COMP2140/7240 Web/Mobile Programming

THE UNIVERSITY OF QUEENSLAND AUSTRALIA | CREATE CHANGE

React Native Assessment Item

The **React Native Assessment Item** has a **20%** weighting. Important points to note:

- You will also need to do a **Project Demonstration and Code Review** for your **React Native Project** worth an additional **12%** weighting in Week 13.
- You can choose between doing the Standard project or a Custom project. To do a Custom project please email (COMP2140@eecs.uq.edu.au) with your project idea by Weds 8 Oct 2025. The Teaching Team has incorporated opportunities for students to pursue Custom projects, allowing those with existing skills or ideas to utilize this course as to explore and implement their ideas. Please note that there are specific elements that must be included in your custom project.
- The React Natie project is intended to evaluate your proficiency in designing and developing mobile applications built in React Native and Expo. You are given access to a RESTful API for saving and retrieving data.
- **Table 1** provides an overview of the Standard project and guidelines for getting approval for a Custom project.
- **Table 2** provides the detailed requirements/rubric for the Standard React Native Project. Please make sure you carefully read them and feel free to ask questions.
- **You must come up with your own UI design for your project.** You are allowed to use CSS Framework and Component Libraries for React Native such as NativeWind, React Native Reusables, GlueStackUI, React Native Paper or BNA UI. No paid components or templates are allowed.
- You can use open source React Native/Expo Components. You can use other components if the components are not high level form building or API calling ones that won't allow you to learn the fundamentals of React Native. You are encouraged to explore and incorporate other React components but will need to read the documentation yourself.
- **Expo File-Based Routing must be used.**
- You are not building a full SaaS application – No login is required. For the Standard Project, an API to store and retrieve your data is provided. See the Getting Started Guide slide deck for the API endpoint.
- Reading documentation for React Native, Expo Libraries, Component Libraries and PostgREST queries is required for this assessment item. Lectures and Applied Classes cover the essentials, but you will need to look up documentation.
- You can use Typescript if you know how program with it.

Table 1: Available Project Options

Project Name & Description	Main Features
FormBase *A Mobile App that builds custom forms for data collection* FormBase is a prototype mobile app built with Expo and React Native that lets users create flexible data-collection forms on the fly. Each form can include a mix of field types — text, multiline text, dropdowns, image/camera and location (i.e. capture the users location) with options for marking fields as required or numeric. Once a form with fields is created, users can save records. Users can define custom filter (including numeric comparisons and string comparisons), and visualize location results on a map. The prototype demonstrates how lightweight, mobile-first tools can empower users to design, collect, and explore structured data without needing a custom backend for every use case but lacks features for editing/updating fields.	Allows users to: • Add / Edit Forms • Add Fields to a Form (i.e. text, multiline, dropdown, location and image or camera) • Mark fields as required or numeric • Enter data and save as Records • Search and filter Records with criteria (text and numeric operators) • View Records in a list • Copy or Delete Records • Visualize location-based Records on a map
Custom Project Guidelines • You must submit your project idea proposal to COMP2140@eecs.uq.edu.au by Wed 8 Oct 2025. • You will need to provide your own API to store and retrieve your data. You can use external services or deploy an API to your own UQ Cloud Zone. The Teaching Team have used PostgREST (https://docs.postgrest.org/en/v12/) for the API and a self-paced tutorial is provided, if you also wish to take this approach. With PostgREST you only need to setup a database and then enable the RESTFul API. • Custom projects must use React Native/Expo, use Device API's (i.e. Location, Media capture [image, photo, audio or video] and a Map) and include a Filter Criteria Builder or equivalent component in terms of complexity	

Code Submission:

You must submit a single **zip file** named 's1234567_Firstname_Lastname_ReactNative.zip' (replace with your student number) that includes your **source code folder(s)** and a Readme.md. The Readme.md file should include:

- Your GenAI use reference. This should be a statement of where/how you used have GenAI (e.g. ChatGPT or Claude or Gemini)
- The Emulator and Device you tested your app (Expo Go app) on

If you have developed a custom project, please ensure that API keys are provided, so that the marker is able to test your code. Please don't include node-modules when you submit.

Additional Questions:

If you have any questions about this assessment brief, you're welcome to post them on the course Ed Discussion and we'll get back to you soon.

A Message About Plagiarism:

⚠ **Plagiarism is considered a serious offence at UQ.** Failure to declare the distinction between your work and the work of others will result in academic misconduct proceedings.

- The use of Generative AI (i.e. ChatGPT, Claude, Google Gemini, Microsoft Chat and Github Copilot, Claude Code, Cursor, Codex etc) is allowed for this assessment item to assist you in designing your web application and learning new concepts. However, treat what you're producing here as a "trade secret" and don't share your code with other students. Also include details of where Generative AI has been used in a Readme.md file or withing your code comments.
- If you're inspired by design or code from online tutorials or any other external source, ensure you reference any inspirations for academic purposes (using APA/IEEE referencing styles) in an Readme.md file.

Standard Project Detailed Specification

Table 2 contains the detailed specification for each required screen in the Standard Project. Note: Screens/Functionality are listed below however you don’t need to stick to the screen structure as you have flexibility with your design.

Table 2: Detailed Specifications

Screen/Functionality	Key Required Features
App design and Navigation	• Must support global navigation to About and Home/Welcome Screens • Must support display of a Form and sub-screens within a Form (e.g. Form Home, Records List and Map Screen) • A combination of a slide out panel and tabs can be used, but you can design your own navigation
My Forms	• Add/Edit and List Forms (/form resource in the RESTFul API) • Main fields: Name, and Description. All fields are required.
Specific Form Creation and Data Entry	• /field resource in the RESTFul API and relates to a form • Ability to Add fields (text, multiline, dropdown, location and image [or camera]) and select if the field is required and is_num. For a dropdown an additional comma separate list must be entered and save to the options jsonb field. Location is a field that captures the users current location in longitude and latitude. Either and image picker or camera field can be implemented. • Main fields: form_id, name, field_type, options, required, is_num • **Note: Editing/Deletion/Ordering of Fields is not required for the assessment. It is assumed that fields will not be added after data is entered. This means that you don’t need to implement or consider form field changes in your app.** • After fields are added, allows data entry and saving of the record. • Fields are then rendered using the appropriate control (i.e. text input, dropdown, location picker, and image/camera control) for the form. • Once a form is saved, the record is saved to the /record resource in the RESTful api. The json needs to be save to the values field. Field names should be included as keys in the saved json.

	• Validation must be applied for required and is_num fields. Location data should be stored as json for long and lat. The Image should only store the file path of the selected image back via the RESTFul API. The location data and selected image should be displayed on the form.
Records	• List of Records (displays all details for a record) for the current Form with each field displayed including the json of the Location and Image. • Ability to Delete a record and Copy the content to clipboard (as json). • **Note: Editing of records is not required.** • The Records Screen should include a Filter Criteria Builder. The Builder should allow fields, and operator (e.g contains) and value to be added as criteria. The search builder must present numeric operators (equals, greater than, less than, greater or equal, less or equal) when the selected field is marked as numeric, and string operators (equals, contains, startswith) if the field contains text. Multiple criteria should be able to be added with the user able to select between And and Or logic. No complex grouping of and/or clauses with parenthesis is required for the assessment. Applying the filter should display matching records.
Map	• Displays a Map with Markers positioned at the long and lat from the Location field for the current Form • The Map should only display if a form includes a location field • The Marker when clicked should display the Record details. Images and location (long and lat) should be displayed.