

Final Project and Presentation

Grading Template

About

This is a template; Do not edit directly. Make a copy of this template for each group project, and fill it out during presentations.

For Students: I will use this sheet to grade your projects while you present. You can improve your score, improve the accuracy of my grading, and reduce your presentation time by trying to match your presentation to each rubric item below exactly, in order.

Project Information

Project Name	TranslateMe App
Date	12/16/2022
Time Slot	4:05pm

Group Information

Students who do not present some aspect will receive a deduction. Students who no-show without a provable emergency receive a 0.

<i>Student Name / Info</i>	<i>Grad?</i>	<i>Showed Up?</i>	<i>Presented?</i>
Terrell D Lemons	N/A	✓	✓
Zhiqiang Liu	N/A	✓	✓
Tommy Vu	N/A		
Vong	N/A		

Requirements

<i>Requirement / Notes</i>	<i>Demo'ed?</i>	<i>Showed Code?</i>	<i>Effort (1 - 10)</i>
<i>Fresh Clone</i> Student(s) performed a fresh clone of their project before presenting.	✓	✓	
<i>Callbacks</i> User action executes a callback somewhere	✓	✓	
<i>Logging</i> User action or business logic causes emission to Logcat.	✓	✓	
<i>Layouts</i> Use of ConstraintLayout and one other container/layout type.	✓	✓	
<i>Resources – Two Images</i> At least two image resources found in app	✓	✓	
<i>Resources - Interactivity</i> Two resources change in some way due to user interaction.	✓	✓	
<i>Resources (extra credit)</i> Embedded a playable video or audio resource, somewhere.	✓	✓	
<i>Resource Qualifiers - Translations</i> App is translated to multiple languages.	✓	✓	
<i>Resource Qualifiers - Layouts</i> App has multiple layouts that get used via qualifiers.	✓	✓	
<i>Resource Qualifiers - Wildcard</i> App has a third type of resource that gets selected via qualifiers (not translations or layouts).	✓	✓	
<i>Persistence – Rotation Survival (ViewModel)</i>			

Requirement / Notes	Demo'ed?	Showed Code?	Effort (1 - 10)
App has data that survives rotation via a ViewModel.	✓	✓	
Persistence – App Exit Survival App has data that survives a complete app exit.	✓	✓	
Persistence – Database App allows user to enter data that can be saved to a local SQLite database, and fetched later. Data survives app exit. User is able to save/fetch multiple entries for the same type of data (i.e., multiple rows in a database table). This requirement may be tweaked slightly the week before presentations, if needed.	✓	✓	
Persistence – Preferences w/ DataStore or Firestore App let's user set and get preferences using the DataStore package. Preferences survive app exit. Alternatively, Firestore may be used.	✓	✓	
UI Flexibility – Fragments Present App has multiple fragments, each with their own controllers and layouts.	✓	✓	
UI Flexibility – Fragments in Layout App has at least one layout that contains multiple fragments at the same time.	✓	✓	
UI Flexibility – Fragment Reuse Fragments in app are reused in multiple layouts.	✓	✓	
RESTful GET App is able to fetch data from a remote REST endpoint using the Jetpack library, via a GET request.	✓	✓	
RESTful POST			

<i>Requirement / Notes</i>	<i>Demo'ed?</i>	<i>Showed Code?</i>	<i>Effort (1 - 10)</i>
App is able to send data to a remote REST endpoint using the Jetpack library, via a POST request. Data sent to server comes from user input. Student confirmed the sent data can be fetched after an app exit. (i.e., sent data survives app restart).	✓	✓	

Graduate Requirements

Make note of each graduate student in the group and the grad requirement they present.

<i>Grad Student</i>	<i>Description</i>	<i>Effort (1 - 10)</i>

✓

✓

Overall

<i>Creative / Useful (1 - 10)</i>	
<i>Completeness (1 - 10)</i>	
<i>Misc Notes</i>	