GROUP ASSESSMENT ITEM COVER SHEET

Student Numbers:	Emails:	FIRST NAMES	FAMILY / LAST NAMES
3274832	c3274832@uon.edu.au	Karl	Foley
3185732	c3185732@uon.edu.au	Francois	Janse van Vuren
3252194	c3252194@uon.edu.au	Jacobus	Janse van Vuren

Course Code	Course Title
INFT2051	Mobile Application Programming
(Example) ABCD1234	*(Example)* Intro to University

Campus of Study: Callaghan (eg Callaghan, Ourimbah, Port Macquarie)

Assessment Item Title: INFT2051 Mobile Application Programming - Final Project Due Date/Time: 1/11/2018 -11:59pm

Tutorial Group (If applicable): Word Count (If applicable): Max 500

Lecturer/Tutor Name: David Cornforth

Extension Granted: ☐ Yes ☑ No Granted Until:

Please attach a copy of your extension approval

NB: STUDENTS MAY EXPECT THAT THIS ASSIGNMENT WILL BE RETURNED WITHIN 3 WEEKS OF THE DUE DATE OF SUBMISSION

Please tick box if applicable

☑ *Students within the Faculty of Business and Law, Faculty of Science, Faculty of Engineering and Built Environment and the School of Nursing and Midwifery:* We verify that we have completed the online Academic Integrity Module and adhered to its principles.

☐ *Students within the School of Education:* We understand that a minimum standard of correct referencing and academic literacy is required to pass all written assignments in the School of Education; and we have read and understood the School of Education *Course Outline Policy Supplement,* which includes important information related to assessment policies and procedures.

We declare that this assessment item is our own work unless otherwise acknowledged and is in accordance with the University's Student Academic Integrity Policy
We certify that this assessment item has not been submitted previously for academic credit in this or any other course. We certify that we have not given a copy or have shown a copy of this assessment item to another student enrolled in the course, other than members of this group.

We acknowledge that the assessor of this assignment may, for the purpose of assessing this assignment:

- Reproduce this assessment item and provide a copy to another member of the Faculty; and/or
- Communicate a copy of this assessment item to a plagiarism checking service (which may then retain a copy of the item on its database for the purpose of future plagiarism checking).
- Submit the assessment item to other forms of plagiarism checking.

We certify that any electronic version of this assessment item that we have submitted or will submit is identical to this paper version.

Turnitin ID: (if applicable)

DATE STAMP HERE

Signature: _____ Date: 31/10/2018

Signature: _____ Date: 31/10/2018

Signature: _____ Date: 31/10/2018

Signature: _____ Date: _____

Signature: _____ Date: _____

Insert this way

MEDICLIP

Group B

Group Members

Francois Janse van Vuren, Jacobus Janse van Vuren and Karl Foley

Table of Contents

Group Member Contribution

Francois Janse van Vuren: 33% contributed
Jacobus Janse van Vuren: 33% contributed
Karl Foley: 33% contributed

All group members have agreed to the percentages stated above.

Purpose of the Project

MediClip aims to improve information flow in the health industry. Hospital information systems are mostly terminal based. Currently patient information needs to be printed out or manually written down on forms to be available at the patient's bedside. MediClip solves this problem by having the information available in real-time on the doctor or nurse's mobile device. Patient notes are also saved directly to the hospitals database. Having doctors and nurses be able to view patient information and add notes to a patient's files while still at the patient's bedside, allows for seamless information sharing around the hospital. The systems currently in place can cause lost or inaccurate data because information must be written down and then transferred digitally, which creates more work for hospital staff.

MediClip is a cross-platform app as the team aims to have the app available on both iOS and Android mobile platforms. This is made possible by utilising Xamarin.forms cross-platform UI toolkit and writing all the code for both platforms in C#. The app works by combining the mobility and ease of use of mobile devices with the up to date information available in the hospitals SQL database. During the apps development, both the SQL database and the web API are hosted on Microsoft Azure. For security reasons this is not ideal. The aim of the MediClip team is to work with hospitals to integrate the web API with their own database on the local network. Ideally, patient data will only be accessible while on the hospitals secure Wi-Fi network and no data will be stored on the mobile device itself.

The MediClip app communicates with the database by sending JSON queries to a RESTful web API designed specifically for the MediClip app. The MediClip web API then communicates with the database to both POST and GET information as required. The information from the database is then shown on the mobile app in an easy to read format. The user is also able to add photos to a patient's note by utilising their device's built-in camera. This feature aids doctors and nurses in better understanding the patient's current condition. When writing a note, the user also has the option to use MediClip's shake-to-clear feature. This feature makes it easier to quickly clear a text field, as deleting text is far more tedious on a mobile device where the user does not have access to a mouse and keyboard. All doctors and nurses signed into the MediClip app will see the updated information as soon as it is added to the SQL database. When viewing patients via the app, they are all listed in the ward that they are currently in. This makes it easier for doctors and nurses to only focus on the information that is relevant to them at that time.

Reference List

A1

Xamarin Help Website, Adam Pedley, https://xamarinhelp.com/use-camera-take-photo-xamarin-forms/, Explanation of how to use the Xam.Plugin.Media package, to have our app use the camera to take photos and store the photos. We modified the provided algorithms to work with our code and had to install various NuGet packages to allow for camera permissions.

A2

GitHub Repository, James Montemagno, https://github.com/jamesmontemagno/MediaPlugin, Algorithms and an explanation of how to use them inside your code to obtain camera functionality. Had to change permission settings so that the application could access the device's camera and other media. Changed where photos are stored and how their file names are generated.

A3

stackoverflow, Jesper Christensen, https://stackoverflow.com/questions/35862657/disabling-back-button-c-sharp-android-xamarin-code-not-responding, This algorithm was used to disable the android back button when the user is on the home page, so they don't accidentally go back to the login page. No adaptions were needed as this was a simple algorithm.

A4

The Empire of Falleentium Fandom wiki page, CrusaderFFDP, http://falleentium.wikia.com/wiki/File:Bob_Ross.jpeg, No adaption of the file "bobross.jpeg" was required. The picture was used for a profile picture for one of our patients called Bob Ross.

A5

Psytherapy Website, Author Unknown, http://www.psytherapy.co.uk/about/blank-person-male-1/, The file was renamed from blank-person-male.png to blankPersonMale.png. The image was used as a placeholder for when a patient profile picture is unavailable in the database.

B1

Course Material, Author unknown, Lecture slides and lab material, Explanation and pre-written algorithms for getting information from an external database using JSON queries through a web API. Had to make changes to the algorithms to work with the API we created, we also changed different aspects to further work with our application e.g. posting and getting lists.

B2

stackoverflow, Ademar, https://stackoverflow.com/questions/9145667/how-to-post-json-to-a-server-using-c, Explanation and algorithms on how to POST information to our API. We had to make changes adapting the algorithm to work with our API.

C1

Course Material, David Cornforth, Week 5 Lab, Explanation and algorithms that allow the application to activate the accelerometer sensor and obtain readings from the sensor. We added a method to the algorithm that would clear a text field when the accelerometers shake reading returned true.

C2

W3 Schools, Author unknown, https://www.w3.org/TR/accelerometer/, Explanation and algorithms on taking the accelerometer readings and checking to see if device was shaken. We lowered the reading amounts (sensitivity) and changed the code to return a bool depending on if the device was shaken or not.

C3

Xamarin Forum, Laser, https://forums.xamarin.com/discussion/25375/datetime-tolocaltime, Explanation and algorithms that will allow our app to access the current date and time that is set on the device. We created a variable that calls the current date and time and added this to the string that generates a photo name. This helps make each photo name unique.

D1

YouTube, Tony Seo, https://www.youtube.com/watch?v=ddXVMdeA5D0, The video tutorial was used to learn how to create a RESTful web API service and how to connect it to an SQL database. Code for establishing a connection with the SQL database was taken from the video and adapted to suit MediClip's use case.

D2

stackoverflow, trx, https://stackoverflow.com/questions/41965076/web-api-to-return-result-from-sql-database, Explanation and algorithms that enable the API to GET data from an SQL server. The list was changed to suit the required model. The reader's while loop was also adapted to suit the SQL required table and model.