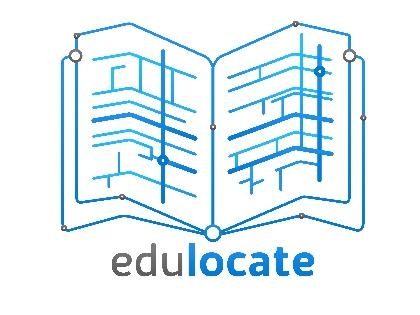


**Specification Document**

**Project Name: An app specified by the Department for Education**

**Team Number : 14**

****

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Document Information**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
| **Project Name:** | An App specified by the Department for Education | | |
| **Prepared By:** | Team 14 | **Preparation Date: 08/03/19** |  |
| **Email / Phone:** |  |  |  |
| **Document Version No:** | 1.12 | **Document Version Date: 08/03/19** |  |
|  |  |  |  |

**Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ver. No.** | **Ver. Date** | **Revised By** | **Description** |
| 1.0 | 15th Nov 2018 |  | Initial draft of specification consisting primarily of the “Purpose/Executive summary” |
| 1.1 | 18th Nov 2018 |  | Added “Existing systems” section to the draft |
| 1.2 | 10th Dec 2018 |  | Changes made to the “Existing systems” section and addition of draft test plan |
| 1.3 | 11th Dec 2018 |  | Addition of “Design plans” and “Deliverables” sections |
| 1.4 | 12th Dec 2018 |  | Addition of “Team roles” section |
| 1.5 | 16th Feb 2019 |  | Addition of “Functional and non-functional Requirements” |
| 1.6 | 17th Feb 2019 |  | Addition of “Dependencies” |
| 1.7 | 18th Feb 2019 |  | Addition of “Constraints” |
| 1.8 | 21st Feb 2019 |  | Addition of team coding standards document |
| 1.9 | 1st March 2019 |  | Development Documents added |
| 1.10 | 7th March 2019 |  | Addition of final “Definition of Terms” section |
| 1.11 | 7th March 2019 |  | Review of Document prior to submission, various formatting changes made to ensure document is consistent |
| 1.12 | 8th March 2019 |  | Final review before submission, contents table added. |

**Table of Contents**

Purpose (Executive Summary) Page 5

Background Analysis Page 6

Research of Similar Systems Page 6

Project Purpose Page 12

Team Roles Page 12

Deliverables Page 13

Project Plan Page 14

Functional Requirements Page 16

Non-functional Requirements Page 18

Dependencies Page 19

Constraints Page 19

Hardware & Software Page 20

References Page 21

Definition of Terms Page 21

Test Plan Page 22

User design & Planning Page 27

UML & Class Diagrams Page 38

Contribution matrix Page 51

## Purpose (Executive summary)

**--------------------------------------------------------------------------------------------------------------------------**

In this document we will show how we as a team have planned to fulfil the specification. The project we have been tasked with is to create an app for the Department of Education. This, would show the user information about schools. This document will give an outline of our plan for this task, so that our client and Newcastle University have an idea of how we mean to produce the app and hit deadlines.

Analysis

This section highlights the apps currently available on the marketplace that would be our competition or that use similar ideas to us. This will be used to identify the criteria we are looking for in our app. It will also show who pushes the demand for systems like ours so we can tailor our app to our target audience.

Roles and Deliverables

In this section we will go over the role of the members in our team. This includes what the team expects from each role and how this will coincide with the project plan. Furthermore, it will show how we as a team will provide the Department of Education our product.

Project Plan

In this section we have made a diagram which plans a timeframe for every part in the production of our app. This will identify key tasks and their deadlines so that we stay on track when producing deliverables.

Hardware and Software

In this section we have identified the main hardware and software which we plan to use to create our system. This includes for example the type of programming language we will be using at certain parts and why it has been agreed on.

Requirements

In this section we have outlined the requirements and main points the client and we want from our app. This then gives us a list of key features to implement with the importance and urgency of each.

Project Assumptions

In this section we looked at the parts of the specification which have some ambiguity or have been left to our control. Here we go through these and set assumptions which will be used throughout our project.

**Background and Analysis**

Domain Analysis

In the modern age, parents are becoming increasingly concerned with which school they allow their children to attend. However, despite the clear need for a simple, clear way to compare schools, the primary way now is via the government’s official website, which can be difficult to navigate and contains a lot more information than most would concern themselves with. The main goal of this project is to provide an alternative method that is simple and easy to use that shows the information most important to parents trying to choose the right school for their children.

For building our system, obvious area of interest will be mapping APIs and ways to get data from a server side database onto the app.

**Research on similar Systems**

**SchoolGuidei**

****

This image shows the comparison table for the app, below is a more detailed explanation of this.

Summary

The app ‘School guide’ allows users to compare a range of information about primary and secondary schools such as Ofsted rating and pupil/teacher ratio. The information is presented in a table format and a maximum of 5 schools can be compared at a time. To initially find schools, users can enter a postcode or the name of school in to a search bar. From this search a maximum of 5 schools can be selected and compared. Schools can also be looked at individually and when this is done the information is presented in more visual format with various infographics used to represent key data. Contact information and website links for schools are also provided.

Most of the data used is obtained from the Department of Education however School Guide also includes an independent star rating (out of 5) based on “key performance data”. Parents can also leave rated reviews, but this does not affect the on the main star rating.

Positive aspects

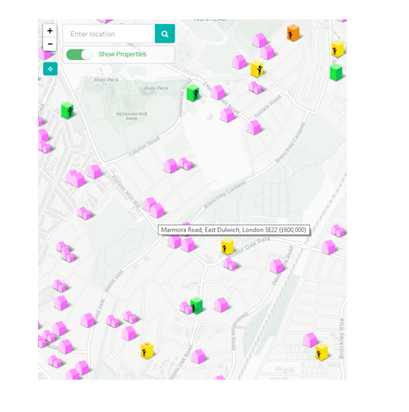
As the comparison table only contains a select amount of data the user is unlikely to have an overwhelming experience. Moreover, the purpose of the website is simple and well defined and there are no unnecessary features. This goes hand in hand with the simple colour scheme and layout.

Negative aspects

The list generated from the search can only be sorted with respect to the star rating and not, for example, distance. Also, the only filter provided is state/independent school even though many more could be included such as religion or latest Ofsted rating. There is also no maximum distance filter and instead at least 20 of the nearest schools are always shown meaning schools that are very far away are mixed up with local schools. These factors make it difficult for the user to find the schools they want to compare.

When comparing schools, it is not possible to customise the information displayed meaning users cannot hide information they are not concerned about. Another improvement that could be made would be to highlight the best school for each metric, for example the school with the best average attendance.

**Locratingii**



This image shows the map screen of the “Locrating” app, below is a more detailed explanation of what is shown here.

Summary

Locrating  is system designed for people who are looking to move to a new house and concerned about the schools in the local area. It has a lot of the functionality that we aim to produce for our application.   The main feature of the system is an interactive map displaying icons representing schools and listed properties (on Zoopla). A user of the system can filter the properties and schools they want to see on the map, selecting from a range of options such as OFSTED inspection rating and religious character for schools, and property type and minimum number of bedrooms for properties.

When the user hovers over a school icon the catchment area is displayed. The user can click the icon to look at information regarding that school including a link to their website. Properties and schools can also be added to a shortlist, from which they can be compared with each other.

Positive aspects

The app contains very useful information for new/perspective homeowners. Information about schools is presented neatly grouping related data in to different sections and makes good use of colour and icons. Overall the system is intuitive to use.

Negative aspects

Information not presented clearly when comparing. Each shortlisted school has its own block which contains all the relevant data for that school. This makes it tricky to directly compare specific pieces of data such as attendance or average GCSE score.

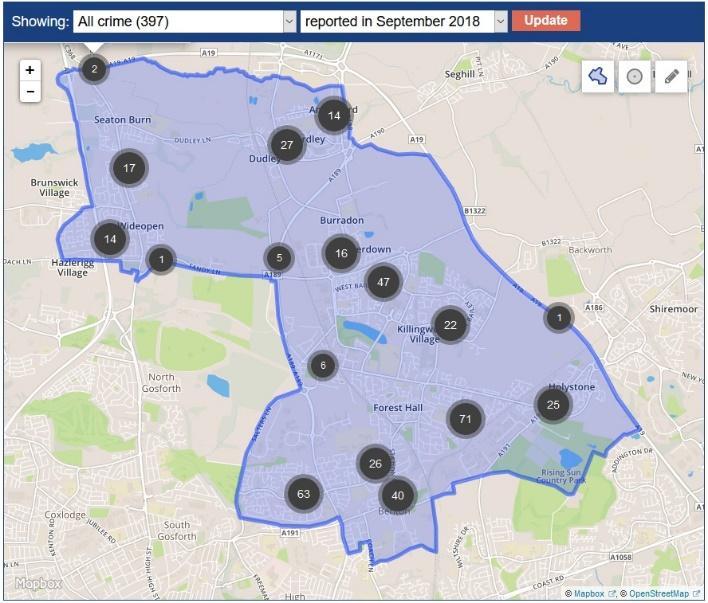
The system would be better if distances between schools and properties or a given location could be calculated and compared.

**Police.uk Crime Map iii**

Evaluation

Although this system does not cater to schools, it does have some interesting features for those who are concerned about their local area. The information is first displayed on the map with areas selected in one of 3 ways: By postcode, by a radius of a single point, or within a user-defined area.

Crime reports within certain areas are then shown in clusters which can be clicked on, which direct the user to another page which contains a table of the crime data.



The above image shows the the main user screen of the “police.uk” site,the blue area is to mark out the entire post code entered (other options are included, shown in the top right of the image) The circles with numbers are to show where data points are logged at.

The data can be pulled from the site in a .csv file, with sections for each police force being given independently, this will allow other developers to make use of the data supplied by the nation’s police forces.

Positives

The layout and general design of the site are very simple to use and allow the users to easily navigate and find the information they may be looking for. The system also supplies raw data for others to take and use in their own systems, as well as documentation of the use of the APIs used to create the system.

Negative Aspects

The site does not allow users to compare different areas, except for the numeric clusters shown within the area they have defined.

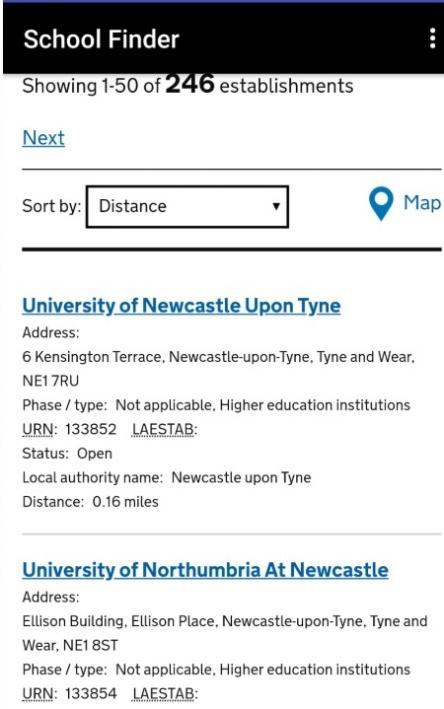
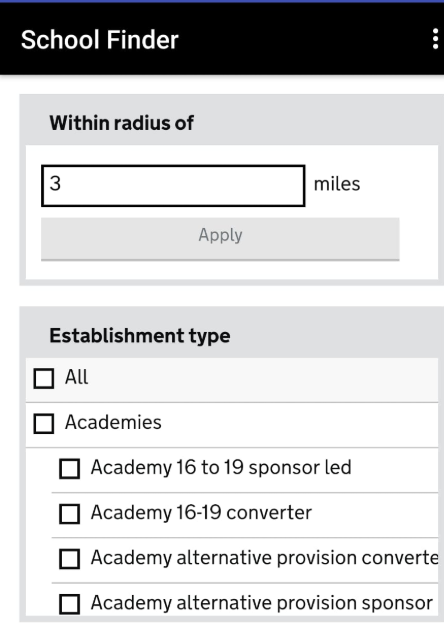
The system also does not allow the user to change the timescale for crime reports, it only allows one month at a time. An improvement could be made to allow users to search other the course of, for example, a year which could give them a much better idea about the levels and type of crime in a given area.

**School Finderiv**

Evaluation

School Finder is a mobile phone app that uses data from the UK Government to show data about schools it allows the user to search for schools by name of the school, location, and local authority.

When searching, the user can select a town, and radius from the town, as well as being able to select the type of schools they want to be shown. The information given by the app is rather limited, with the app generally feeling like a compressed version of the compare-school-performance.service.gov.uk site, which gives information about the Ofsted ratings and general performance of any given school, the site has a lot more functionality including the ability to compare schools.



The above images show the filter and results pages from the “school finder” app.

Positives

The app can be handy for starting research about local schools, as it allows to search in a radius from any given location, which can be a good starting point for any parents looking for details on local schools, the app also gives the contact information for the school which would allow parents to get in to contact with the school.

Negatives

The app has very limited functionality and a poor user experience, the app is clearly not properly designed for mobile and feels very difficult to use. Overall we would not recommend the app to anyone and feel the system as a whole needs to be redesigned to be more user and mobile friendly. The UK gov site is much more suited to the needs of the user and is easier to navigate then this mobile application.

Overall Summary

Overall, we feel that the Police.uk system is much more refined for ease of use and has a good way to display the information to the user. It could be a good base to work from when deciding how the user could interact with out app and how we could design the map screens.

One good feature from the “School Finder” app and the UK gov parent site, is that is allows for the user to compare the schools the find, this feature would be a great asset for our app.

Summary of existing systems

* If parents wish to compare data about primary and secondary schools then there already are a range of options available.
* A ubiquitous feature of the systems looked at is the ability to generate a search by typing in a postcode. This is an essential feature for our system however as we will be developing a mobile app, we will also include the option to use the users current location instead.
* Locrating and the crime map demonstrate that easy to digest, useful information can be conveyed with the use a map API.
* Systems also demonstrate that tables are an adequate way to compare numerous stats about numerous schools at once.
* Systems designed for mobile have poor user experience and limited functionality.
* The better systems are designed for desktop users and a lot of the functionality comes at a cost.
* For most systems, the user can not customise the data that they want to be shown or prioritise it in order of importance.
* Systems only show past/current data and do not make predictions. This is one feature that we will consider including in our system to make it more competitive.

**Project Purpose**

With our interpretation of the task described in the brief we will be building an application which will allow a user to check the schools in an area. We will be focusing on primary and secondary schools and using information on each school so that a user can compare schools.  
Our application will be able to use information from a database with the records of the schools results and general information about performance which has been given by the Department of Education. This will then be used to compare the schools against each other so a user can make a decision on the school based on fields or categories they choose (e.g. GCSE results).  
The base of our application is using Google Maps API so that the user can search a map of England. When the user types in the area for search our app would be able to find it quickly and easily. This system will allow an overlay of the schools on top of the map.  
We want out application to be simple and quick to run for our user. This is because our target audience will be parents, who are likely less confident using technology. Having a concise and simple to use platform will allow all parents to use the application better and in turn be more use.

**Team Roles**

Team Leader:

Sachin Chopra’s role is to lead the project, delegate tasks to the group by looking at people strengths to get the best quality work out of the group. Another responsibility is to make sure deliverables are completed on time so that the project can be completed in the desired time frame.

Deputy Team leader:

Sam Cernik’s role is to work with the team leader to assist with his responsibilities and to come to decisions with the group and help settle disputes that may arise. This also includes if the team leader is not present to become the acting team leader. Another responsibility is to document the minutes in meetings to keep a track of what has been proposed.

Lead Developer:

Luke Dyer and Alex Noddings’ roles are to front the programming for the project. This means that they are the first port of call when something involves the backend of the application. They will build and help support the junior developers in creating the applications software and oversee the code created.

Lead Designer:

Arman Jalilian’s role is to front the design for the application. This includes designs and mock-ups of the application before creation and to dictate the overall look of the app including colour scheme, logo and the flow of the app for the developers to create.

Junior Developer:

Matthew Jones, Daniel Boam and Matthew Elliott’s roles are to help the lead developers in the smaller aspects of the application. Another responsibility is to test the application with the help of the lead developers so that bugs are problems are found quickly.

Designer:

Charlie Leopard’s role is to help the lead designer in every responsibility that he has to create the best possible design for the app by having more than one input in the decisions made.

Documentation and Research:

Everyone is responsible for creating large documents which are needed to document our project. This also includes team reports and team contracts which will also be handed in for our team. The research into similar systems and other applications is also being conducted by the whole group which will be used for such parts like the dragons den

**Deliverables**

Final design Document:

A document based on the specification supplied by the Department for Education, which will detail how we aim to implement the specification into our final project. The project will be split in to 2 key sections: A C based backend, and an android application.

This document will be submitted on the 8th March 2019

C# Backend:

The back end of our system will run on a C# (using ASP.NET corev) designed by the lead programmers, this will allow us to have a modern, high performance system and allow better control of the data compared to having it included wholly within the android application.

Android Application:

The front end of our project will be a java Android Application that displays data to the user. This will allow us to serve to only Android systems, however, they account for roughly 85% of the market.

Poster for Trade fair:

A promotional poster for the app, created to display at the trade fair after the final submission date.

Final Specification:

The final specification will be a version of this document that contains all the necessary details of how we plan to carry out the project, as well as how the team is constructed and how we plan to work.

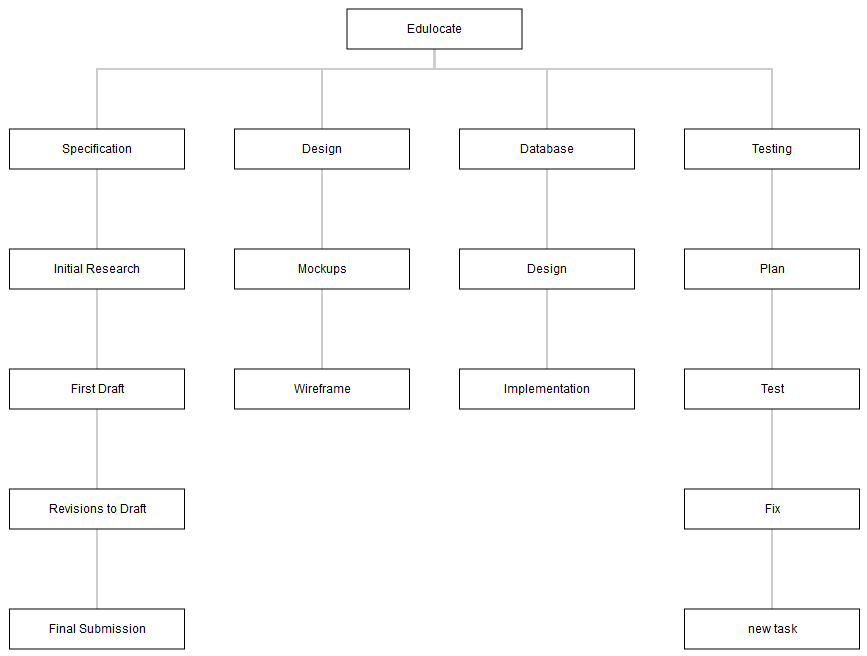
Testing:

We will test our application throughout the development process, in keeping with our Agile development model. With this development model we aim to locate and fix any issues with the program before they become a larger issue.

Final submission:

The final submission date will be 29th April 2019, this will be the final version of our system, created to the best of our ability. When the system is submitted, we aim to have no known errors remaining.

**Project plan**

****

**Functional Requirements**

|  |  |  |
| --- | --- | --- |
| Requirement | Importance | Comments |
| **Opening the app** |  | The first impression the user gets of the app |
| When app opened the edulocate logo will be displayed until it fully loads | M |  |
| A progress bar will tell the user how much of the app has been loaded | L |  |
| Once fully loaded, the home screen will be displayed to the user | VH |  |
| **Home screen** |  | User asked to enter basic info used to generate the map interface |
| User can determine local area by entering postcode into a text field | VH |  |
| Alternatively, user can determine local area by opting to “use current location” | H |  |
| App will remember the postcode the user entered last | M |  |
| User can determine the size of the “local area” by setting a max distance | M | Default 5? miles |
| User can select the type of school they are interested in, (primary, secondary, sixth form) | H |  |
| User can enter the google maps interface by pressing a “Go” button | VH |  |
| Button will only be available to press if valid information has been entered by the user | L | Could have default values that are valid instead. |
| **Google maps interface** |  | Main part of the app |
| Will show a map of local area determined by the user | VH |  |
| Will show pins to represent schools in the given radius | VH |  |
| Pins will convey some information about the school | M | eg. colour to indicate ofsted score |
| User can tap a pin to select the school and reveal additional information about it | VH | Presented in a visual manner |
| Additional information includes:    Exam results, Ofsted rating, Population, Funding, Headmaster name, website link  **[Add and remove stuff]** | H | Research what info parents find most important |
| Once selected the user can save the school to a shortlist or add to the comparison table | VH |  |
| Once selected the user has the option of calculating the commuting time from their home to the school | L | Using car or public transport? |
| Once selected user can access a schools twitter handle | M | Link or embed? |
| Once selected, user can access a news feed for the school | L | Could have option to follow a school |
| User can tap the corner of the screen to reveal a drop down menu of filters | H |  |
| User can adjust filters to ignore schools that are not relevant to them, | H | Eg.  due to the school’s religion/sex/type |
| ?visual feedback form filters? | L |  |
| App will generate predicted results for schools | L |  |
| App will limit how much user can scroll and zoom out, | M | Would prevent user from viewing another country |
| User can choose between a “Satellite” or “Map” view. | L |  |
| **Comparison Table** |  |  |
| Each row in the table will represent a school that has been added and each column will represent a stat eg. average grade achieved in maths | VH |  |
| User can select/change the stat shown for each column | H | Drop down menu, can not select same stat twice |
| App will limit the number of columns the table has to **(8?)** | H |  |
| The highest and lowest result for each column will be highlighted | M |  |
| Table can be exported for future use or for sharing purposes | L | Csv format? email? |
| **Website** |  |  |
| Will only allow access if valid login is entered | VH |  |
| User can delete a school if it has been closed down | H |  |
| User can add a new school if one has opened | H |  |
| User can update details for an individual school | H |  |
| Before any changes are made the user will be prompted  and asked to confirm the changes | M |  |
| **Database** |  |  |
| Will automatically update annually using data from the department of education | H |  |
| Multiple copies will be stored so that if a mistake has been made, it can be easily rectified | H |  |

**Non-Functional Requirements**

|  |  |  |
| --- | --- | --- |
| Requirement | Priority  (H, M, L) | Comment |
| Software Documentation | H | It is vital that we keep high quality documentation of the software to allow us to fault find, and to explain how to fully use the software. |

**Dependencies**

With our app “EduLocate” there are certain parts which might have large dependencies when looking at the overview of what we are able to do. The first being that we are creating the app through Android Studio and nothing else. We would like to create the application for the IOS Appstore and more but down to time constraints for the projects it is not foreseeable for us to make a cross platform application. As Android however make up more than 80 percent of the world usage of smartphones, we are currently only using a small percentage of potential users. As IOS and Android makeup of the world usage when created for IOS of app is 99% when create that would solve this issue.

Another dependency would be the fact that because our app has several parts which the user would have to be connected to the internet in some way, the app is severely at a detriment if there is no connection. Apart from the fact that you would have certain schools from the app on your phone saved already all the Google Maps and news feed features included with the app would be rendered useless. As there will also be a website where someone will be able to update the information on the database for the app an internet connection for our project is vital.

Furthermore, we are going to be dependent on the Google API for the app as the maps section is the crux of the EduLocate user path. If Google maps is down or there is something wrong at their end it would definitely affect our app as the user would not be able to access the main searching part. As this is an essential part it could only be rectified if it gets fixed by them.

Lastly, we are dependent on the actual schools updating their information via the website for our database. If this is not done there will be several glaring errors or not updated information with the schools. This means it will be hard to compare data when they are not from the same time periods giving skewed results. If only some of the schools update their information, then it will still not be that useful compared to most of them.

**Constraints**

|  |  |  |  |
| --- | --- | --- | --- |
| Constraint Type | Constraint | Importance | Description |
| Legal | Gathering information from users to define feedback | L | Due to ethical reasons and GDPR storing personal information about a  potential user is quite a difficult process with strict restrictions on who  can view this information and how it is stored |
| Technical | Amount of data we can display | M | Due to the sheer quantity of data provided it is hard to truly display all  information about a school a user might want. This meant we had to  shortlist many categories that the user would want to the most. |
| Technical | Operating System | L | We are constrained in which Operating system we are developing for.  Due to the experience of our developing team we are only developing  for Android rather than for IOS, in addition developing for both operating  systems requires additional resources that would delay the development  of the app. |
| Financial | Web Server | H | Hosting our own server to host both the database and website could  prove to have a financial strain in the future of the app, if it were to be  maintained. |
| Time | Deadline | H | We are locked to a strict submission deadline through University, so we  must delegate and make changes to tasks accordingly. This could mean  that features and quality of life aspects of the app could be dropped  along the way to get to the deadline. |
| Financial | Google API vs SDK | H | Using Googles API has a limit for normal users, therefore we must be  careful with the number of requests we make per day to ensure we do  no incur further financial cost of using the services. |

**Hardware & Software**

Hardware

Our hardware requires for the project are rather few. The main requirements will be access to a computer running a modern operating system and with access to Android Studiovi.

We will also, most likely, require access to android mobile phone devices to test the app on.

Software

We will be using a range of software in this project, this is due to the large number of different resources we will require to complete the task.

The most important software is Android Studio, which we will be using to create the actual app.

We will also require the use of some image editing software i.e. adobe photoshop.vii

In addition to this, we will also require some way to store any documents or data.

For the storing of all text and design documents we have opted to use google Driveviii, this allows us access to our files from anywhere, as well as a way to share files within the team.

We will also be using Githubix for this project, this will be used to organise the way the code for our app is written. It allows us to easily combine the work of multiple team members into one version and allows for easily reviewing of submitted code.

**References & Resources**

i Schoolguide: <https://play.google.com/store/apps/details?id=ca.boxclever.schoolguide&hl=en>

ii Locrating: https://www.locrating.com/

iii Police.uk Crimemap : https://www.police.uk/northumbria/E6/crime/

iv School Finder: https://play.google.com/store/apps/details?id=com.trentapps.school

v ASP.NET core: https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-2.2

vi Android Studio: https://developer.android.com/studio

vii Photoshop: https://www.adobe.com/uk/products/photoshop.html

viii GoogleDrive: https://www.google.com/drive/using-drive/

ix Github: https://github.com/features

**Definition of Terms**

This section will aim to explain, in layman’s terms, any technical language used in the document.

Agile Development

A development approach that relies on ‘Sprints’ to produce functional code every iteration. In our case a Sprint lasts two weeks, therefore it is expected to have some working application every time a Sprint ends.

API

An API (‘Application programming interface’) can be an interface for any framework or SDK we use. For example, when making a request via the google maps SDK, we must place a request with the Places API as the relevant code/service is not stored client side.

ASP.NET core

ASP.NET Core is a web framework that was developed by Microsoft and written in C#, it will be used to setup an API to allow requests to be made by our client.

Client

A client in terms of our application means the users phone, where the client only ever makes requests to the server

Server

The server is where our database and website are stored. The server also has additional responsibilities handling requests from the client.

Database

A database is a uniform way of storing data, which is what we have done with the schools’ data.

Entity Framework

Entity Framework is an object mapping framework which we used to turn our text format data about schools into C# objects and then placed into the database.

SDK

Unlike an API that only provides certain services that would be difficult to implement, an SDK provides tools to produce your own software applications on a specific platform.

Endpoint

An endpoint is where the request from our client will be received.

HTTPS

HTTPS is used for secure communication over networks or most commonly the internet. All requests are encrypted to ensure data cannot be interrupted during transit.

OS

An operating system is what the application will run on, for example we are only usingAndroid instead of IOS. Another example of an operating system is Windows or Linux.

**Updated Test Plan**

Objectives

We will be testing with user acceptance testing, which will be done every single sprint in order to test our newly implemented features that we have created within that sprint and see if they break any of the previously implemented functionality/code. This would be done by the product owner who knows exactly how the feature should work in order to confirm if the functionality is working as intended.

The developer team will be using unit tests as well as the user acceptance testing mentioned above, which will be created as the developer’s code. These will be used by the developers in order to test if their code is working and then by the lead developers when checking pull requests when the developing team submit their code for inspection. With our Java code, the dev team will be unit testing with JUnit and the backend C# code will be unit tested with NUnit

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no. | Test page | Functionality we are testing for | Test Description | Expected result | Actual Result |
| 1.0 | Home Screen | A user should be able to put their postcode in to see nearby schools. | Input a postcode into the search bar | It will display schools near the inputted location. | User can see nearby locations to search for when inputting postcode, address or location. |
| 1.1 | Home Screen | A user should be able to allow location services in order to save time inputting their location to find schools. | Allow location services and go onto the map | It will display the schools near the user’s locations | The app notifies the user if they want to use location services instead and shows them locals places to search from |
| 1.2 | Home Screen | A user should be able to set a maximum distances for finding schools in order to only find close schools. | Set a maximum distance on the home screen and then input a location | It will only display schools within the maximum distance from the point | A slider enables the user to select the max distance from a location and show all schools in that limited radius |
| 1.3 | Home Screen | A user should be to set what type of school they want to select e.g. Primary, Secondary & College/Sixth Form | Select a type of school from radio buttons/drop down box | It will only display the three possible options that the user can pick | A radio button display shows that |
| 1.3 | Home Screen | A user should be able to set a commuting distance for schools so that they can see schools within a certain travel duration from their location. | Set a commuting distance on the home screen and then input a location | It will only display schools within the commuting distance from the set point. | N/A this feature is yet to be developed. |
| 1.4 | Map Screen | A user should be able to immediately exclude religious/single sex schools so that they won’t be shown in search results. | Attempt to filter single sex schools and then input a location that is next to a single sex school that you have filtered out. | It won’t show any of schools that should have been filtered out. | The app attempts to show non-religious/religious schools however due to lack of information provided by schools it may not always work |
| 1.5 | Map Screen | A user should be able to see brief details about a school. | Click on a school on the map | It should display the details about the school including its rating | On click further information about the school is shown. |
| 1.6 | News Screen | A user should be able to see social feeds from schools. | Input a location and then click on a school then its twitter icon. | It should display the school’s twitter feed if it has one | N/A, this feature is still in development |
| 1.7 | News Screen | A user should be able to see news stories about nearby schools. | Input a location then click on the news button. | It should display news relevant to schools in the area around your search area. | N/A, this feature is still in development |
| 1.8 | Map Screen | A user should be able to see a visual comparison between how good schools are on the map. | Input a location. | It should have some form of showing you which schools are better than others just from the map | N/A, this feature is still in development |
| 1.9 | Map Screen | A user should be able to see a generated score showing how good the schools are. | Input a location with a couple of filters | It should display a score to the user based on the filters that the user has set when searching for schools | N/A, this feature is still in development |
| 2.0 | Map Screen | A user should be able to filter search results to change the produced score of the schools. | Input a location then try to apply a couple of filters to see if the scores change | It should display a different score dependant on the filters that the user is searching with | N/A, this feature is still in development |
| 2.1 | Comparison Screen | A user should be able to generate a table of details based on the selected filters by the users | Input a location and select two schools then click the compare button | It will go to the comparison screen and show the stats of both schools in a table | A table view will be displayed based on the filters the user had selected at the time. |
| 2.2 | Comparison Screen | A user should be able to export data to an excel/table file to look at the results later or share them | Input a location and select two schools then click the compare button then click the button to export the data | It will produce a excel/table file containing the comparison data of the schools | N/A, this feature is yet to be implemented |
| 2.3 | Settings Screens | A user should change accessibility settings for text/icon size and colour scheme. | Go to the settings menu and adjust the text size. | It will adjust the text on the app to the size chosen by the user to make it more readable for visually impaired people. | Some setting have been implemented and a button is available to change some accessibility settings. |
| 3.0 | Website | A user should be able to login using their login credentials | Enter Username and password, submit form | If credentials are successful then the user will be informed and given access, and informed if they are denied access | N/A. this feature is yet to be implemented |
| 3.1 | Website | A new user should be able to request a new login if they are an admin | Press request new login, and enter credentials | If approved by admin login details will be emailed/sent to user | N/A, this feature is yet to be implemented |
| 3.2 | Website | A user should be able to view information about schools through the database | Press view information about schools | A new page with a table view of the schools and information associated should appear | N/A, this feature is yet to be implemented. |
| 3.3 | Website | A user should be able update information about a school/schools | Press update fields on database view | User should be able to re-enter information for that field (will be validated) | N/A, this feature is yet to be implemented |
| 3.4 | Website | A user should be able delete a school | Press remove school on database view and confirm the user wants to do so | It will allow the user to delete the school after confirming they want to | N/A, this feature is yet to be implemented |
| 3.5 | Website | A user should be able to add a school | Press add school on database view and confirm the user wants to add an empty school record | It will allow user to add a school after confirmation and focus on the empty school to add field information | N/A, this feature is yet to be implemented |

\*NB: All features that have not been implemented are subject to change as the submission date for the specification and the final submission date is a month apart.

**User Design & Planning**

Colour Scheme

The colour scheme chosen is an exact replica of the colour scheme used within the **gov.uk** website. We attempted to experiment with different colour schemes in order to go with a unique approach. However, having spoken to both client and target audience, we found that the colour scheme below would give a user a sense of authenticity and security whilst browsing through the app as it has a link to a government-based website.

Keeping the colour scheme to a maximum of four would not only encourage us to take a minimalistic approach with the design; but also provide a colour coded navigation system that a beginner user can benefit from. For example, blue for icons/symbols, light-blue for subheadings/small navigations, black for title and text and white for logos and text on top of elements. The use of these colours may change depending what approach we take on layout.

The minimalism would provide a professional link with the user and the app itself. It would also allow a user to clearly see information as we would believe choosing a school is one of many life-defining decisions.

https://lh3.googleusercontent.com/HNNyh4CXsghKZ-aeQ8wtutreNB0gMeWkcx8JQNzAdnNnIzRml5-yb3k5psR6GEd_66ZW9amwUDpAmIWPSljXkxo7BM6Ikf3L5pscS5tGX2r7dZcB2ijXaM_KXNWPEX6L1LJySTTB

Logo

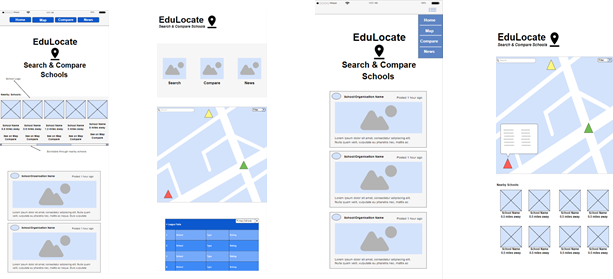


This logo was created in the final mock-up just for layout purposes. However, having spoke to the client, the use of the **gov.uk** logo was permitted. This would be an advantage for our team as it would both give the user a very comforting feeling as the logo is used site-wide by the government, and allow us to put our focus to other functionalities of the app. Although it increases productivity and time-efficiency in other departments, creating a logo would ensure that our app is unique which would also be beneficial. This logo would not be final as it requires thought in the fonts used however it is a good foundation to develop from.

**Mock-ups**

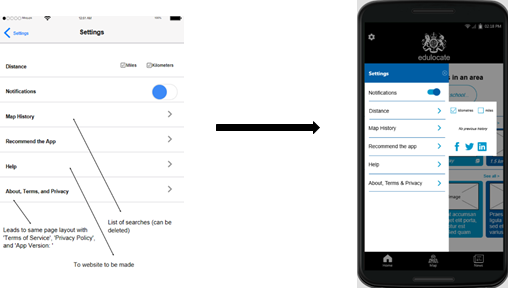
Introduction

Before proceeding to build a functional prototype of our app, we wanted to design how it will look like and function on a basic level by creating wireframes of the app to show a simple representation of the features that need to be implemented. We used the information we had gathered from the similar systems in the market and what we wanted to take from them, as well as how to differ.

The wireframes were created using an online resource (**Moqups.com**) with various available templates – this allowed us to visualise the app in terms of colour scheme, layouts, and functionality such as how the page would link together and how filters would be applied. Several designs were made for different possible layouts, so we could take different features from different mock-ups and come out with the best design to continue with. 

Development from wireframes into functional mock-up

Working from the initial wireframes, we thought as a team that keeping the app more streamlined would benefit the user in finding information more efficiently. We did this by not cramming pages and only offering the most useful information. Then, if a user needed to, they could find more information using external links. We took some of the wireframes and used either the information provided, layout used and even the scale of elements to develop into a mock-up that we could roam and see where we could go from there. The functional app was made using ‘**MockPlus**’, a downloadable application that offers mock functionalities an app would require.

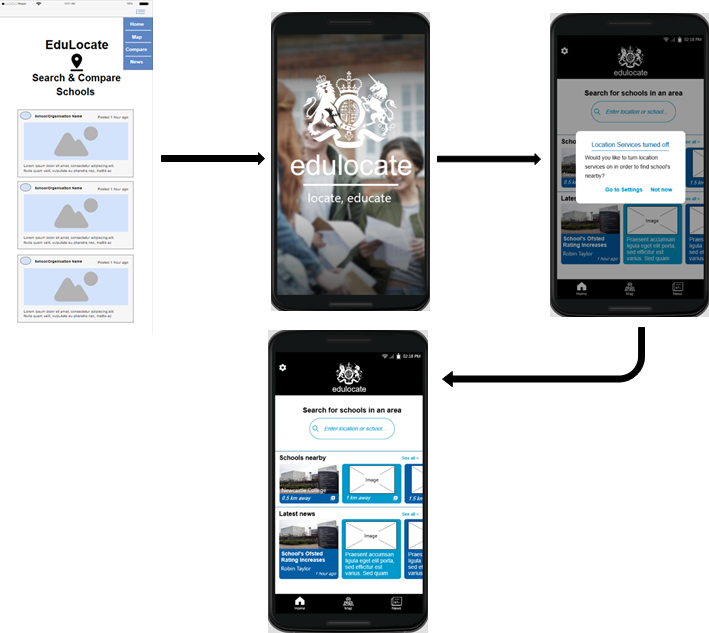


Settings

The settings page of the app is ultimately a section where a user can learn about the app itself and edit how they would want information to be displayed. Making a whole page for the settings may confuse some of our target audience about what section they are in. Creating a pop-up from the side would still give a background view of the current page. Further developments to this section could be an accessibility option; giving our current audience a chance to view or learn about information more in a way that is beneficial to them. However, we are currently in talks of what accessibility features we would need to implement. Some features of which we could implement that are used by current apps in the market would be:

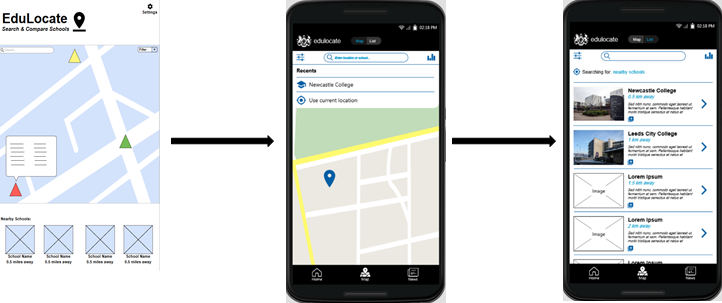
* Colour blind indicators and test on first launch of app.
* Text under all icons.
* Magnifying small prints and information not easy to read.
* Narration of text on-click.
* Ensuring all images are of high quality (preferably 300ppi) so that when it comes to scaling an image on a larger device or enlarging images by hand, the image does not become unrecognisable.
* If app is used on larger screens, the amount of information displayed is expanded.
* Controlling the amount of information displayed to either avoid overwhelm or avoid frequent button clicking for those who have trouble with motion.

The features that we’d implement would be within the ‘Settings’ section to activate to both be an advantage for those in need of the features and an advantage for those who do not need the features.



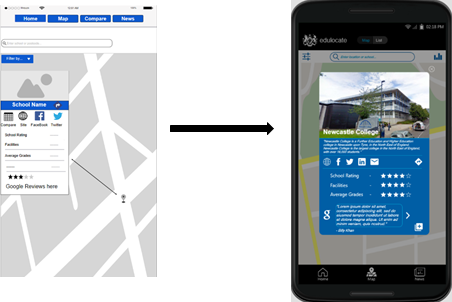
Home

The main development from the initial wireframe for the homepage would be the way it presents itself. For example, on opening of the app, the homepage is the first page that would be viewed. We thought that an introduction to the app would give a user comfort in using the app as it’s another feature that hasn’t been implemented in other apps. Also, the background image used in the introduction page would further emphasise that the app is within the category of education. We also implemented a feature that allows users to turn on their location services right before they reach their homepage. This is to make the user experience better so that they can view all necessary information and search using the “nearby” feature. The developments that were made on the actual homepage include a horizontal scroll bar as they will be vertically scrolling for most of the app journey and quick access to features so that the user would not need to go onto different elements in getting a result that could be gained from a quick click.



Maps

The main change from the wireframe and the mock-up was the separation of the ‘Nearby Schools’ and the actual map view. This was done using a separate ‘List’ page. This also goes back to the “not cramming” factor that we are trying to maintain. By separating both pages, it gives a user an option of how they want their schools to be laid out. By laying out in a list form, they can see the mileage from themselves or an address they’ve inputted and by laying out in map form they can clearly see the exact location of where the school is based. One thing that we did not implement in the maps was the colour code of the school’s rankings. This is because we are still in talks of how we would want that information to be laid out and so implementing that would not open our minds to other features.



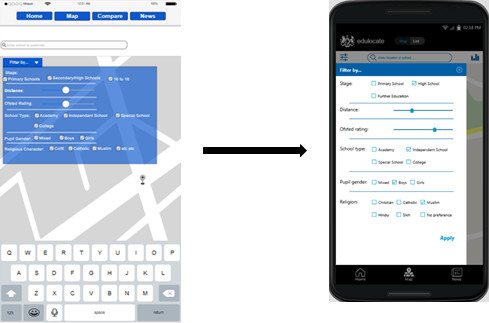
School Information

We stuck quite closely to the wireframe design when making developments to the mock-up. This is as we wanted a base for how we wanted to information about a school to be displayed. Again, we did not want to stick all the information about the school on the screen, but we wanted to display all the necessary information and then allow the user to click on certain elements to be taken to external sites where they would hold more information. If a user wanted to see more information about the school side-by-side with another school, the comparison icon in the bottom right corner would allow the user to add a school to the comparison selections.



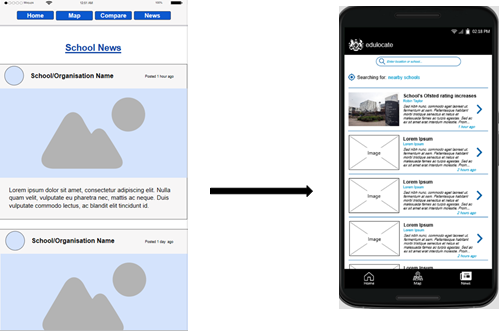
School Comparisons

The school comparison table within the functional app is very similar to the original wireframe; just a touch up on the design. A feature that would hopefully be added would be different views of the information e.g. chart views and landscape views, just for the user to see a wider range of results in their preferred view. We also added an additional ‘Recent Comparisons’ section for the user to come back and see previous school comparisons without having to search over again. The comparisons can also be accessed easily from most pages, mainly the map and the list as there is a direct icon for access.



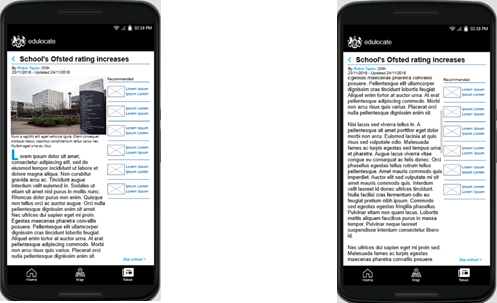
Filter by…

A user would use a filtering option to make their roaming experience smoother whilst also increasing time effectiveness.  The display of how we offer filters would need to be much more streamlined as it looks as if information is crammed; a factor we do not want to implement. So far both the wireframe and functional mock-up look similar however knowing that we would need more information to be filtered by, the method of displaying it all on one press of a button would not be ideal. To develop from this, we could create drop-down menus on each category of the filters.



News

Adding a news page would allow a user to search for news about a given area or a given school to allow them to gain a more informed decision on where to apply for themselves or relative.  A title of an article would require the date published, author, publisher and brief description on the actual article. The developments have included a search bar to search for a specific thing and a more streamlined view of the page. Below shows an example of an article when clicked on.



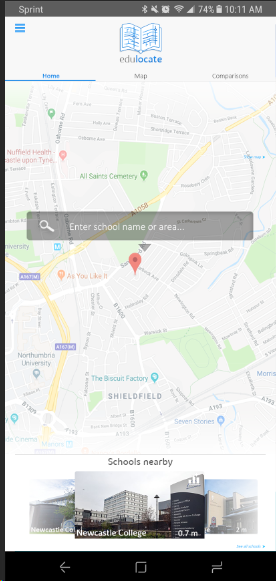
Maintaining the nav bar on-screen throughout the journey of a user would allow them to never get lost and return to a previous point to either start the journey again or move on to another feature. The article itself is built up of two columns, one with a scroll bar (the actual article) and a static column which a user can continue their search of articles and select ones related to the one they’re currently on. The thought process behind these decisions was again to improve a user’s journey by making it much easier for them to roam.

**Evaluation of mock-up**

The mock-up created was not to achieve a final design on the app or to achieve an app that would look like the final product; but to build both a foundation for the designers to develop from and a sense of workload to be implemented by the programming team. Although the colour scheme seems to be a final choice, the layout can be improved vastly to achieve an outcome of a very streamlined app.

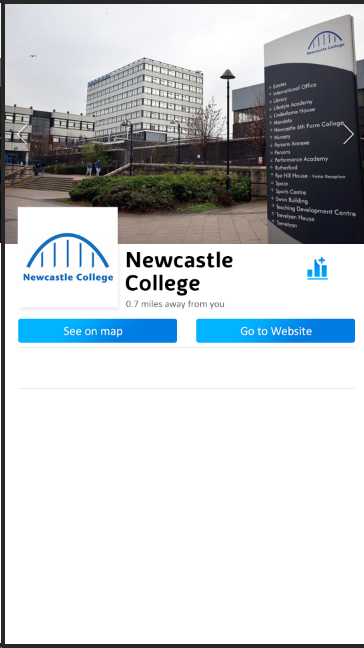
**Final Designs & Colour Schemes**

Our finalised design kept the functionality of our original mockups but the decision was made streamline the app even further. The goal was to achieve some form of minimalism so the user would have the easiest time using the app. This was also reflected in our colour scheme, where we decided to use a modern aesthetic for Edulocate to overall make visibility of key features more obvious. This meant using lighter shades of blue, but also making use of whitespace to focus the user’s attention on the key aspects of a page, however we still wanted to keep the colour scheme somewhat inline with the .GOV websites colours.

Final Homepage Design:

As you can see by the final homepage design there is a lot less emphasis on darker colours. The screen seems a lot brighter and easier to navigate. Furthermore, the inclusion of a carousel at the bottom makes the home page more functional. The inclusion of a singular search bar that collapses makes the homepage seem cleaner and less cluttered. The button to collapse is large to make it obvious to a user that there are further options to select from.

**Information Page Design (not completed):**



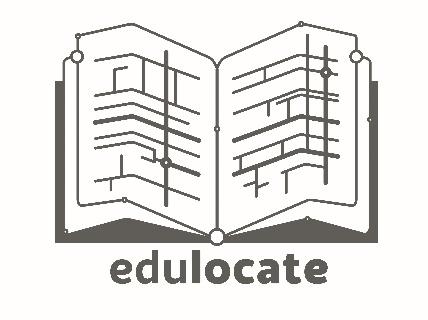
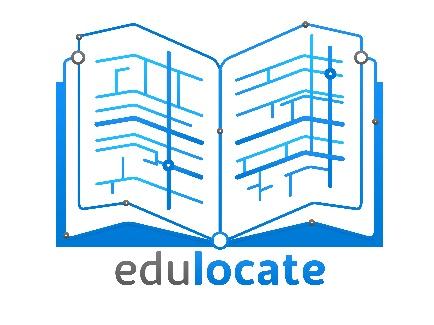
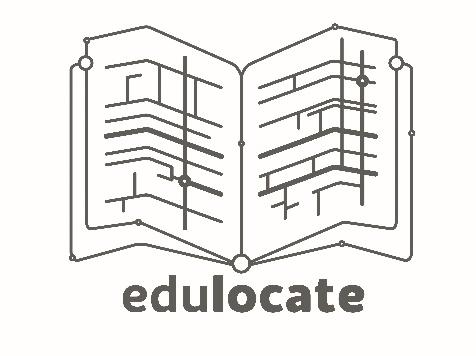
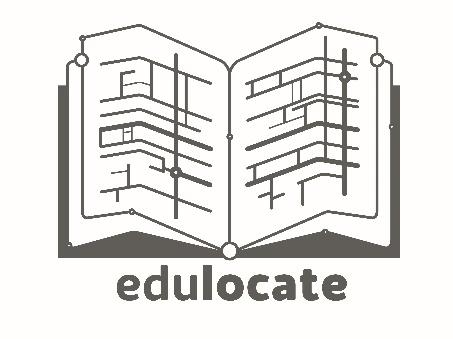
Our information page even though not completed shows some direction of what we are trying to achieve. The more extensive use of blue makes the buttons standout whilst the use of whitespace again makes the key features stand out. This page also adds functionality in the ways of viewing information about the school.

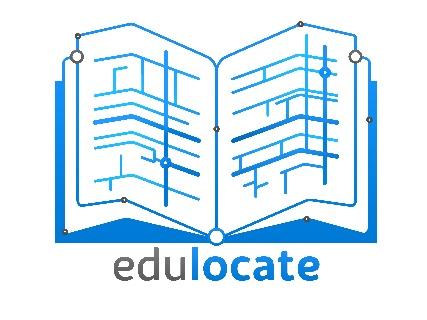
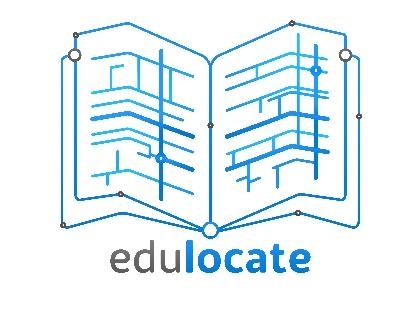
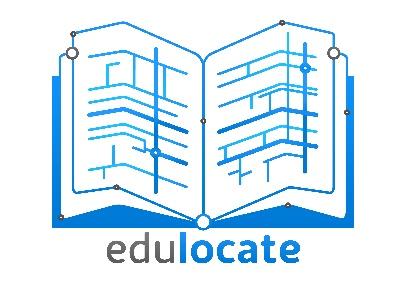
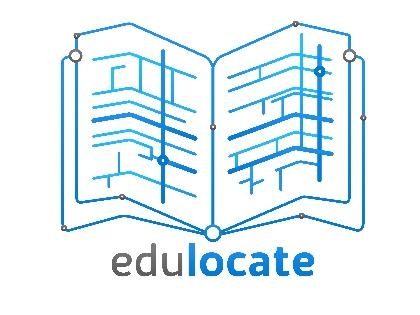
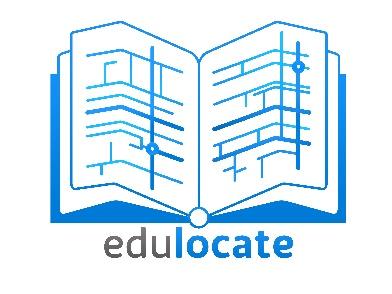
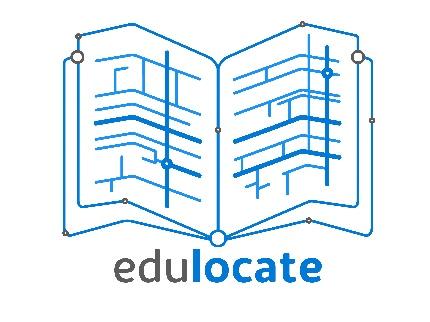
In terms of our final designs we are still deciding the layout of the other pages, however its safe to assume that they will have very similar structure and aesthetic than the two pages shown.

**Logo Development**

We also decided to design our own logo for Edulocate to make it seem more distinguishable rather than just using the Government provided Logo. The idea for the logo was to make it clear what the function of the app was. We decided to stick to a book style logo to relate back to the fact we are a school-based app, but the logo also needed to keep a technological feel.

**Logo Examples:**

****



Shown above you can see how our logo design has focused on those two aspects. It also follows our minimalism principle and making sure not too much is going on for the user. The outlined logo is the one we decided to go with in the end.

**UML diagrams & Descriptions**

Development approach

Our team decided to take an Agile approach to producing the app due to the flexibility to change features and application approach. This also meant the client could see iterations of our app when a sprint cycle had finished and give feedback for improvement for the app. To prepare for agile development we decided to create user stories to define necessary functionality in our app.

Epics

As a user, I want to be able to see all relevant schools in the area to know which to consider

* As a user, I want to be able to put my postcode in so it knows which schools are near
* As a user, I want to be able to have my location detected so that I can save time inputting it
* As a user, I want to be able to set a max distance for schools so that I only get ones within a reasonable distance
* As a user, I want to be able to set a commuting distance for schools so that I know how long it will take to/from get to school
* As a user, I want to be able to immediately ignore/exclude some schools so that religious/single sex schools aren’t brought up if they aren’t appropriate

As a user, I want to be able to see the information for a school nearby to know more about them

* As a user, I want to be able to search for specific schools if I just want to see their details
* As a user, I want to be able to see brief details about a school to assist in making a decision
* As a user, I want to be able to find social feeds from schools to find what the school is up to
* As a user, I want to be able to find news stories about nearby schools to see if there has been any major incidents or accomplishments

As a user, I want to be able to compare schools to see which is best for my child

* As a user, I want to be able to see a visual comparison between schools to make decisions without needing to compare too many statistics myself
* As a user, I want to be able to generate a score for schools based on factors important to me
* As a user, I want to be able to easily change what filters and data are in use so that I can look at a range of statistics/points
* As a more concerned user, I want to be able to generate a table of details based on any number of filters so that I can make a more thorough decision
* As a user, I want to be able to export data to a table/excel file to be able to look at it later or share

              Other user stories

* As a user, I want to have easy to change accessibility settings for icon and text size, and colour scheme
* As an admin, I want to be able to remove closed schools so that users don’t see them any more
* As an admin, I want to be able to add a new school so that users have up to date info
* As an admin, I want to be able to update a single schools information so that users aren’t given out of data information
* As an admin, I want to be able to update all data each year easily, without updating schools manually

Introduction to Design

The main approach for this system was to split the frontend and backend and enable them to communicate as a client-server relationship. Our frontend is going to be developed in a Java environment built on Android Studio, whilst our backend development is being implemented in a C# environment using ASP.NET. The database will exclusively communicate with our server, whilst requests from the application would also be handled server side. Overall the decision to split our application made it so the user would have a lightweight, streamlined application, additionally due to the added functionality of our website it meant it would be a lot more convenient to update school information server side. Our frontend designs are mostly focussed on the users experience and how they can from point A to B, through activity diagrams, whilst the backend designs explain and outline the handling of data, this includes how we standardised data and the use of services to modularise the backend. Figure 1.0 & Figure 1.1 outline the overall reach of the system.

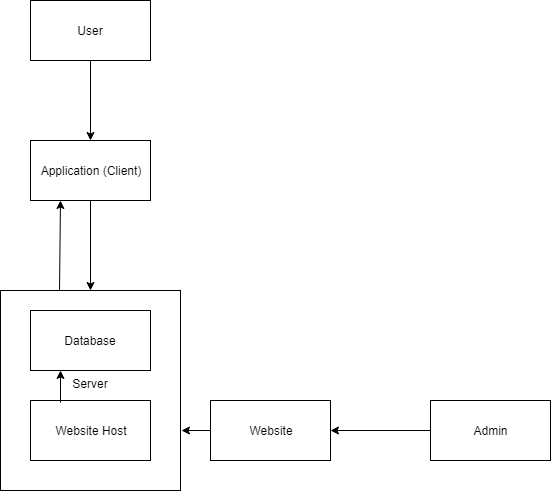


Figure 1.0

General outline of the system. The application requests information from the database through a server. The Website is also hosted on the server to allow Admins to update information about schools.

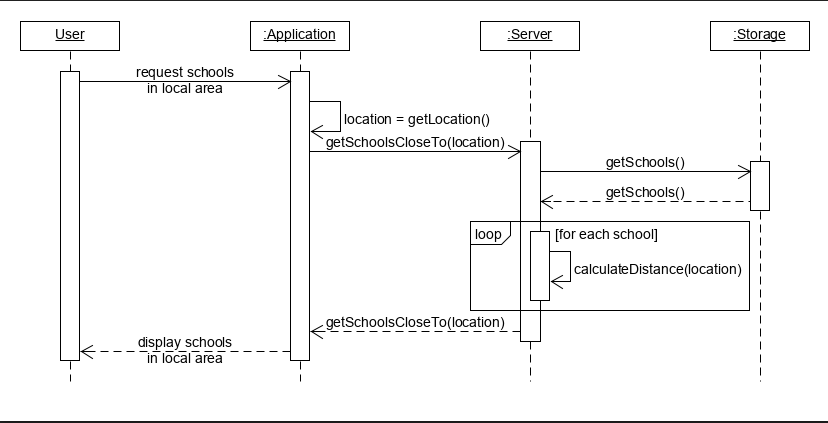


Figure 1.1

The use case diagram shows the interaction between the User, Application & Database. The Idea of this sequence diagram is to show how the individual segments of our system work together autonomously. The idea of the system is that the User only ever must see the application and nothing else, where most of the communication for requests are done through the Application and the server.

Overview of System

Client-Side view

* The client-side application was programmed in Java using Android Studio (Version 3.3.1). Assets and other GUI components are produced using various graphics programs such as Adobe photoshop.
* An API was produced to enable HTTP Standard requests to be made to the server.
* Google Maps SDK will be used to implement a basic map system. Mostly the functionality of the map’s SDK will be handling the initial search for an area and placing restrictions where the user can find a school. Our server will be able to locate nearby schools through the previously mentioned API.
* The Client-Side requirements will mostly be, in addition to other features. This includes a more effective way of showing how a school is performing. Some requirements such as showing news / twitter feeds for local schools has been shown to fairly challenging in working out the logistics of going through a huge amount of data to consistently find reliable information about news or social media outlets.

Server-Side view

* The server will use a C# environment to handle the flow of data in our system. C# will be developed using Visual Studio due to the many tools and integrations it is useful for working alongside a Database and server.
* Entity framework will be used to contain our school data and place them into the database. This will make it easier to handle our information and make populating the database easier. (Go to tooling for explanation on how the information was found).
* An endpoint will also be setup to receive requests from the client application, the transfer of information will be done in a JSON file that can be read on either end and converted into the relevant objects.
* Services are used to modularise where data is processed, reducing dependency in the system.

Website

* The website was initially not considered to provide much functionality to our project.  However, we found that having an admin website to enable updating of the database would be a feature that would make the app more functional.
* The website will be built using HTML/CSS, authentication will be done server side through some form of PHP or Javascript.
* Authentication will be implemented in the form of a basic login system. A new user will request credentials which will be approved by a system admin.
* Updating will be done through either uploading new spreadsheets with updated information of schools or a website view of the database where individual fields can be updated.

Tooling

* Specific tools had been made to get information about the schools. Using python scripts, we were able to pick out specific data points from the spreadsheets provided by the .GOV websites and placed them into a JSON file formatted to be easily accessible by our scripts.
* Additionally, we used a python script to attempt to find social media handles for the schools through their respective websites.

GitHub

* GitHub was our only source of version control, we felt like it was more accessible and easier to teach than GitLab.
* GitHub Desktop or Git Bash was used to create local copies of code and push to branches when necessary.

Client activity view

The frontend UML mostly consist of Activity diagrams that show how the user will directly interact with the application. This means that none of the backend processes are show as this is only what the user will see. The idea of this layout is to make it obvious how the app will be used and how aspects of our frontend interlink to make the app seem more autonomous. The application was split into three main screens that held the most importance which are the home screen, map screen & the comparison screen. We decided to focus our designs and application to streamline the usability, which we thought would make the experience easier for the user. Overall this also made the activity diagrams quite modular as there are clear boundaries of which screen the user will be on.

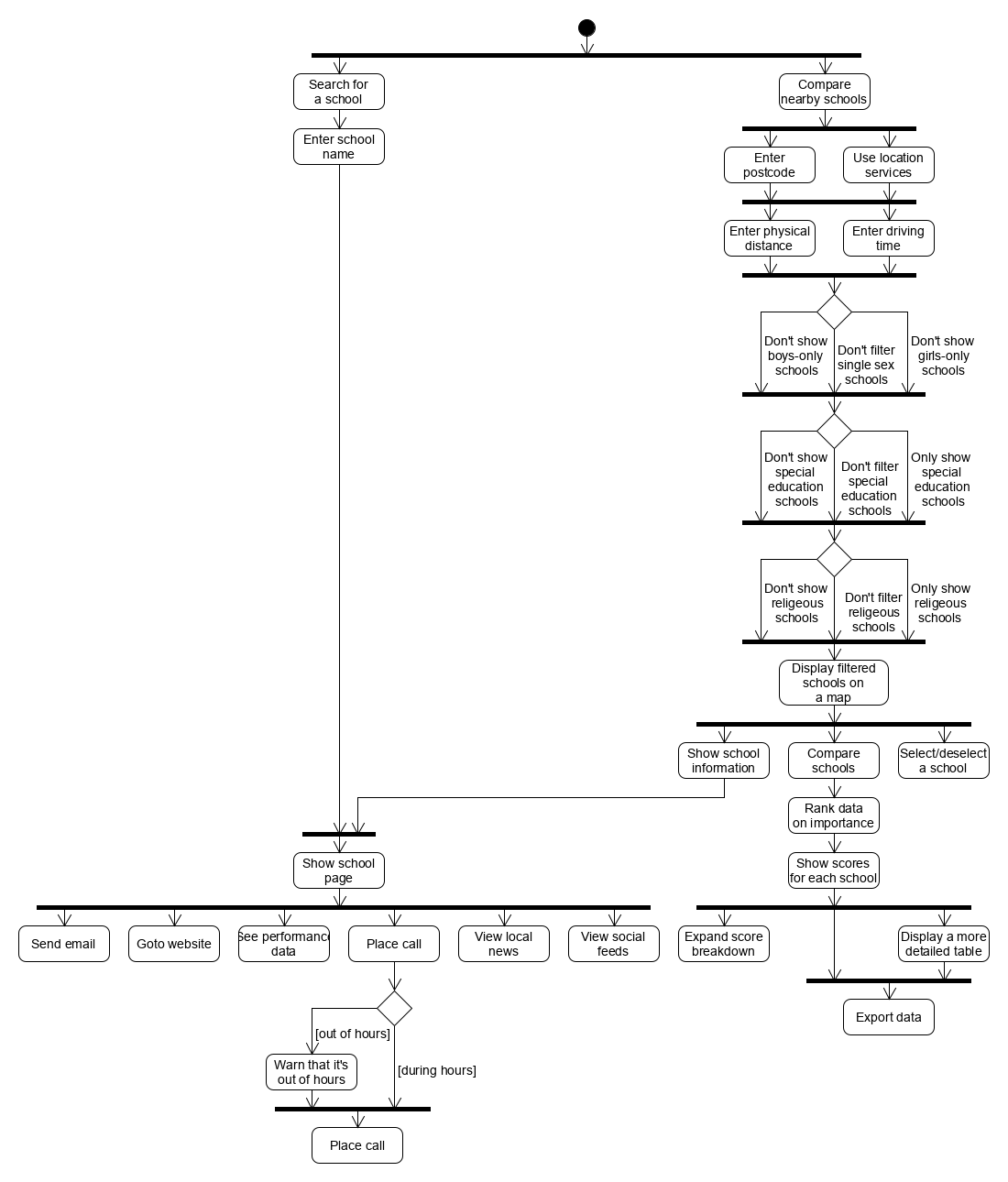
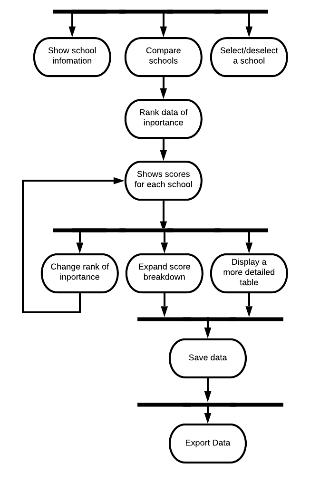


Figure 1.2

This activity UML describes a general experience a user would have when navigating Edulocate. It shows the different decisions the user can take and what it will result in. This is much different to figure 1.1 that shows the full extent of our system. This UML alternatively describes a basic interaction between user and the application.

  
  
  
  
  
  
  
Figure 1.3

This Activity UML is a continuation of Figure 1.2 and extends on the user interaction for comparing schools. In addition to the previous UML is shows specifically the comparison segment of our app.

Figure 1.2 and Figure 1.3 summarises the initial design we were trying to achieve for the application. However due to lack of features and details we decided to revisit the activity diagram to show a more realistic overview of the frontend system and what user should see (see figure 1.5).  Additionally, a general use case UML was produced to show key information the user should be able to access through the features on our app (see Figure 1.4).

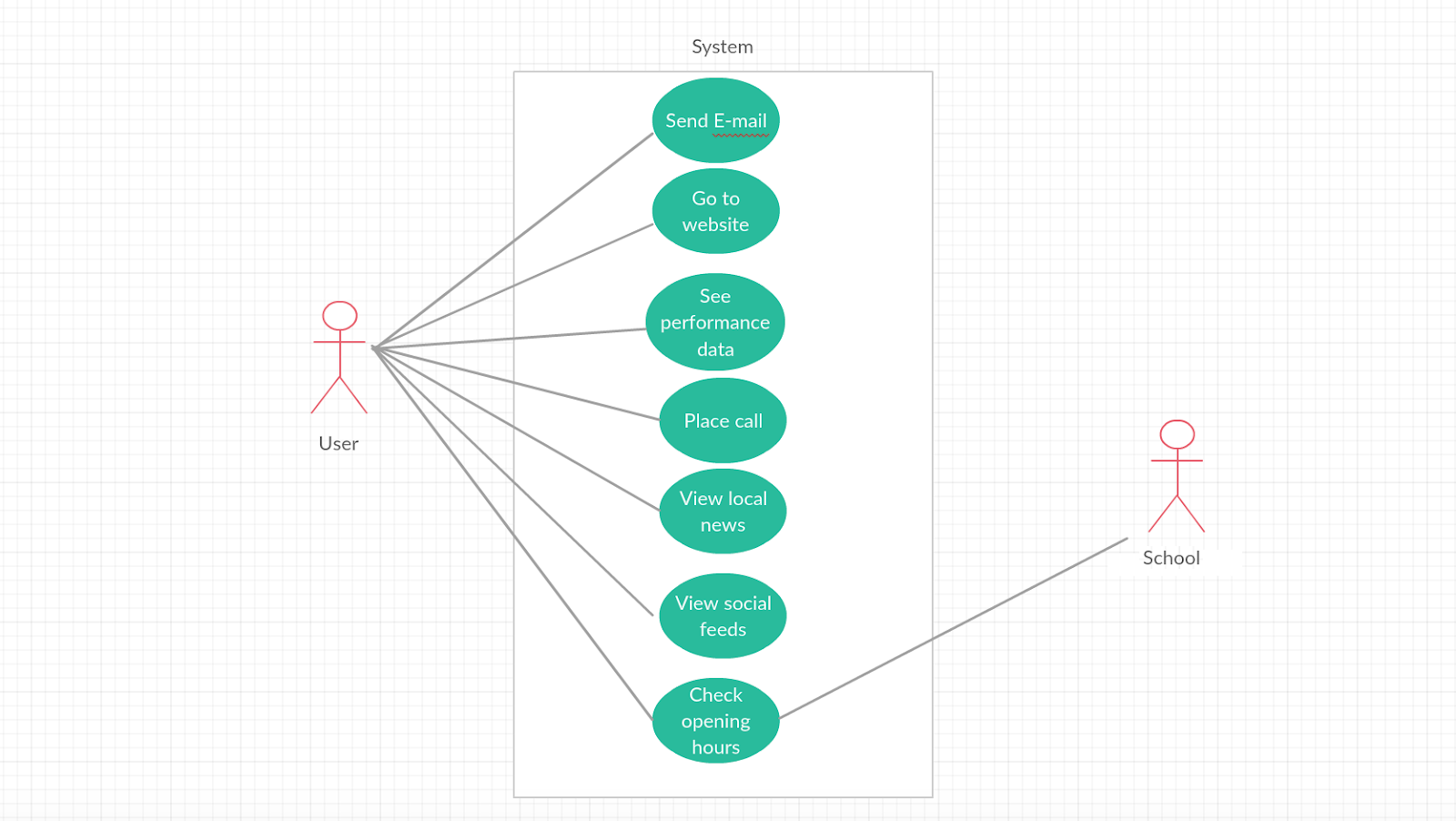
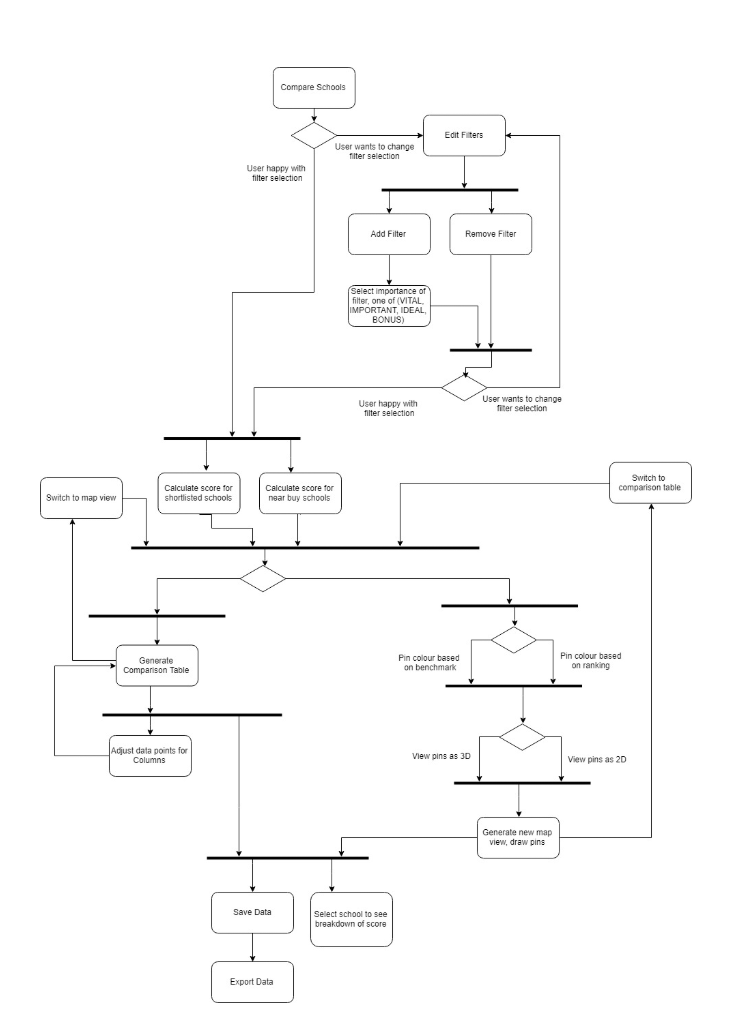
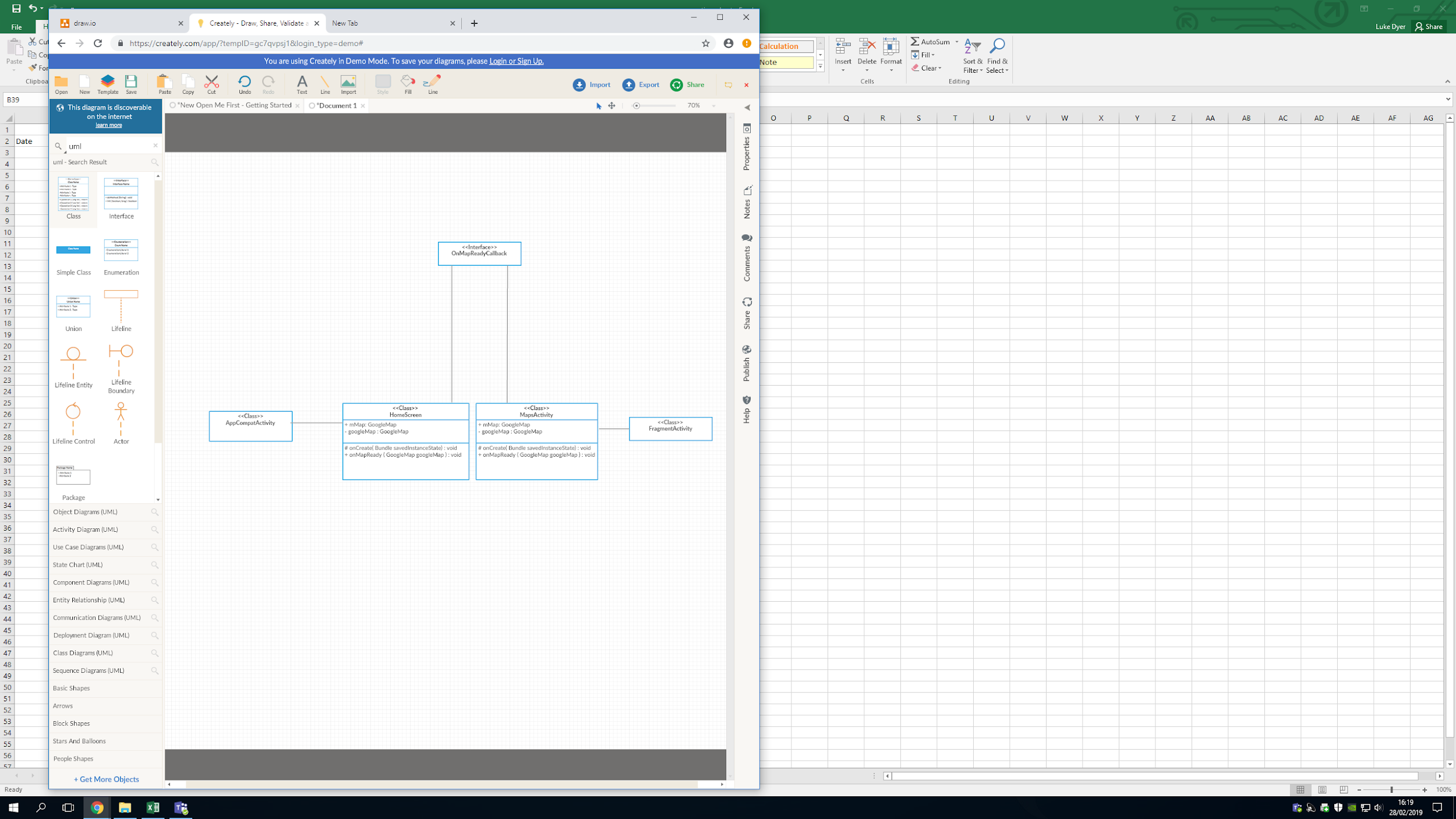


Figure 1.4

This use case shows what the user should be able to access from the system. More specific information like opening hours etc. Should be accessible from the Schools website itself.

Figure 1.5

This is our finalised Activity UML that shows an even broader outlook on the app including specially defined features that we decided to add. Overall this activity diagram is a more realistic and accurate overview of our system.



**<<Class>>**

**SDK**

This front-end class diagram shows the interaction between the google maps SDK & API with our application. onCreate() initialises the map and is called when the activity is called. The map is then is then ready to receive input. The idea is that only one of the activities (pages) in our app can be active at once therefore we can add flags to change behaviour depending if an activity is going to be stopped or paused, this includes if the activity is ready such as the onMapReady method. A lot of the detailed implementation will be missing due to progression of our frontend development; however, a lot of the key functions are still mapped such as our SDK to communicate with the backend API to make requests. Furthermore, some basic functionalities like drawing a circle over the map and calculating score has been included as they can be considered core features that a user might want.

Backend architecture & integration

As discussed, our backend is built in a C# environment, using entity framework to handle data for the database. The idea for the backend is too split functionality into various services that can be requested at specific times during runtime of the app. These services will have specific functionality which will enable for the backend system to work independently if certain services went down. Additionally, the modularity introduced from services allows the backend environment to be expanded and modified a lot easier has dependency is reduced in the overall system.  Figure 2.0 shows the outline of the system architecture.

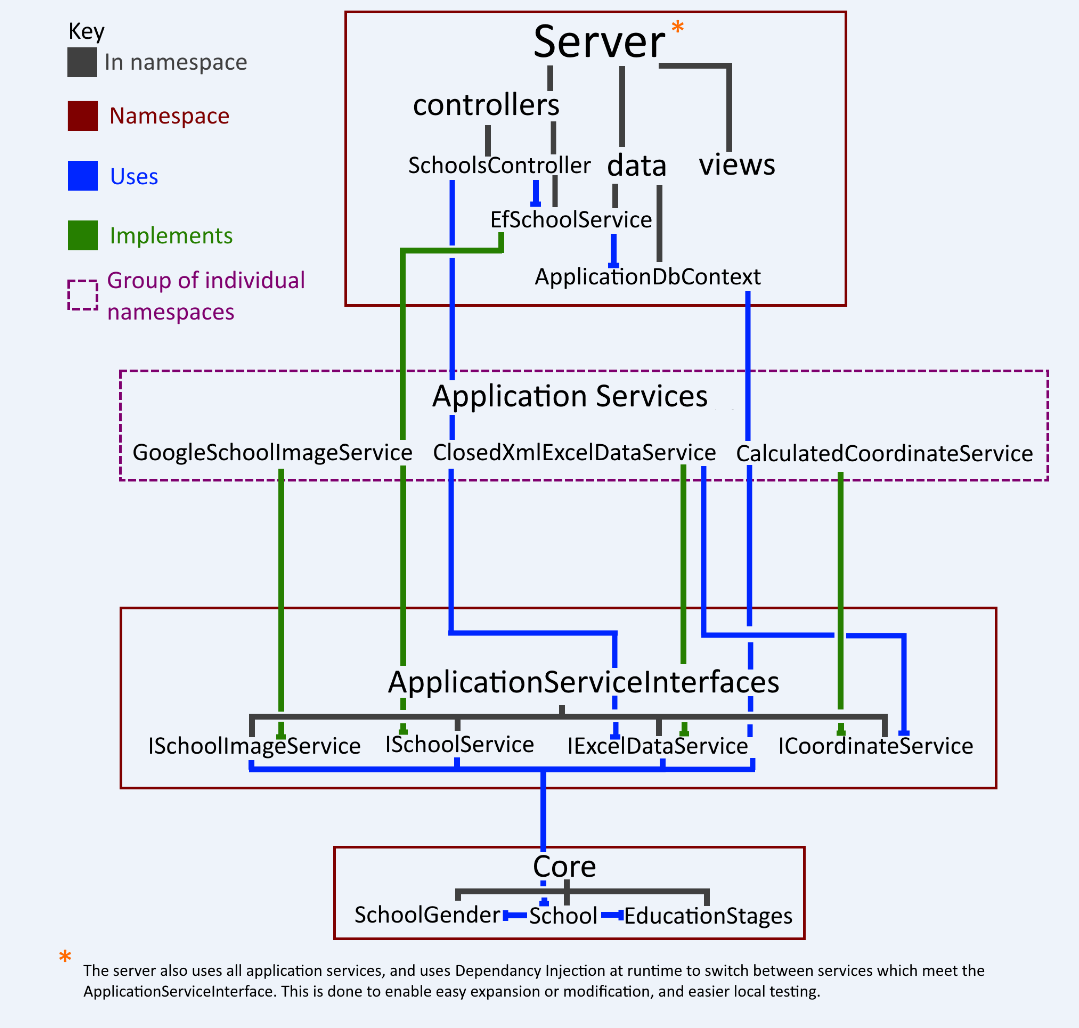


Figure 2.0

This architecture UML shows how our back-end will function independently without any interaction with our front-end application. In general, it shows what services will be provided for the front-end in terms of requesting information about schools and figuring out what schools are local to given position of the user.

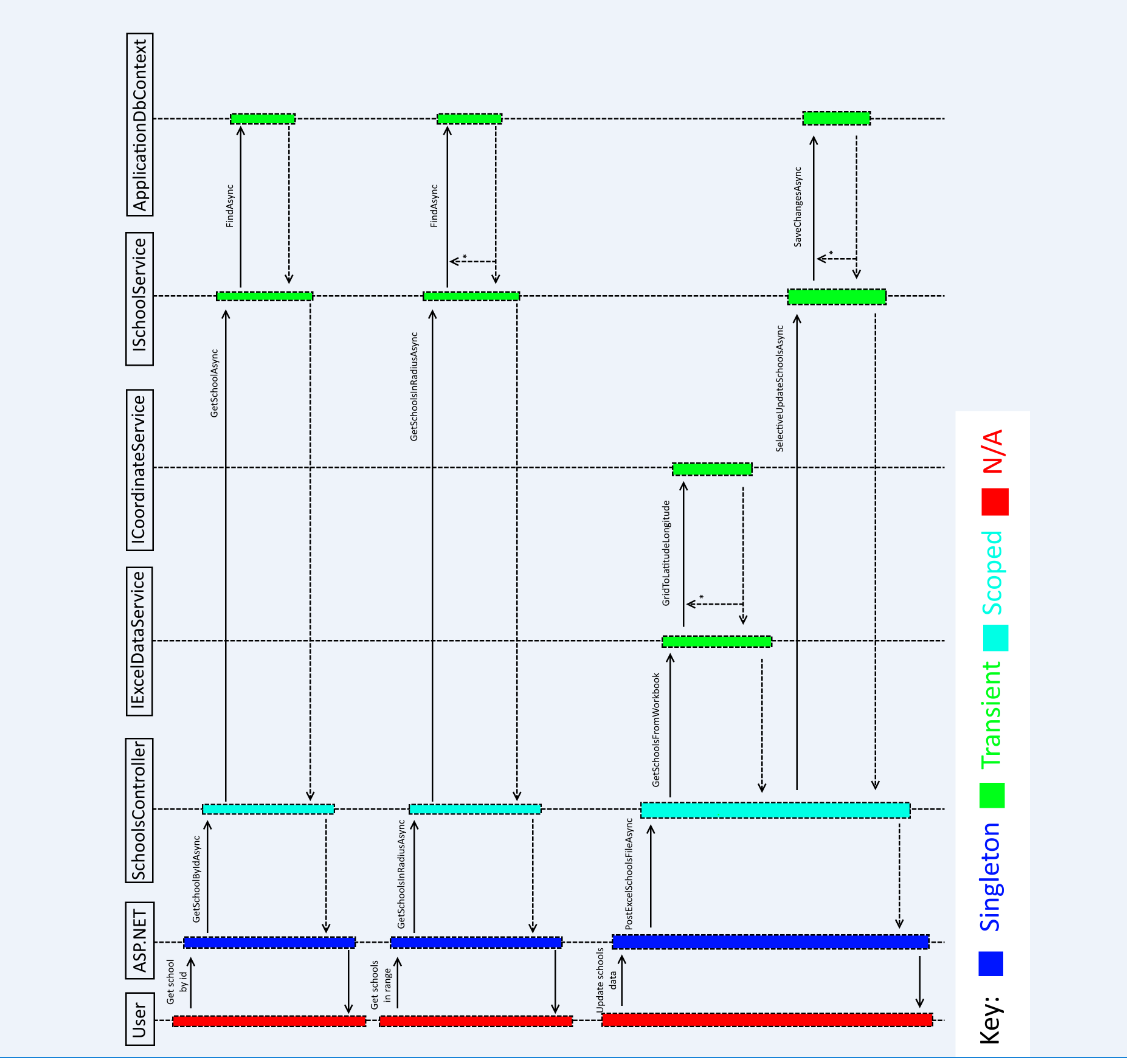
For requests to be made in on our frontend system an API was produced to standardise the information going in and out of the system (this was done using HTTP standards).  This reduces the chance of errors during requests as standardisation of data makes it easier to prevent and find problems, but also this makes programming for requests a lot simpler has we only have to consider data in one normalised form. Figure 2.1 shows this additional layer of communications and how a request would be handled by the backend.

Figure 2.1

This Sequence UML shows how a request would be handled depending on what is needed by the application. It shows the use of each service and how requests can be handled independently in different services.

Specification Contribution Matrix Team Number 14

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Section | Sachin | Sam | Alex | Arman | Luke | Charlie | Daniel | Matthew J | Matthew T E |
| Purpose (Executive summary) | R | R | R | R | R | R | R | C.R | R |
| Background Research & Analysis | R | R | R | R | R | R | C,R,M | R | C,M,R |
| Project Purpose | R | R | R | R | R | R | C,R | R | R |
| Roles & Deliverables | R | R | R | R | R | R | R | C,R | C,R |
| Project Plan | C,R | R | R | R | R | R | R | R | C,R |
| Hardware & Software | R | R | R | R | R | R | C,R | R | R |
| References | R | R | R | R | R | R | R | R | C,R |
| Definition of Terms | C,R | R | R | R | R | R | R | R | C,R |
| Solution Requirements | R | R | R | R | R | R | C,R | R | R |
| Constraints & Dependencies | R,M | C,R | R | R | R | R | R | C,R | R |
| Software Design | C,R,M,T | R | C,R | C,R | C,R | C,R | R | C,R | R |

Key

C = Create

M = Modify

R = Review

T = Test