in your first sprint, check with your teacher to see whether you have time to begin your next sprint to incorporate some of the optional features.
7. Submit and present your project as directed by your instructor.

CONCLUSION

1. What part of the project do you feel was most challenging? Explain the lessons you learned from this experience.
2. What were the dynamics of your teamwork? What worked well? What can be improved? Write a reflection and discuss with your partner.

Problem 3.1.10 Requirements

Product Requirements List

1. Problem Statement
2. Product Backlog
3. Sprint Task List
4. Prototype of the first sprint task
5. Source code for the product
6. Demonstrated functionality of at least:
    a. The Backendless back end for data storage
    b. Android activities and fragments, where applicable
    c. User authentication
    d. Users able to create textual posts
    e. One more feature of your choice
7. Documentation of any features that the team added that go above and beyond