1.9 - Unit Assessment - Restaurant App v0.1

OVERVIEW

Students will design and code a Restaurant App (for a real existent restaurant or for a made-up one of their choosing) that uses a *Button, TextView, EditText,* and one other UI element researched by the student in the Android Support Libraries. Apps must have two activities, must include at least two user interactions, and should showcase elements of good design.

Estimated Duration: 3 days

STANDARDS ADDRESSED

CSTA Standards

- 3B-AP-16: Demonstrate code reuse by creating programming solutions using libraries and APIs.
- 3B-AP-17: Plan and develop programs for broad audiences using a software life cycle process.
- 3B-AP-19: Develop programs for multiple computing platforms.
- 3B-AP-24: Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems.
- 3A-AP-16: Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
- 3A-AP-21: Evaluate and refine computational artifacts to make them more usable and accessible.
- 3A-AP-23: Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

Maryland Computer Science Standards

- 12.AP.PD.03: Develop different programs for various computing platforms (e.g., desktop, web, mobile).
- 12. AP.PD.07: Compare multiple programming languages or libraries and discuss how their features make them suitable for solving different types of problems
- 12. AP.M.03: Create programming solutions using libraries and APIs through the application of code reuse.
- 10. AP.PD.01: Systematically design and implement programs for broad audiences, solicit user feedback, and refine programs based on user feedback.

- 10. AP.PD.03: Evaluate and refine computational artifacts to improve usability, accessibility, and efficiency.
- 10. AP.PD.05: Represent the design elements and data flow (e.g., flowcharts, pseudocode, etc.) of the development of a complex program through the use of various visual aids and documentation techniques
- 10. AP.C.02: Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions

UNDERSTANDINGS

Factual Knowledge	Procedural Knowledge	Conceptual Knowledge
Students will know: • The parts of Android Studio IDE • Basic Android terminology • Basic Android UI elements, their names and functions • The Android App Lifecycle	*Students will be able to:* • Write XML code for various UI elements within a LinearLayout • Inflate and interact with those UI elements using basic Java code • Use variables to store and track data • Declare an intent and pass information to a new Activity • Deploy and test out their first apps	*Students will understand:* • How XML, Java, design & planning all contribute to a good app • The process of creating an app in Android Studio

ESSENTIAL QUESTION(S)

- How do Android apps get designed and made?
- What makes an app design “good”?
- How do we balance aesthetics and functionality in app design?
- How do user interfaces impact the experience of both the programmer and the end user?
- If I get errors while coding, how can I solve them?

OBJECTIVE

Students will demonstrate their understanding of major unit facts, procedures, and concepts by designing and coding a Restaurant App.

ASSESSMENT / PERFORMANCE TASK

Students will design and code a Restaurant App that uses a *Button*, *TextView, EditText*, and one other UI element researched by the student in the Android Support Libraries.
Completed apps:

- must have two activities,
- must include at least two user interactions,
- and should showcase elements of good design.
- Will be assessed according to the Restaurant App v0.1 Rubric

MATERIALS NEEDED

The basic course materials (computer, Android device, USB cable, internet access, screen projection/sharing system) are needed every day. Additionally, today you should make sure the following links all work and are accessible to students:

Rubric

- A copy of the Restaurant App v0.1 Rubric for each student

For the teacher:

- Completed Restaurant App v0.1 Sample Code on GitHub
    - This contains more than the required two user interactions to showcase some possible variety. The aim is for students to be creative in what they pick, both in terms of UI elements and interactions.
- App Development 1 & 2 GitHub Guide

LEARNING PLAN

DAY ONE

	ACTIVITIES	SUGGESTED TIMING
1.9.1	Warm Up: Restaurant App Brainstorm	10 minutes
1.9.2	Restaurant App Project Overview	5 minutes
	Independent Work Time	40 minutes
	Exit Ticket (or Day 2 Warm Up): Progress Check	5 minutes

DAY TWO

	ACTIVITIES	SUGGESTED TIMING
	Warm Up: Progress Check (skip if done as Exit Ticket on Day 1)	5 minutes

	Independent Work Time	50 minutes
	Exit Ticket (or Day 3 Warm Up): Progress Check	5minutes

DAY THREE

	ACTIVITIES	**SUGGESTED TIMING**
	Warm Up: Progress Check (skip if done as Exit Ticket on Day 2)	5 minutes
	Independent Work Time; App Submission	40 minutes
	App Demos	15 minutes

ACTIVITY 1.9.1 - WARM UP (Day 1)

1. Ask students to think about how they would create an app for a restaurant (perhaps even draw a sketch, or keep it more theoretical like what features would it have).
    - **NOTE**: Because we will be coming back to this project in Unit 2, it might be good to brainstorm a whole lot now and then approach it as "well, let's tackle the parts we've learned how to do now first" and that might motivate students to want to learn the topics in unit 2 so they can implement their other ideas.
2. Invite students to share their responses. As students discuss what belongs in a restaurant app, recognize all ideas as awesome but highlight simple ideas that can be done with what we have learned so far.
    - For example, the following are too complex for now: ordering and payment (we'll get to that in Restaurant v0.2), a full menu (though perhaps showing one or two menu items with picture and price could work). Instead, focus on visual design/layout, info that is relevant to a restaurant app user (e.g. picture, address, hours), and what basic interactivity can go into that.

ACTIVITY 1.9.2 - RESTAURANT APP PROJECT OVERVIEW

1. Provide an overview of the Restaurant App Project to students.
2. State key requirements of version 0.1 of the project, which will be the end-of-unit summative assessment:
    - 4 UI elements,
    - 2 activities,
    - 2 user interactions,
    - good aesthetic and functional design

3. Provide students with a copy of the Restaurant App v0.1 Rubric and ask them to review. Respond to any questions that arise.
4. Try to encourage student creativity within defined grading criteria (e.g. interactions can be anything, visual creativity, what their second *Activity* will be).

INDEPENDENT WORK TIME (Days 1-3)

1. Students work primarily independently coding their Restaurant App v0.1.
2. Provide help / encourage peer tutoring as you see fit.
3. If you taught basic git in Lesson 1.8:
    a. Remind students to set up their repo at the beginning of this project, right after drawing their sketch of the app wireframe/prototype.
    b. Then remind students daily to push to GitHub at the end of each class period if not more frequently.
    c. Code can be submitted by sharing a link to their GitHub repository.

DAILY EXIT TICKET (Days 1 + 2) -OR- WARM UP (Days 2 +3)

Check in with students to gauge their progress on the project and / or ask questions about the rubric to ensure students understand.

APP DEMOS (Day 3)

Each student takes a turn to show off their app for their restaurant, demonstrating how the user can interact with it, and highlighting any unusual or unique features.