Department of
Computing and Mathematics
ASSIGNMENT COVER SHEET

Unit title:	Mobile Applications Development
Assignment set by:	Ashley Williams
Assignment ID:	1CWK100
Assignment title:	Implement the Spacebook API as a React Native application
Assessment weighting:	100%
Type: (Group/Individual)	Individual
Hand-in deadline:	11th March 2022
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.

LO2: Research and demonstrate knowledge and practical application of current and novel mobile device techniques

LO3: Demonstrate knowledge and apply software development methodologies that are relevant to industry today

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 – Implement the Spacebook API as a React Native application

1. Introduction

This assessment is coursework based, and worth 100% 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

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

LO2: Research and demonstrate knowledge and practical application of current and novel mobile device techniques

LO3: Demonstrate knowledge and apply software development methodologies that are relevant to industry today

3. Coursework Overview

To complete this assessment, you are required to develop a mobile application. The precise detail of the coursework tasks are detailed in section four below. However, to summarise, you will be developing an application that can interface with an existing API. You are required to write the application in React Native and compile the code into an Android application.

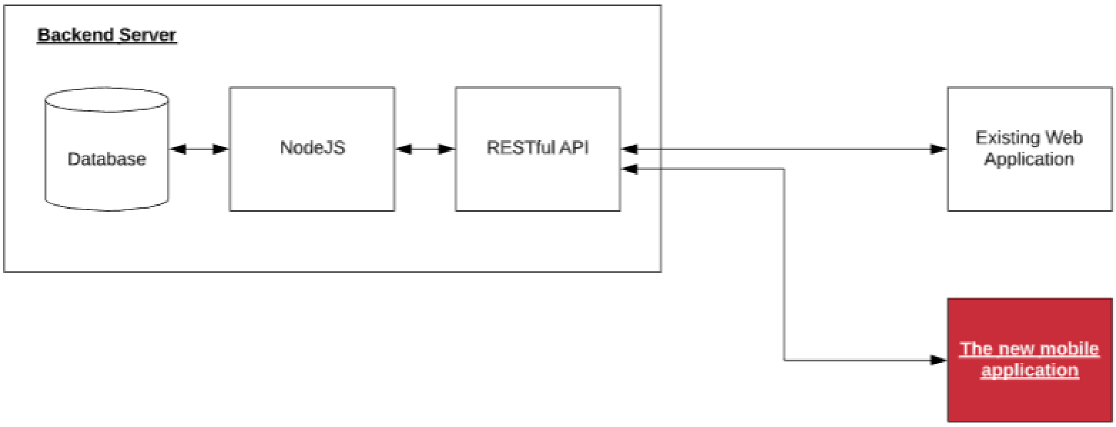
4. The Assessment (1CWK100)

4.1 Scenario

Spacebook is a totally unique, non-plagiarised social media platform which allows astronauts to communicate with each other. Users sign up for an account with Spacebook and can then add their friends, view each other's profiles, and write on each other's walls.

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

The Spacebook 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

Each week you will be provided with a checklist of activities working towards your assignment submission. More information on these activities will be given in the scheduled sessions. To begin, the most important activities are to fully familiarise yourself with the API, and to get your development server working.

4.2.1 API Specification

The latest version of the API specification has been made available on Moodle. Refer to the recorded lectures and labs to 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. Refer to the course material to 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 follow along with the weekly assignment checklists to ensure that nothing is missed.

4.3.1 Extension tasks

Extension task 1: Alter your application so that users can save local drafts of posts 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 the date and time when a draft is posted.

4.4 Additional Guidelines

The below will be assessed as part of the assignment.

4.4.1 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. **Please submit a link to your repository in your projects README file.**

Read more: See the "Introduction to Git" lab on Moodle

4.4.2 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, using a linter to help you. 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

Please state your chosen style guide in your projects README file.

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 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 of approximately 5 minutes, which highlights **all** your application's functionality.

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)** **60%**	The application runs and at least one end point has been attempted. Although there are large issues with its implementation.	Only a handful of end points have been attempted. At least one of them works to an acceptable standard.	Only a handful of end points have been attempted. All work to an acceptable standard.	Most of the end points have been implemented. Although there are large flaws with the design and implementation.	Most of the end points have been implemented. Although there are a few minor issues throughout.	All end points are implemented. Although there are a few bugs/inefficiencies.	All end points are fully working and implemented to a professional standard.
Extension Tasks **(assessed via source code and screencast)** **10%**	No extension tasks attempted	A demonstrated partially working solution to at least one of the extension tasks	A demonstrated partially working solution to both extension tasks	At least one of the extension tasks works, albeit with a few minor bugs or inefficiencies.	Both extension tasks work, albeit with a few minor bugs or inefficiencies.	One task is completed to a professional, fully functioning standard. The other is mostly complete.	Both tasks fully complete to a professional standard.
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 version control repository)** **10%**	No version control or attention given to code quality.	A repository exists with at least one commit. Little attention given to code quality.	A repository exists but has been used inconsistently. Some attention to code quality but inconsistent throughout the application.	A repository exists but has been used inconsistently. Code quality is good in places but lacking in others.	Consistent and appropriate use of version control. Good code quality to an existing style guide in places, but not throughout the application.	Consistent and appropriate use of version control. Excellent code quality to an existing style guide, though with a few exceptions.	Consistent and appropriate use of version control. Excellent code quality to an existing style guide and use of a linter.