Department of
Computing and Mathematics
ASSIGNMENT COVER SHEET

Unit title:	Mobile Applicatons Development
Assignment set by:	Ashley Williams
Assignment ID:	2CWK50
Assignment title:	Implement the Chittr API as a React Native application
Assessment weighting:	50%
Type: (Group/Individual)	Individual
Hand-in deadline:	20th March 2020
Hand-in format and mechanism:	Via Moodle

Learning outcomes being assessed:

LO1: Develop and use a variety of advanced mobile applications and location aware mobile development technologies, operating systems and environments.

LO3: Create and assess intelligent location aware mobile applications

LO4: Research and demonstrate knowledge and practical application of current and novel mobile device techniques

Note: it is your responsibility to make sure that your work is complete and available for marking by the deadline. Make sure that you have followed the submission instructions carefully, and your work is submitted in the correct format, using the correct hand-in mechanism (e.g. Moodle upload). If submitting via Moodle, you are advised to check your work after upload, to make sure it has uploaded properly. Do not alter your work after the deadline. You should make at least one full backup copy of your work.

Penalties for late hand-in: see Regulations for Undergraduate Programmes of Study (http://www.mmu.ac.uk/academic/casqe/regulations/assessment.php). The timeliness of submissions is strictly monitored and enforced.
All coursework has a late submission window of 5 working days, but any work submitted within the late window will be capped at 40%, unless you have an agreed extension. Work submitted after the 5-day window will be capped at zero, unless you have an agreed extension.
Please note that individual tutors are unable to grant extensions to coursework.

Exceptional Factors affecting your performance: see Regulations for Undergraduate Programmes of Study (http://www.mmu.ac.uk/academic/casqe/regulations/assessment/docs/ug-regs.pdf). For advice relating to exceptional factors, please see the following website: https://www2.mmu.ac.uk/student-case-management/guidance-for-students/exceptional-factors/ or visit a Student Hub for more information.

Plagiarism: Plagiarism is the unacknowledged representation of another person's work, or use of their ideas, as one's own. Manchester Metropolitan University takes care to detect plagiarism, employs plagiarism detection software, and imposes severe penalties, as outlined in the Student Handbook (http://www.mmu.ac.uk/academic/casqe/regulations/docs/policies_regulations.pdf and Regulations for Undergraduate Programmes (http://www.mmu.ac.uk/academic/casqe/regulations/assessment.php) . Bad referencing or submitting the wrong assignment may still be treated as plagiarism. If in doubt, seek advice from your tutor.

As part of a plagiarism check, you may be asked to attend a meeting with the Unit Leader, or another member of the unit delivery team, where you will be asked to explain your work (e.g. explain the code in a programming assignment). If you are called to one of these meetings, it is very important that you attend.

Assessment Criteria:	Indicated in the attached assignment specification.
Formative Feedback:	Lecture/Lab discussion and interactive with tutor onwards from when the assignment is set.
Summative Feedback Format:	You will be given individual feedback via Moodle, as well as common feedback for all the class.

Mobile Applications Development

Assignment 2 – Implement the Chittr API as a React Native application

1. Introduction

This assessment is coursework based, and worth 50% of the overall unit mark. The tasks that you are required to complete for this assessment are outlined in this coursework specification.

2. Aim

This unit encourages you to analyse real world situations critically. The assessment mimics industry projects by requiring you to engage with multiple disciplines. By the end of the unit, you will have completed the development of a mobile application that uses a variety of advanced mobile application technologies. It is encouraged that you maintain a portfolio of projects throughout university (e.g. through GitHub) that can serve as a portfolio of your work when applying for jobs. This project could serve as one aspect of your portfolio.

The following skills will be essential for successful completion of this coursework (and including such a project in your portfolio would demonstrate these skills to potential employers):

- Real world problem solving: You will need to analyse a real-world situation, develop solutions for multiple problems when developing the application, and then evaluate your solutions.
- Technical skills: This assessment requires you to write an application in JavaScript using the React Native framework. You will then export your code into an Android application. In addition to these technologies, you will gain an understanding of RESTful APIs and the OpenAPI specification. The unit will also provide you with some experience in interacting with applications developed using NodeJS, ExpressJS, and MySQL.
- Modern relevant JavaScript frameworks: From their website – "Facebook released React Native in 2015 and has been maintaining it ever since. In 2018, React Native had the second highest number of contributors for any repository in GitHub. Today, React Native is supported by contributions from individuals and companies around the world including Callstack, Expo, Infinite Red, Microsoft, and Software Mansion. Our community is always shipping exciting new projects and exploring platforms beyond Android and iOS with repos like React Native Windows and React Native Web."

2.2 Assessment Learning Outcomes

Learning Outcome 1: Develop and use a variety of advanced mobile applications and location aware mobile development technologies, operating systems and environments.

Learning Outcome 3: Create and assess intelligent location aware mobile applications

Learning Outcome 4: Research and demonstrate knowledge and practical application of current and novel mobile device techniques

3. Coursework Overview

To complete this assessment, you are required to develop a mobile application. The precise detail of the coursework task are detailed in section four below. However, to summarise, you will be developing an application that can interface with a microblogging API (see

https://www.lexico.com/en/definition/microblogging). You are required to write the application in React Native and compile the code into an Android application.

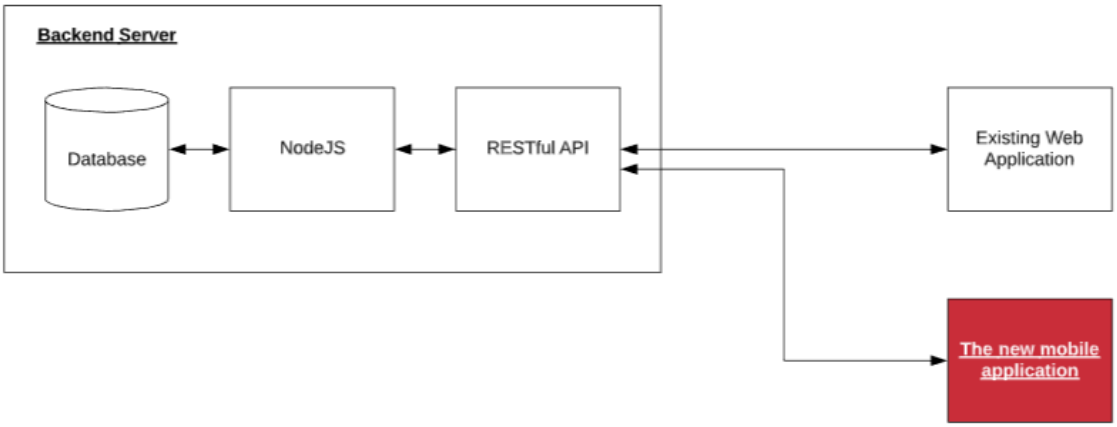
4. The Assessment (2CWK50)

4.1 Scenario

Chittr is a totally original, unique and non-plagiarised platform for microblogging. Users who sign up for an account can publish 'Chits' – short, textual based posts of no more than 141 characters. Users can also follow their friends and peers to keep updated with what their friends are Chitting about.

The Chittr team have an existing Web application that interacts with their back-end RESTful API. However, a decision has been made by the Chittr company directors to enhance usability through the development of a mobile front end. Chittr has hired you to develop a mobile application for them.

The Chittr team are dictating that the application is to be developed using React Native with a focus on initially delivering an Android project. The plan is that future iterations of the project will look towards adapting your codebase and rolling out an iOS equivalent.



4.2 Getting Started

It is recommended that you first complete labs 1-4 before attempting this assignment. The labs will show you how to create a React Native application, how to run the application on an Android device, how to structure applications in React Native, and how to interact with RESTful API's.

4.2.1 API Specification

The latest version of the API specification has been made available on Moodle. Lab 5 and lecture 5 will help you interpret the API specification using Swagger.io

4.2.2 Running the server

You have been provided with a copy of the backend server (on Moodle) and API for development. Lab 5 will show you how to download the server, configure it to point to your Mudfoot instance, run the server, and interact with the server.

4.3 Specific Tasks and Recommended Order

You will be assessed based on your application's coverage of the entire API. In addition to this, marks will be awarded for code quality/style and your application's usability. The order in which you

complete tasks is up to you. However, it is recommended that you develop your application in the following order:

1. Design your application's layout and structure (sketch wireframes to help you with this)
2. Create a new application using the reference guide for help
3. Implement your application's navigation structure using your wireframes
4. Implement the GET /chits end point to ensure that you can successfully query the API and present the results within your application
5. Implement the POST /user end point so that new users can sign up for accounts
6. Implement the login and logout end points
7. Implement the remaining 'User Management' end points except for anything that requires working with images
8. Implement the remaining 'Posting Chits' end points except for anything that requires working with images (at this stage, do not implement the geolocation tagging when creating new Chits).
9. Implement the 'Follower management' end points
10. Implement all end points that require working with images
11. Edit the POST /chits end point to include the option for a user to tag their current location.

4.3.1 Extension tasks

Extension task 1: Alter your application so that users can save local drafts of Chits before sending them to the API. You will need to save these drafts to permanent storage within the mobile device and have functionality to view, edit and delete these drafts (much in the same way that mail clients work).

Extension task 2: Alter your solution to Extension task 1 by allowing users to schedule when a draft Chit is posted.

4.4 Additional Guidelines

All of the below will be assessed as part of the assignment.

4.4.1 Wireframing

Wireframes should be created to initially sketch out your application's design, flow and layout. Iterate through multiple designs and then make a decision on which one you believe is better. Justify this decision in your submitted evidence document (see section 4.5 for more detail on submission elements).

Read more: https://www.usability.gov/how-to-and-tools/methods/wireframing.html

4.4.2 README

Your application should include a README file. The README file should contain relevant meta-data as well as detailed instructions on how to build and run your application (including any caveats/necessary workarounds).

Read more: https://www.makeareadme.com/

4.4.3 Version Control

It is encouraged that you use existing and recognised version control methods for managing your project. Marks will be awarded to those who can evidence that they have used version control software consistently (and appropriately) from the start of their project.

Read more: See "Helper Lab 3: Introduction to Git"

4.4.4 Code Quality

It is vital that you consider code quality from the start of your project. Use of consistent style and detailed comments will be assessed. You should make use of one of the many JavaScript style guides available on the internet. For example:

1. AirBnB (1,773 commits from 424 contributors): https://github.com/airbnb/javascript
2. Google: https://google.github.io/styleguide/jsguide.html
3. JS Standard (1,632 commits from 131 contributors): https://github.com/standard/standard

4.4.5 Testing

React Native projects are initialised with 'Jest' preinstalled. Jest can be used to test React Native application's. Marks will be awarded to application's which are accompanied with robust test scripts. Use your evidence document to explain, with screenshots, your method of testing.

Read more: https://jestjs.io/docs/en/tutorial-react-native

4.4.6 Project Management

In your evidence document, provide details (and screenshots) of how you managed your project. This may include software engineering principals and methodologies (e.g. Agile, KANBAN) which you have adhered to, and tools which you have used to aid project management (e.g. Trello).

Read more: https://www.atlassian.com/agile/kanban

4.5 Submission

Submission of this coursework will be online, through the university's Virtual Learning Environment (Moodle). You must upload **a single zip file**, which includes the following:

1. All of your source code along with any additional files that are required to run and build the application (delete your node_modules directory before submitting).
2. A screencast lasting no more than 5 minutes, which highlights all of your application's functionality.
3. An evidence document which describes, with screenshots, the following aspects of the project:
    a. How the application's structure and flow was designed from the API specification
    b. The information contained within the README file
    c. How version control software has been used within the project
    d. Details of the code style guide used and how it was used
    e. What testing has been carried out
    f. How the project was managed/what tools were used for organisation

4.6 Assessment Marking Criteria

	Fail (0 to 29%)	**Marginal Fail (30 to 39%)**	**3rd Class (40 to 49%)**	**2nd Class: 2 (50 to 59%)**	**2nd Class: 1 (60 to 69%)**	**1st Class (70 to 85%)**	**Exceptional 1st (86 to 100%)**
Functionality **(assessed via source code and screencast)** **50%**	A working application, which successfully interacts with the API and implements some API end points (multiple types of RESTful interactions).	Everything to the left but with the majority of API end points implemented.	Everything to the left but with all end points implemented. (Some end points may be partially working).	Everything to the left but with all end points working to specification.	Everything to the left, but also with Extension task 1 mostly completed.	Everything to the left, but with Extension task 1 fully implemented.	Everything to the left, but with Extension task 2 also implemented.
User Experience **(assessed via source code and screencast)** **20%**	Very little consideration to usability.	A demonstrated basic understanding of usability concepts.	A natural feel to app navigation and usability.	Consistent style, navigation and usability features throughout the app. Some error handling.	Use of a style framework for handling usability. Natural feel to navigation and consistent throughout. Good error handling.	Excellent use of style and usability, use of a style framework, and some consideration to accessibility. Application handles all errors gracefully with appropriate validate and checking.	Exceptional and consistent style, usability, use of frameworks and accessibility features.
Additional Skills **(assessed via source code and evidence document)** **30%**	Some skills mentioned in section 4.4 are partially evidenced	Most skills mentioned in section 4.4 are partially evidenced	All skills mentioned in section 4.4 are partially evidenced	Most skills mentioned in section 4.4 are mostly evidenced	All skills mentioned in section 4.4 are mostly evidenced	All skills mentioned in section 4.4 are fully evidenced	All skills mentioned in section 4.4 are fully evidenced in a clear and concise manner