

Computer Games Development CW208

Technical Design Document

Year IV

Oisín Wilson

C00213826

[03/05/2020]

Faculty of \_\_\_\_IT CARLOW\_\_\_\_\_

Open-Book and Remote Assessment Cover Page

Student Name: Oisín Wilson

Student Number: C00213826

Lecturer Name: Oisín Cawley

Module: Final Year Project

Stage/Year: 4th Year

Date: 03/05/2020

Declaration

This examination/assessment will be submitted using Turnitin as the online submission tool. By submitting my examination/assessment to Turnitin, I am declaring that this examination/assessment is my own work. I understand that I may be required to orally defend any of my answers, to the lecturer, at a given time after the examination/assessment has been completed, as outlined in the student regulations.

Contents

[**Architecture**](#_807pxy9g865t) **5**

[**Features -**](#_srx8345srqew) **6**

[CRC Cards](#_3znysh7) **10**

[**React Native CRC Cards:**](#_64yvtdbs9njg) **11**

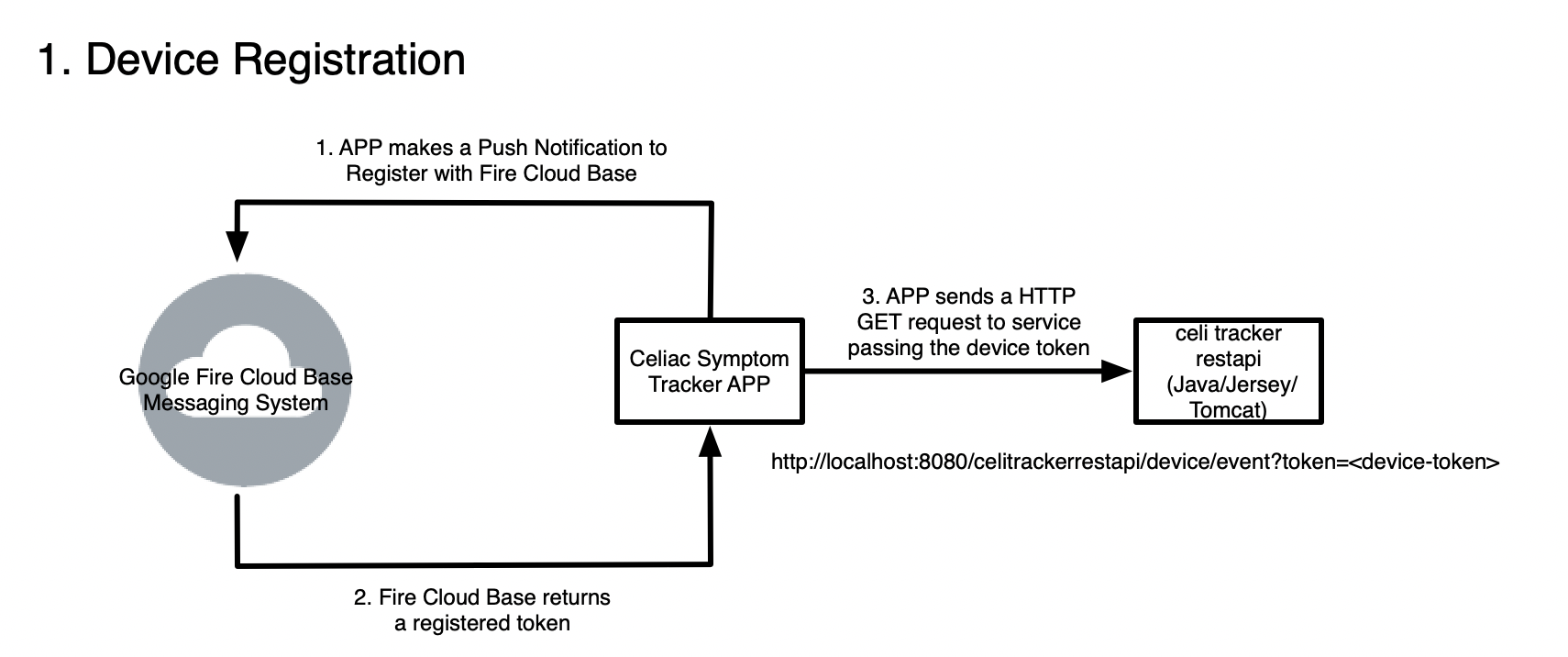
[**REST API CRC Cards**](#_tyvcl2s7canu) **14**

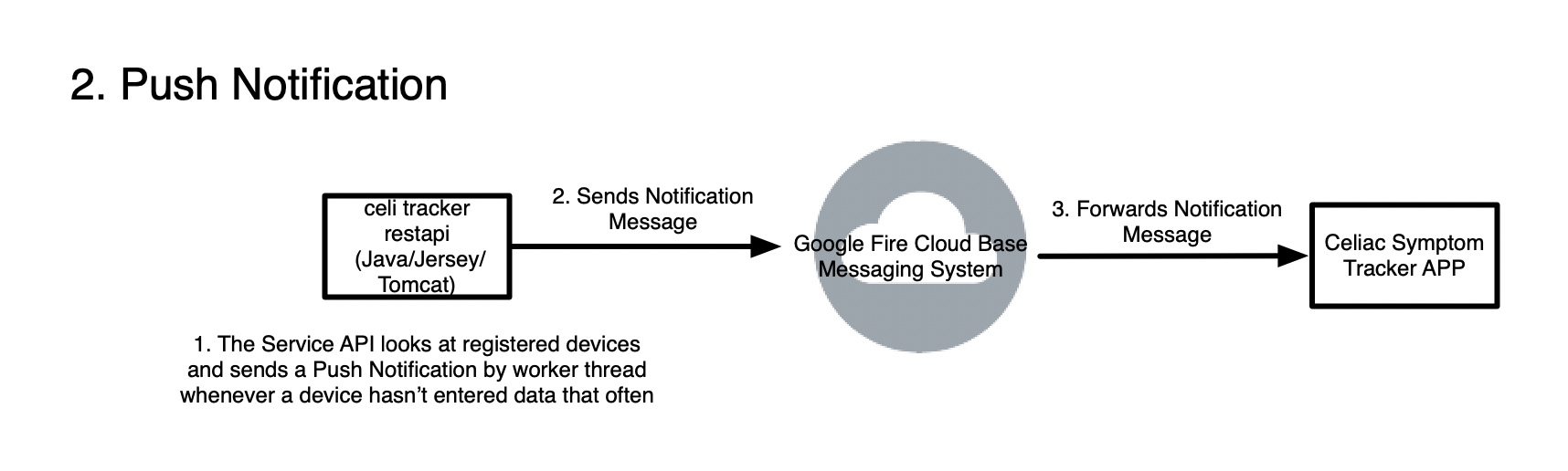
[**References**](#_2fnz2x82snsq) **16**

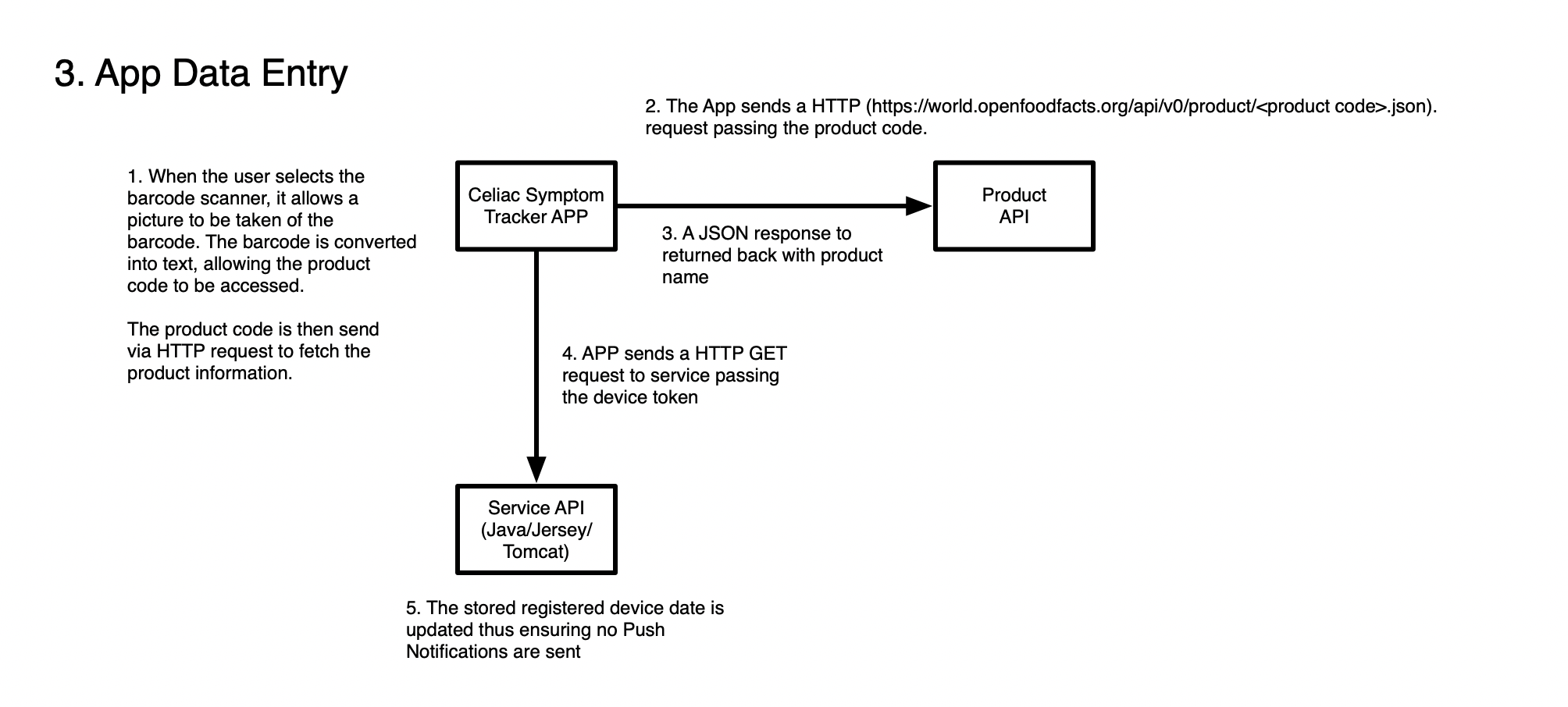
# 

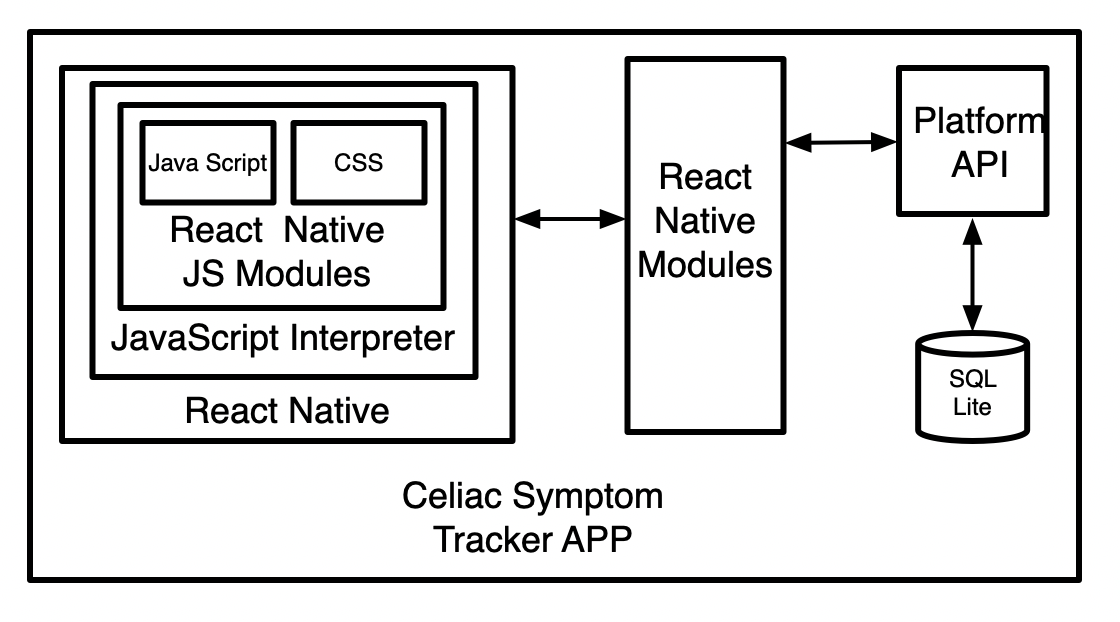
# **Architecture**

**Overall Architecture:**





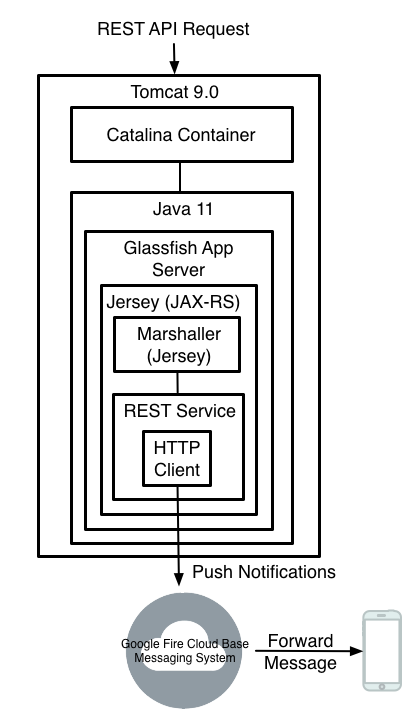


**APP Component Architecture:** 

**UML - APP Class Diagram**

# 

**REST API Component Architecture:**



# 

# 

# **Features -**

Feature : Circle View

Task 1:

Import necessary libs

Task 2:

Fetch,. Filter and store data from database

Task 3:

Pass data to render class

Task 4:

Work out angles needed

Task 5:

Apply imported data

Task 6:

Render

Feature: REST API

Task 1:

Create Notification manager

Task 2:

Set Notification Timer

Task 3:

Check device token list

Task 4:

Check if current time > the device time + notification timer

Task 5:

Send HTTP POST request to do push notification

Task 6:

Register new device in device token list

Feature: Input

Task 1:

Import relevant libs

Task 2:

Set initial state for class

Task 3:

Create local variables for the Input Text box and the CheckList

Task 4:

Create a drop down selection box for mood, gluten and meal

Task 5:

Create navigation button by the meal input for the barcode scanner

Task 6:

Use submit function which checks if any state was altered and inserts it to db

Feature: Calendar

Task 1:

Import relevant libs

Task 2:

Fetch,. Filter and store data from database

Task 3:

Render with date relative data

Feature: Screen Navigator

Task 1:

Import all needed classes from project

Task 2:

Create Stack Navigator

Task 3:

Create screen objects from desired classes

Feature: Bottom Bar Navigator

Task 1:

Import relevant libs

Task 2:

Create Image and Touchable areas

Task 3:

Attach Screen Navigator to buttons

Task 4:

Render

Feature: SQL Database

Task 1:

Import SQL libs

Task 2:

Create new database if not existing

Task 3:

Initialise Tables for Database

Feature: display slider

Task 1:

Create slider object

Task 2:

Attach Display types to Values on the slider

Task 3:

Render

Feature: Barcode Scanner

Task 1:

Import necessary libs

Task 2:

Get camera permission from the phone

Task 3:

Check if the camera has scanned anything

Task 4:

Create a confirm and retake button

Task 5:

Make a HTTP GET call to see if product code was in FOOD REST API

Task 6:

Return product name to meal input

Feature: Pi Display

Task 1:

Import necessary libs

Task 2:

Fetch,. Filter and store data from database

Task 3:

Pass data to render class

Task 4:

Apply imported data to renderer

# **CRC Cards**

# React Native CRC Cards:

|  |  |
| --- | --- |
| ***Class Name : App*** | |
| Subclasses : | |
| Superclasses : React.Component | |
|  |  |
| Responsibilities | Collaborators |
| This is the entry point for the project | Const - Screen navigator |

# 

|  |  |
| --- | --- |
| ***Class Name : InputScreen*** | |
| Subclasses : ScanScreen | |
| Superclasses : React.Component | |
|  |  |
| Responsibilities | Collaborators |
| This class has all of the input methods to the database | Const - Screen navigator |

|  |  |
| --- | --- |
| ***Class Name : ScanScreen*** | |
| Subclasses : | |
| Superclasses : ***InputScreen, React.Component*** | |
|  |  |
| Responsibilities | Collaborators |
| This will get permission to the mobiles camera | Const - Screen navigator |
| This will take the cameras input and scan for a barcode |  |

# 

|  |  |
| --- | --- |
| ***Class Name : HomeScreen*** | |
| Subclasses : CircleViewManager, PiDATA | |
| Superclasses : React.Component | |
|  |  |
| Responsibilities | Collaborators |
| This screen will manage the rendering of the data display types | Const - Screen navigator |

# 

|  |  |
| --- | --- |
| ***Class Name : CircleViewManager*** | |
| Subclasses : CircleView | |
| Superclasses : HomeScreen, React.Component | |
|  |  |
| Responsibilities | Collaborators |
| This will fetch the data from the database and sort through it and then pass the sorted data to the renderer class | Const - Screen navigator |

# 

|  |  |
| --- | --- |
| ***Class Name : CircleView*** | |
| Subclasses : | |
| Superclasses : React.Component, CircleViewManager | |
|  |  |
| Responsibilities | Collaborators |
| This takes the sorted data stack and renders it | Const - Screen navigator |
|  | m\_dateStack |

# 

|  |  |
| --- | --- |
| ***Class Name : PiViewManager*** | |
| Subclasses : PiView | |
| Superclasses : HomeScreen, React.Component | |
|  |  |
| Responsibilities | Collaborators |
| This will fetch the data from the database and sort through it and then pass the sorted data to the renderer class | Const - Screen navigator |

# 

|  |  |
| --- | --- |
| ***Class Name : PiView*** | |
| Subclasses : | |
| Superclasses : PiViewManager, React.Component | |
|  |  |
| Responsibilities | Collaborators |
| This takes the sorted data stack and renders it | Const - Screen navigator |
|  | m\_dateStack |

# 

|  |  |
| --- | --- |
| ***Class Name : CalendarView*** | |
| Subclasses : | |
| Superclasses : React.Component | |
|  |  |
| Responsibilities | Collaborators |
| This will fetch the data from the database and sort through it and then render the relevant data to the selected day of the calendar | Const - Screen navigator |

# 

|  |  |
| --- | --- |
| ***Class Name : DatabaseManager*** | |
| Subclasses : | |
| Superclasses : | |
|  |  |
| Responsibilities | Collaborators |
| This class will create an SQL database and hold all the functions to access and alter that database |  |

# 

|  |  |
| --- | --- |
| ***Class Name : PushNotificationManager*** | |
| Subclasses : | |
| Superclasses : React.Component | |
|  |  |
| Responsibilities | Collaborators |
| This class will register the device with the external Notification API |  |
| This class will push updates to the external Notification API |  |

# 

# REST API CRC Cards

# 

|  |  |
| --- | --- |
| ***Class Name : Device*** | |
| Subclasses : | |
| Superclasses : | |
|  |  |
| Responsibilities | Collaborators |
| This will store a devices notification token and the timestamp of its last update | String token |
|  | Long timeStamp |

# 

|  |  |
| --- | --- |
| ***Class Name : Event*** | |
| Subclasses : Device | |
| Superclasses : | |
|  |  |
| Responsibilities | Collaborators |
| This will serve as a way of changing the time between sending a notification |  |
| This will also be how you reset the device timestamp |  |

# 

|  |  |
| --- | --- |
| ***Class Name : NotificationManager*** | |
| Subclasses : Device | |
| Superclasses : | |
|  |  |
| Responsibilities | Collaborators |
| This class manages the actual notification call to the firebase cloud |  |
| This class indefinitely waits for updates from the react native app and after a certain amount of time it sends a notification |  |

# 

# **References**

**Web-site**

Umlet - UML application used for diagrams [**https://www.umlet.com/**](https://www.umlet.com/)