2021

<student name/s>

<student number/s>

8/11/2021

<your application name>



Replace image with one with some relevance to your application here

CAB432 Assignment 1

Contents

[Mashup Purpose & description 2](#_Toc79613994)

[Services used 2](#_Toc79613995)

[Twitter Standard Search API (v.1.1) 2](#_Toc79613996)

[API 2 2](#_Toc79613997)

[API n 2](#_Toc79613998)

[Mashup Use Cases and Services 2](#_Toc79613999)

[Restaurant Search in a Foreign City 2](#_Toc79614000)

[US 2 3](#_Toc79614001)

[US n 3](#_Toc79614002)

[Technical breakdown 3](#_Toc79614003)

[Architecture and Data Flow 3](#_Toc79614004)

[Deployment and the Use of Docker 4](#_Toc79614005)

[Test plan 4](#_Toc79614006)

[Difficulties / Exclusions / unresolved & persistent errors 4](#_Toc79614007)

[Extensions (Optional) 5](#_Toc79614008)

[User guide 5](#_Toc79614009)

[References 5](#_Toc79614010)

[Appendices 5](#_Toc79614011)

*This is a template for your assignment report. It is not compulsory to use it, but it will probably save a lot of effort if you do. Some of these sections may not be very long, but you should make sure that you cover the key sections describing the application functionality and implementation.*

*Throughout this document, you should assume that black text in italics is there as guidance and you should read it, follow the instructions and then* ***delete it*** *when you have entered your own text. Some examples are not italicized, but should obviously be replaced by your own material.*

*The sections are based on those mentioned in the spec, but some of them have been grouped together under Technical Description for convenience. It should be straightforward to copy and paste any existing material into this template.*

*This report should be around 10-15 pages including screenshots, but this is a guide only – we will not be enforcing a page limit or marking you down for submitting something with 16 pages instead. But be sensible, we really don’t want something that is 25 pages or more.*

*[Our thanks to the students who allowed us to use images from earlier reports as examples here and in the template for Assignment 2.]*

Introduction

### Mashup Purpose & description

*This is written in a high-level professional tone. Tell is in about a paragraph or so what the app is supposed to do. Explain to us the need for your app and how it provides something that is novel. If there is something especially amazing about your app, tell us briefly what to look for. At this point you can show 1-2 basic screenshots of your application to illustrate the approach, but leave the more detailed screenshotting to the use cases below.*

### Services used

*Overview of what API endpoints are used & what they are providing – data sources? Analysis? Here we really do just want a link and a brief description of the API and the services it provides. So, if I used something like Twitter, I would go to their API docs, and I would include something like the first entry below, and you should repeat this for the others used.*

#### Twitter Standard Search API (v.1.1)

Returns a collection of relevant Tweets matching a specified query – may also be filtered based on popularity or geocoding [and whatever other obvious details we might decide to include]

Endpoint: https://api.twitter.com/1.1/search/tweets.json

Docs: <https://developer.twitter.com/en/docs/twitter-api/v1/tweets/search/api-reference/get-search-tweets>

#### API 2

*More here…*

#### API n

*And similarly here…*

## Mashup Use Cases and Services

*Your User Stories should go here. The basic structure is provided, and you should fill in the role and then the action and the good result that follows. Underneath the formal statement of the User Story, you can then tell us how you have implemented this service – basically you would tell us how you get input from the user and then use this to get information from an API and then use those results in another one. This is at a semi-technical level – introduce it at a high level and then give more detail, but stop well short of code excerpts. You should then use screenshots to illustrate the process*

*To illustrate some of what we want, I will make up an example that might not make much sense in the COVID era:*

#### Restaurant Search in a Foreign City

|  |  |
| --- | --- |
| As an | International tourist |
| I want | To find a list of restaurants with my favourite cuisine |
| So that | I can have a good dining experience in a foreign city |

*The example is pretty badly worded, but it will do for now. I would then explain how I might search for a particular type of food in the foreign city and how I might then get their menus and use a translation app (in practice, this might be spread over a couple of user stories, but this is the sort of level that we are after – you would name the APIs, say that you were using Google translate, for example – and then talk about the results. You can expand a bit on the technical details, but the main goal is to show us how it works and the screenshots and your explanations linking them are the most important thing. Here I am not showing images, but when you include them each use case will push out toward a page.*

#### US 2

|  |  |
| --- | --- |
| As a | <next use case here> |
| I want |  |
| So that |  |

#### US n

|  |  |
| --- | --- |
| As a | <next use case here> |
| I want |  |
| So that |  |

## Technical breakdown

*This is a deeper discussion of the architecture, the technology used in creating the mashup, any issues encountered, and overall, how you implemented the project.*

### Architecture and Data Flow

*Explain how your system operates, making it clear how data flows around the system through requests and responses. You are describing the overall architecture of your application at a source code level. The description above tells us something of the application’s use. Now we want to see how that maps to the code organization – show us how your code is organized and tell us how you have split the responsibilities. We should get some sense of how the application works and how the data and control flows around. You may also find it helpful to show us screen grabs of code if that makes your points clearer. Tell us anything you think we need to know about how you have structured the application and made it work, but there also a section below to describe problems.*

*In Assignment 2 we provide you with a number of example diagrams that you may use to show how the application is structured. Most of these are optional even in assignment 2, but you might find the process flow diagram – especially as shown on the left hand side below – helpful in telling us how your application works. Here you may use a similar structure to show the transition between screens and service usage.*

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

### Deployment and the Use of Docker

*This section will be relatively brief. You should discuss – in a paragraph or two, usually – how you used Docker and any particular configuration choices you want to tell us about. You may include excerpts from the Dockerfile and list this file as an appendix to the report*

### Test plan

*Manual testing is fine and our expectations are in line with the example grid below. You can show the results through a screen shot and point us to these from the table. If you haven’t used a test plan like this before, then please clarify it with your tutor. We would in general expect 10-15 test cases or so depending on the application.*

*Your tests should include*

* *Positive outcome cases*
* *Negative outcome cases (error scenarios)*
* *Edge cases*
* *Non-functional cases (ideally, but not required this time).*

*Note that the grid below is unrelated to this application.*



Difficulties / Exclusions / unresolved & persistent errors /

*In this section, you should explain anything that caused you problems and how you overcame those problems. Tell us if there was any issue that prevented you completing the assignment to specification. Tell us about any assumptions or compromises that you have made. Those who worked with an API like Spotify, which presents particular concerns, should discuss the compromises here, and this is also where you can tell us about problems with API keys and responses.*

*More generally, you might consider:*

* *Your major roadblocks and how you resolved them.*
* *Any functionality you didn’t or couldn’t finish*
* *Are there any differences between your brief and what you delivered? If so, explain why.*
* *Are there any outstanding bugs?*

## Extensions (Optional)

*In this section, you can tell us if you wish to how you might extend your app and make it better. This is an opportunity to tell us about good ideas that you had that you didn’t have time to tell us about.*

## User guide

*Tell us how to use your application. You may re-use some of the screenshots from the use case descriptions, but this is more about how to use the app. As long as we can find what we need to do to use your application, this need not be all that long.*

*But either way, screenshots are your friend.*

## References

*Use a standard approach to referencing – see the guidance at* [*https://www.citewrite.qut.edu.au/cite/*](https://www.citewrite.qut.edu.au/cite/)*.*

## Appendices

*Stuff you want to include, but is too long or too complex to include in the main report text. The full Docker file, some longer excerpt from API docs. Whatever helps.*