# React Native Project Assessment Rubric

The assessment will be marked out of 100 and then scaled to be out of 20. **Note:** COMP7240 students have an extra task as per the Course Profile.

**Table 3: Grading Rubric.**

| ID | Feature Description | COMP2140<br>Max Grade = 100 | COMP7240<br>Max Grade = 100 |
|---|---|---|---|
| **1** | **Core Functionality** | **55 marks** | **55 marks** |
| 1.1 | Package.json includes all dependencies, React Native/Expo code runs without errors, follows required folder naming convention and includes a examples with Form with all implemented Fields \| Custom that can be used in the Project Demonstration and Code Review. | 4 | 4 |
| 1.2 | Implements Expo File-based Routing, supports required application navigation and includes a Welcome/Home Screen and About Screen. | 6 | 6 |
| 1.3 | Includes the ability to list, add, edit and delete Forms \| Custom. No paging is required. Add/Edit forms should include all mandatory fields provided by the Form API endpoint (/form). Use best practice for loading data and having shared components for add/edit form. | 5 | 5 |
| 1.4 | Includes the ability to add fields to a form. Text, multiline, dropdown and location should be supported with the user able to select whether the fields is required or a number (i.e. is_num). For a dropdown an additional comma separate list must be stored within options jsonb. Location is a field that captures the user's current location in longitude and latitude.<br>Note: Edit and Delete for form fields are not required. | 10 | 10 |
| 1.5 | Includes the ability to render a form with all added fields for data entry using the appropriate component and ensuring validation for required and number fields. Text, multiline, dropdown and location must be supported. Location needs to capture the users longitude and latitude data. Location data should be stored as json for long and lat. Validation for required and number fields must be included with a message displayed indicating which field has not passed validation. | 15 | 15 |

| | | | |
|---|---|---|---|
| 1.6 | Includes a Record List Screen with the ability to list records with delete functionality. | 5 | 5 |
| 1.7 | The Records List Screen includes a Filter Criteria Builder. The Builder should allow fields, and operator (e.g contains) and value to be added as criteria. The Criteria 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. | 10 | 10 |
| **2.** | **Device API's** | **15 marks** | **15 marks** |
| 2.1 | Includes an image picker or camera field type. The selected image should reference the image file path and store this path back to the field in the /records API resource. The image should be displayed when records are displayed and within the Maps marker. | 7 | 5 |
| 2.2 | The Map Screen displays Markers positioned at the long and lat from the Location field. The Map should only display if a form includes a location field otherwise include a message saying that no location data is available. The Marker when clicked should display the Record details. | 5 | 5 |
| 2.3 | Includes the ability to copy a record from the Records List Screen to the clipboard. The copied data can be in json or pure text. Image support is not required. | 3 | 3 |
| **3.** | **User Interface and User Experience Design** | **15 marks** | **10 marks** |
| 3.1 | Uses a CSS Framework or Component Library or Custom CSS where appropriate. Includes a consistent look and feel across the application. | 5 | 5 |
| 3.2 | Navigation and workflow are intuitive, guiding users through each step in the process of setting up and filtering form data, with features that are easy to use and clear instructions provided where needed. | 10 | 5 |
| **4** | **Code Style and Quality**<br>(Note: Max Grade will be capped at 50% grade points if only 50% of the grade for Core Functionality is achieved) | **15 marks** | **10 marks** |
| 4.1 | Breaks down functionality into well-defined, re-usable components. Code duplication is minimized. JSX and Javascript/Typescript adheres to Functional paradigm (i.e. avoids embedded imperative logic). | 10 | 8 |

| | | | |
|---|---|---|---|
| 4.2 | Implements appropriate error handling for each component, ensuring robust error detection and management while displaying clear, user-friendly error messages that enhances user experience and facilitates troubleshooting. | 2.5 | 1 |
| 4.3 | Well formatted and documented code with inline comments included. | 2.5 | 1 |
| **5** | **Data Privacy (For COMP7240 Students – Research Question)** | | **10 marks** |
| 5.1 | When building mobile apps like FormBase \| Custom that collect user data, developers must consider how data is stored, transmitted, and protected. For this assessment, discuss briefly (in bullet point form) the privacy risks of capturing text, media, or location information through a mobile form app. | | 10 |