PROJECT **3.2.4**

Location Awareness - Development

INTRODUCTION

Now that you have designed your social networking app to be location aware, it is time to start the development iteration. In this project, you will work on your task list following the Agile software development methodology, while making sure to document your work and collaborate effectively with your partner.

Materials

- Computer with Android™ Studio
- Android™ tablet and USB cable, or a device emulator
- Free Backendless account per student
- A Google account and a project on the Google API Console
- Tools with which to create UML class diagrams

RESOURCES

Lesson 3.2 Reference Card for Google Play Services
Resources available online

Project 3.2.4 Requirements
Resources available online

Procedure

Continue to work with your partner following the pair programming model; make sure to switch roles as directed by your instructor.

1. Recall the Agile software development process you used in previous projects. In this project, you will work in the Code and Test cycle, also referred to as Iteration Execution. For a review of Iteration Execution, see the slideshow presentations in Activity 2.2.6.
2. Review your sprint task list and refer back to your prototype.

3 Iteration Execution:

a. Identify all your entities. As directed by your instructor, use a tool to draw a UML class diagram that illustrates the attributes and possible methods for each entity.

b. Begin coding your first sprint, working on the tasks identified in your task list. Make sure to:

- □ Add Google Play services to the project.
- □ Add all the appropriate permissions and features in the Android™ Manifest file.
- □ Create the appropriate Google Maps API Key.
- □ Comment your code as necessary.
- □ Determine whether certain code needs to be original, reused, or found in a predefined library.
- □ Determine what new classes you will need and what class or interfaces each 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.

4 To write your iteration retrospective, answer 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.

5 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.

6 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 3.2.4 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 the following:
    a. The app must use Google Play services.
    b. The app must use a map.
    c. Location detection must be used in at least one screen.
7. Documentation of security issues related to third-party software
8. Documentation of any features that the team added that go above and beyond