

CH'EATS

A new way to use food delivery services

THCH

Cameron Baker

Hayes Choy

Haodong Mo

Thomas Sinn

TABLE OF CONTENTS

ABSTRACT	3
OUR SOLUTION	4
MAIN FEATURE	4
ADDITIONAL FEATURES	6
USER STORIES	6
SOFTWARE ARCHITECTURE AND IMPLEMENTATION DECISIONS	16
FINAL SOFTWARE ARCHITECTURE	16
SEQUENCE DIAGRAMS	17
DESIGN DECISIONS	20
SUMMARY	24
TEAM REFLECTION	25
RESPONSIBILITIES AND ORGANISATION OF THE TEAM	25
ISSUES AND LIMITATIONS	25
PERSONAL REFLECTIONS - How did the project go in your opinion? Would you do it any differently now?	26
Thomas	26
Cameron	27
Haodong	29
Hayes	29

ABSTRACT

The recent surge of delivery food services has been backed by digital advancements throughout the past decade. Worldwide, the market for food delivery stands at \$90 billion USD, or 1 percent of the total food market and 4 percent of food sold through restaurants and fast-food chains¹. The market has already matured in most countries, with Australia's market being dominated mostly by Uber Eats, Deliveroo, DoorDash and Menulog.

These businesses capitalise on the increased convenience and opportunities that come with smartphones, now a staple in our everyday lives. On just a single app, a customer can order a meal, side and drink from multiple restaurants and then have it delivered to their doorstep within the hour. However, there are many apps that do this and this competition among businesses makes it difficult to see which service is cheaper. The price of the same item from the same restaurant can vary up to 20%² across different apps, each having their own service fee, delivery fee and promotions.

We aim to create a website that solves this problem. We want to make the pricing among food delivery services transparent so that users can choose the most affordable option. This will be achieved by showing users the cost of delivery for each service for each particular meal on our website.

Our product could not have come at a more relevant time. With all restaurants being forced to shutdown or transition to takeaway or delivery, there has naturally been an increased demand for these services. Uber CEO Dara Khosrowshahi has reported a tenfold increase in the number of self-serve signups by restaurants between March 12 and March 19³ as well as a 30% surge in customer sign ups⁴. Menulog has likewise reported “a high volume” of requests to join its platform⁵. With the successful launch of our product, users can continue to support local businesses in the most affordable way.

¹

<https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/the-changing-market-for-food-delivery> (November 2016)

² <https://techcrunch.com/2020/03/16/the-hidden-cost-of-food-delivery/> (March 2020)

³ <https://au.finance.yahoo.com/news/uber-eats-growth-coronavirus-delivery-140101683.html> (March 2020)

⁴

<https://www.forbes.com/sites/marcochiappetta/2020/03/25/uber-eats-demand-soars-due-to-covid-19-crisis/#168b586b580c> (March 2020)

⁵ <https://www.businessinsider.com.au/delivery-restaurants-coronavirus-covid-2020-3> (March 2020)

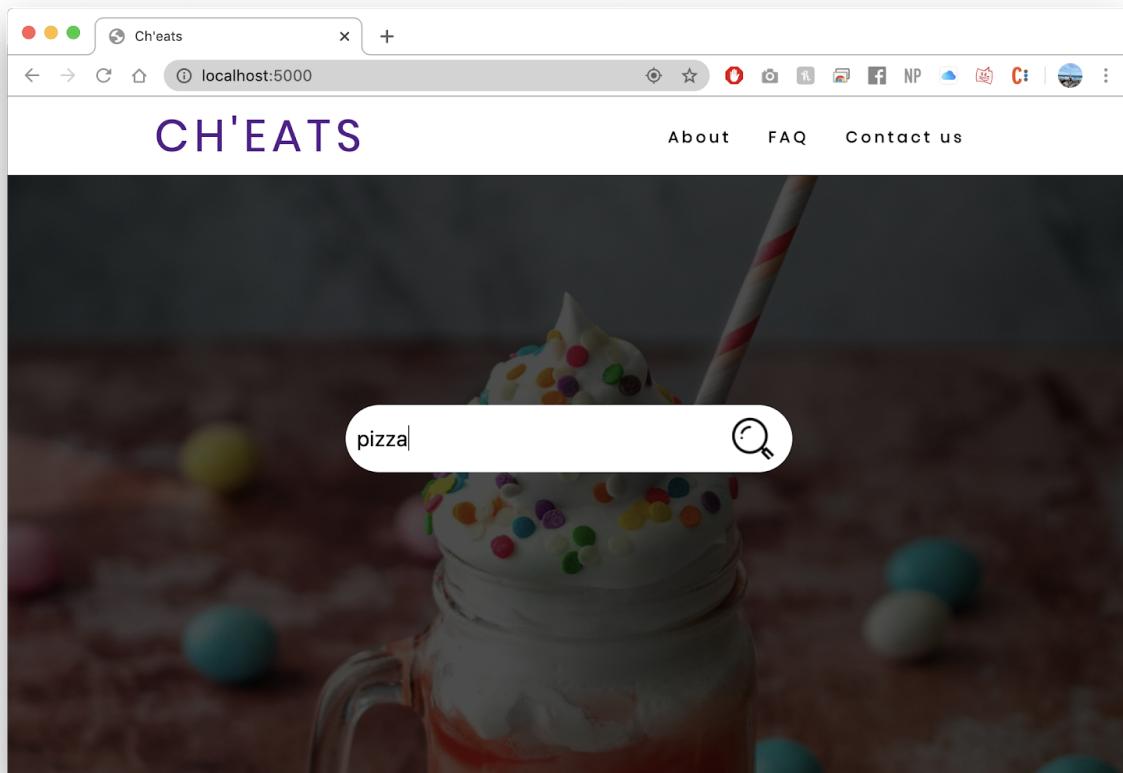
We also wanted foreign users such as tourists and international students to be able to use our website, so we accommodated them by allowing them to convert original prices into foreign currency and text into their own language. This feature allows the exclusive users of EASI (a chinese food delivery application) to transition to using Uber Eats, Menulog, Doordash or Deliveroo and thus increasing the user base.

OUR SOLUTION

When customers decide to order food online, they're faced with multiple decisions - what type of food they want, which restaurant to order from and which menu items to choose. Our solution empowers users to make these decisions as well as an overview of prices across food delivery services so that they can choose which service to order from.

MAIN FEATURE

This process consists of three webpages. On the landing page, users search for what kind of food they want.



The next page displays a list of restaurants that satisfy the search query.

Turramura Pizza Gourmet	Pizza Hut Turramurra	Crust Pizza	Domino's Pizza Pymble
Distance: 29.8 km rating: 4.2	Distance: 29.7 km rating: 4.1	Distance: 28.9 km rating: 3.8	Distance: 27.5 km rating: 3

Pymble Gourmet Pizza Pasta & Ribs	Domino's Pizza Turramurra	Turramura Pizza Gourmet	Pizza Hut Turramurra
Distance: 27.2 km	Distance: 28.8 km	Distance: 29.8 km rating: 4.2	Distance: 29.7 km rating: 4.1

Clicking on a restaurant will bring the user to the restaurant page, where its prices for each menu item can be compared across each food delivery service.

	Uber Eats	DOORDASH	MENULOG	deliveroo
Northern Style Dumplings	7.03	8.61	2.31	17.44
Southern Style Dumplings	19.19	11.89	13.85	4.25
Chicken Fried Rice	14.38	1.65	13.42	15.96
sushi	11.6	11.07	3.38	17.13

ADDITIONAL FEATURES

Changing currency

To accommodate for foreign users, users can convert the prices shown to another currency to get a relative sense of how much they are spending in terms of their native currency.

Reviews

Users can see review ratings for each restaurant to see how popular their food is. This can assist users in trying something new or avoiding restaurants with low ratings, which would imply a lower quality meal. Ratings are obtained from the **Google Places API**.

Recently viewed

Users can see their recently viewed restaurants on the landing page for quick access if they want to order again.

USER STORIES

These features are explored in detail in the following user stories:

FEATURE: Searching for restaurants that offer a type of food

AS A hungry user **I WANT TO** search for a type of food **SO THAT** I can see which restaurants offer this type of food.

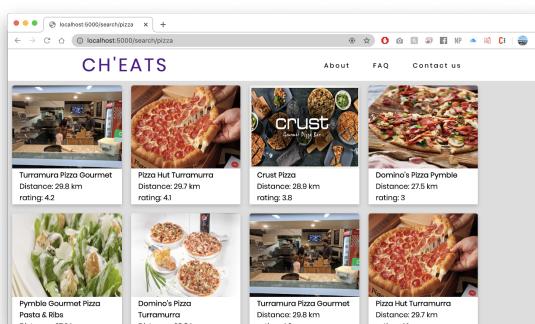
SCENARIO: Search for restaurants that offer pizza.

GIVEN I am on the landing page

WHEN I type “pizza” into the search bar

AND I press the *Search* button

THEN I should see all restaurants that offer pizza that support food delivery



FEATURE: Searching for restaurants

AS A hungry user **I WANT TO** search for a restaurant **SO THAT** I can see if I can order from that restaurant

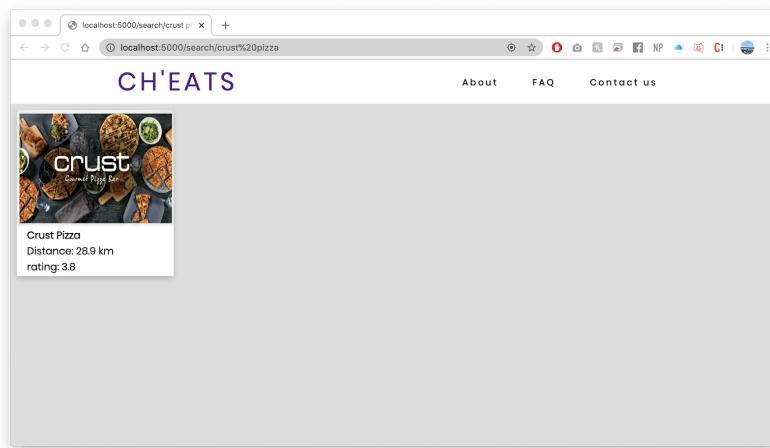
SCENARIO: Search for *Crust Pizza*

GIVEN I am on the landing page

WHEN I type “crust pizza” into the search bar

AND I press the *Search* button

THEN I should see *Crust Pizza*.



FEATURE: Scrollable results page

AS A hungry user **I WANT TO** be able to scroll down the results page **SO THAT** I can see the breadth of my options before I choose the restaurant I want to order from.

SCENARIO: Searching for “pizza” restaurants

GIVEN I am on the landing page

WHEN I type “pizza” into the search bar

AND I press the *Search* button

THEN I should see all restaurants that offer pizza

AND when I should be able to scroll down to see more results

FEATURE: Displaying review ratings of restaurants

AS A hungry user **I WANT TO** see the review rating of restaurants **SO THAT** I can see which restaurants are popular

AS A hungry user wanting to try eating something new **I WANT TO** check a restaurant's reviews **SO THAT** I can see if I want to try eating from this restaurant

SCENARIO: Looking at the review ratings for "pizza" restaurants

GIVEN I have searched for "pizza" restaurants

WHEN I load the result page

THEN I should see review ratings next to all the listed restaurants



FEATURE: Displaying how far the restaurant is away from the user

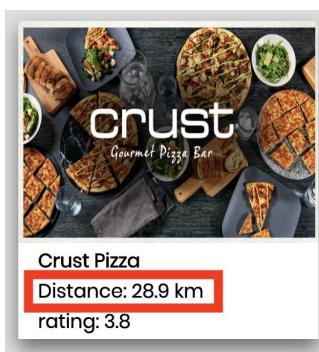
AS A hungry user **I WANT TO** see how far a restaurant is **SO THAT** I can see which restaurants would attract higher delivery fees and have higher delivery times

SCENARIO: Looking at the distances from the user for "pizza" restaurants

GIVEN I have searched for "pizza" restaurants

WHEN I load the result page

THEN I should see the real-life distance between the restaurant and I in kilometres



FEATURE: Filters out restaurants that cannot be ordered from

AS A hungry user **I WANT TO** see only restaurants that I can order from **SO THAT** I will not choose a restaurant that is too far away that will not deliver to me.

SCENARIO: Looking for nearby “pizza” restaurants

GIVEN I have searched for “pizza” restaurants

WHEN I load the result page

THEN I should only see “pizza” restaurants that I can order from within my vicinity

FEATURE: Display menu items

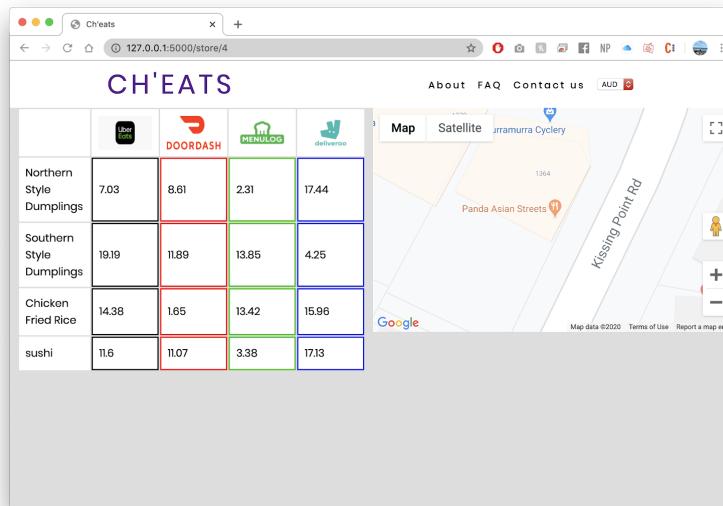
AS A hungry user **I WANT TO** see a restaurant’s menu **SO THAT** I can see what meals I could order from this restaurant

SCENARIO: Showing menu items for *Panda Asian Streets*

GIVEN I have found the *Panda Asian Streets* restaurant in the results page

WHEN I click on the restaurant

THEN I should see the whole menu of *Panda Asian Streets* on its restaurant page



FEATURE: Price comparison of a particular restaurant's menu item

AS A hungry user **I WANT TO** compare prices of menu items of a particular restaurant across multiple food delivery services **SO THAT** I can choose the most affordable way to order the meal I want

SCENARIO: Comparing prices of *Panda Asian Streets* menu items

GIVEN I have found the *Panda Asian Streets* restaurant in the results page

WHEN I click on the restaurant

THEN I should see the prices for each menu item offered by *Panda Asian Streets* for each food delivery service

The screenshot shows a web browser window titled 'CH'EATS' with the URL '127.0.0.1:5000/store/4'. The main content is a table comparing prices for four menu items across four delivery services: DOORDASH, MEETUP, GRUBHUB, and Delivery. The table has columns for the menu item name and four delivery service columns. The map on the right shows the location of 'Panda Asian Streets' at 1564 Kissling Point Rd, Jatramura Cycery.

Item	DOORDASH	MEETUP	GRUBHUB	Delivery
Northern Style Dumplings	7.03	8.61	2.31	17.44
Southern Style Dumplings	10.19	11.89	13.85	4.25
Chicken Fried Rice	14.38	16.65	13.42	15.96
sushi	11.6	11.07	3.38	17.13

FEATURE: Map showing restaurant's location

AS A hungry user **I WANT TO** see the restaurant's location **SO THAT** I can gauge a rough estimate of delivery costs and time.

SCENARIO: Finding where "Crust Pizza" is located.

GIVEN I have found the *Crust Pizza* restaurant in the results page

WHEN I click on the restaurant

THEN I should see a map showing me where *Crust Pizza* is located.



FEATURE: Link to food delivery service's website

AS A hungry user **I WANT TO** easily go to the food delivery service's website **SO THAT I** can make my order once I've been recommended the cheapest food delivery service.

SCENARIO: Ordering a Pepperoni Pizza from *Crust Pizza* off *Menulog*

GIVEN I am on the *Crust Pizza* restaurant page

AND I have found that *Menulog* offers the cheapest Pepperoni Pizza compared to the other food delivery services

WHEN I click on *Menulog* logo

THEN I should be redirected to the *Menulog* website so that I can order my Pepperoni Pizza



FEATURE: Show recently viewed restaurants

AS A returning user **I WANT TO** quickly return to restaurants I have looked at before **SO THAT** I can make my order again quickly and efficiently

SCENARIO: A returning user who has landed on the home page

GIVEN I am on the landing page

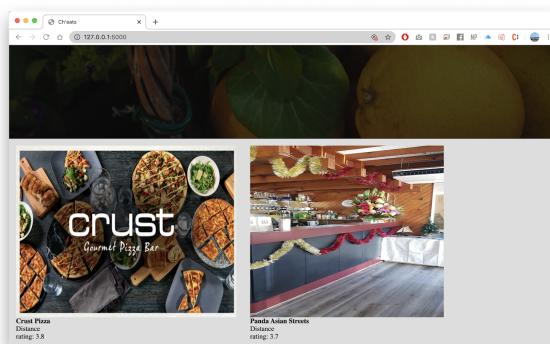
AND I have viewed a restaurant before

WHEN I scroll down from the landing page

THEN I should see all the restaurants I've viewed before

WHEN I click on a card

THEN I will be redirected to that restaurant's page



FEATURE: Access to FAQ

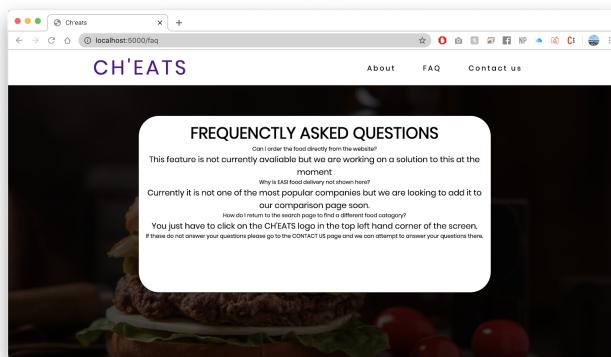
AS A new user **I WANT TO** read a *Frequently Asked Questions* page **SO THAT** I can understand how to use the page better

SCENARIO: A user is confused about how to start using the website

GIVEN I am on any page

WHEN I click on the *FAQ* link in the navigation bar

THEN I should be able to read FAQs to better understand the website



FEATURE: Ability to contact developers

AS A user who wants to report an issue **I WANT TO** contact the developers **SO THAT** I can notify them of an issue or bug to facilitate the improvement of the product

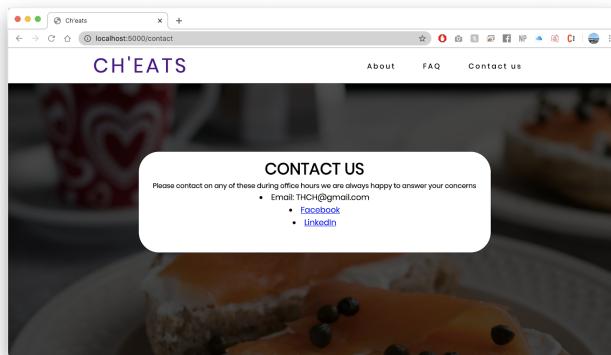
SCENARIO: A user has found a minor bug (such as a misaligned icon)

GIVEN I have an issue to report

AND I am on any page

WHEN I click on the *Contact Us* link in the navigation bar

THEN I should see possible ways to contact the developers via email or social media.



FEATURE: Ability to learn about the website's purpose

AS A new user who has not been informed what the website is for **I WANT TO** read about the purpose of the page **SO THAT** I can determine if this page is of any use to me.

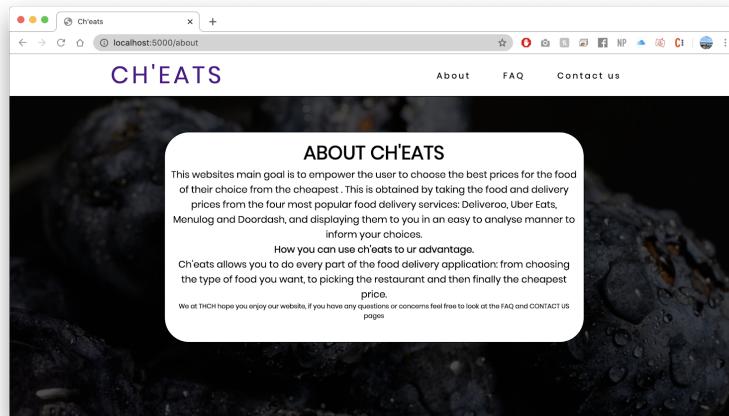
SCENARIO: A new user has come across *Ch'eats* for the first time

GIVEN I am visiting the website for the first time

AND I am on any page

WHEN I click on the *About* link in the navigation bar

THEN I should be able to read purpose *Ch'eats* serves to fulfil



FEATURE: Reset search

AS A user who no longer wants to eat the type of food I searched for **I WANT TO** go back to the landing page **SO THAT** I can search for something different

SCENARIO: A user has just searched for “pizza” restaurants but has changed their mind and wants sushi

GIVEN I am on the “pizza” results page

WHEN I click on the *Ch'eats* logo in the navigation bar

THEN I should be brought back to the landing page

WHEN I type “sushi” into the search bar

AND I press the *Search* button

THEN I should be shown all restaurants that offer sushi

FEATURE: Convert currencies into another currency

AS A foreign user who is accustomed to using another currency I WANT TO convert the current costs of the menu items into my native currency SO THAT I can realise how much I am actually spending

SCENARIO: A user who just flew in from Germany currently in Sydney wants to order from *Panda Asian Streets*

GIVEN I am on the *Panda Asian Streets* restaurant page

WHEN I click on *Panda Asian Streets*

AND select EUR

THEN I all costs on the menu should be converted to EUR

The screenshot shows a web browser window with the URL `127.0.0.1:5000/store/4`. The page title is "CH'EATS". At the top right, there are links for "About", "FAQ", "Contact us", and a dropdown menu set to "EUR". Below the header is a map of a street area with a location pin for "Panda Asian Streets". To the left of the map is a 4x4 grid table representing a menu. The columns are labeled with delivery platforms: "Liber Eats" (black), "DOORDASH" (red), "MENULOG" (green), and "deliveroo" (blue). The rows represent different menu items: "Northern Style Dumplings", "Southern Style Dumplings", "Chicken Fried Rice", and "sushi". Each cell in the grid contains a price value. The prices are color-coded by delivery platform: black for Liber Eats, red for DoorDash, green for Menulog, and blue for Deliveroo.

	Liber Eats	DOORDASH	MENULOG	deliveroo
Northern Style Dumplings	4.15	5.08	1.36	10.29
Southern Style Dumplings	11.33	7.02	8.17	2.51
Chicken Fried Rice	8.49	0.97	7.92	9.42
sushi	6.85	6.53	1.99	10.11

FEATURE: Review competitors in similar industries

AS A food delivery service **I WANT TO** check how much other food delivery services are charging for the same meals **SO THAT** I can adjust my business strategies to become more relevant in the market

SCENARIO: Looking at Asian restaurants from the perspective of Menulog to see how my price compares to Uber Eats, Doordash and Deliveroo

GIVEN I am on the results page for Asian restaurants

WHEN I click on a restaurant

THEN I should see all the prices for each menu item offered by this particular Asian restaurant and note the price differences across the food delivery services

WHEN I have finished comparing all the restaurants in this Asian category

THEN I can evaluate if I should change my prices to reflect my competitors' prices

FEATURE: Determining the cheapest price to advertise to customers

AS A restaurant owner **I WANT TO** see which food delivery service gives my customers the best deals **SO THAT** I can advertise to customers which food delivery service to use for the cheapest meals from my restaurant

SCENARIO: Looking at the *Time for Thai* restaurant page from the perspective of the owner of *Time for Thai* to see what my prices are on each food delivery service

GIVEN I am on the *Time for Thai* restaurant page

WHEN I compare prices for each menu item

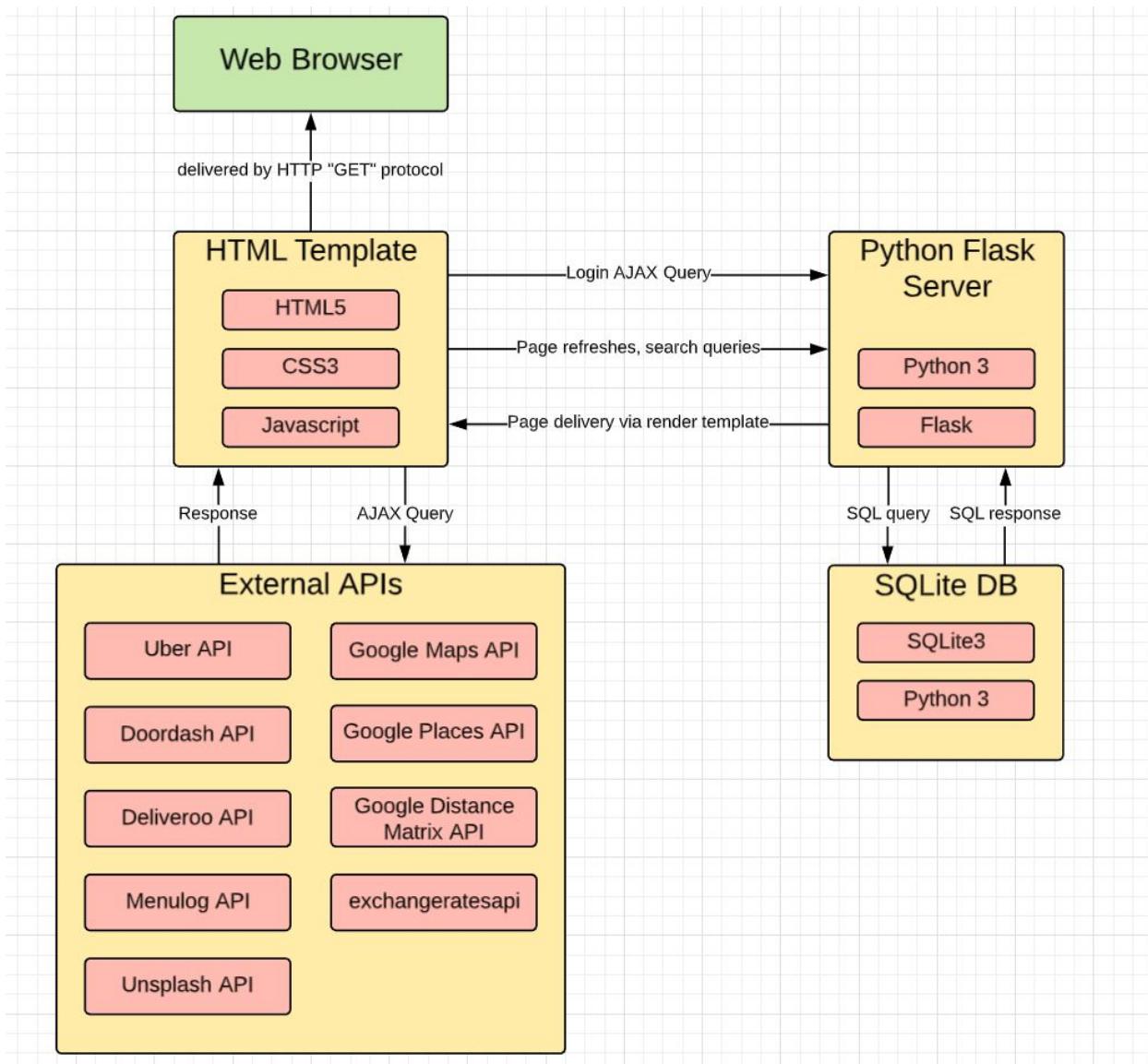
AND notice a trend that my menu items are \$2 cheaper on Doordash

THEN I will advertise this price and instruct customers to order on Doordash on my social medias

SOFTWARE ARCHITECTURE AND IMPLEMENTATION DECISIONS

FINAL SOFTWARE ARCHITECTURE

The diagram below illustrates the overall software architecture of our solution. The SQLite database acts as the model, the Python Flask Server acts as the controller and the HTML template acts as the view. The external APIs will interface directly with the HTML template through Javascript AJAX scripts in order to provide a clear and fair comparison between the different food delivery service apps.



HTML5 (View)

HTML5/CSS3 have both been used for the UI as they are universally accepted by web browsers on both desktop and mobile. This enables users to access our solution across multiple platforms and having the same homogeneous experience. HTML5 further allows for complex UIs without the need for proprietary applications such as Flash to be installed. HTML5 also complements Javascript due to its inbuilt functionality with HTML web pages through its DOM (Document Object Model) manipulation library. This enables us to load webpages asynchronously which makes the webpage more dynamic and natural.

Python Flask Server (Controller)

We chose to use Python 3 as it has the Flask framework which we use as a middleware and server database manager. The framework automates threading and handling security vulnerabilities, and also allows the majority of the application to be contained within a single deployable, improving debugging and portability. Python 3 further has a comprehensive JSON library with similar dictionary/object structures which simplify interactions between the frontend and middleware. Python 3.7 onwards also has an effective SQL library which gives us more flexibility when using SQL queries and more condensed database functions. These features of Python 3 are therefore suitable for a fast and safe prototype since memory, security and allocation issues are all handled by the included frameworks.

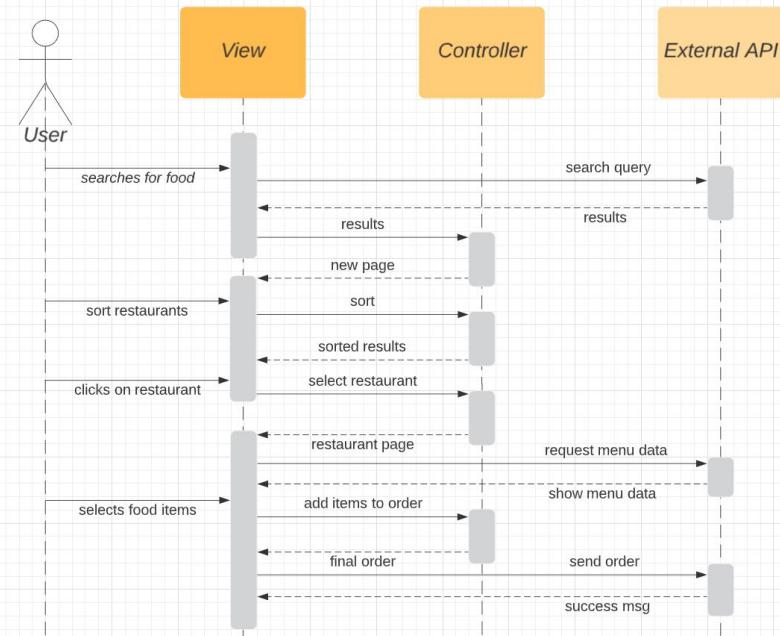
SQLite Database (Backend)

A SQLite3 database is used in the backend to handle data.. We used Python 3 to interface this database as it would also improve the consistency of the database interactions across the rest of the middleware (which is also written in Python 3). SQLite3 allows for memory only databases free of the client server architecture of PostgreSQL providing a more easily debuggable program and faster load times. It also met the SQL standard of being easy to move between both a dedicated server environment and local storage. SQLite3 also does not have unnecessary overheads such as custom types and assertions which would otherwise reduce the overall computational overhead of the system. With SQLite3 being easily integrated with Python 3 functions, it also allowed for a more maintainable link between the database and the Python middleware.

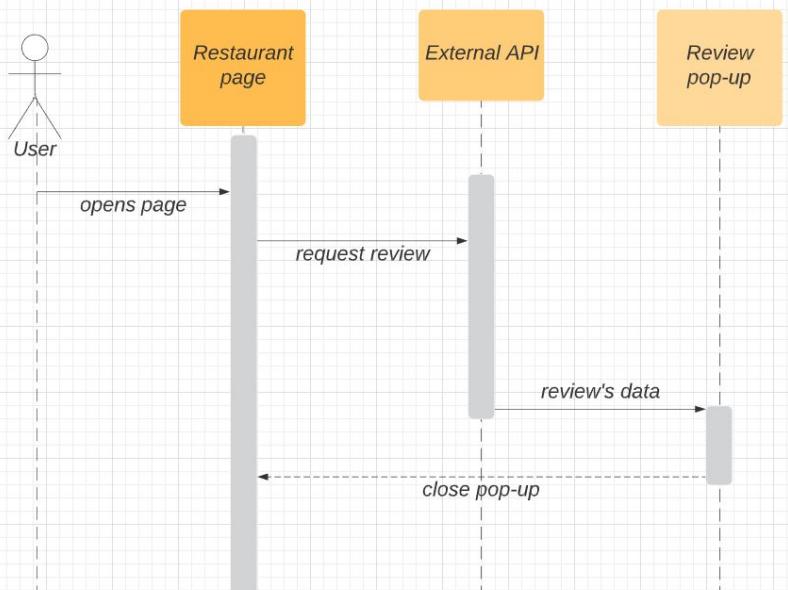
SEQUENCE DIAGRAMS

These diagrams illustrate how each software component works together in order to satisfy the user's requests.

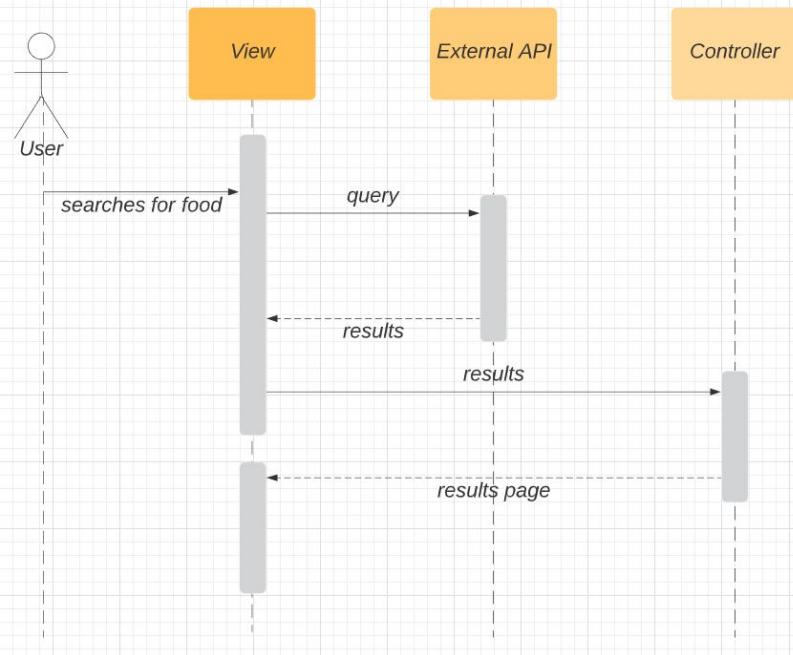
Make an order



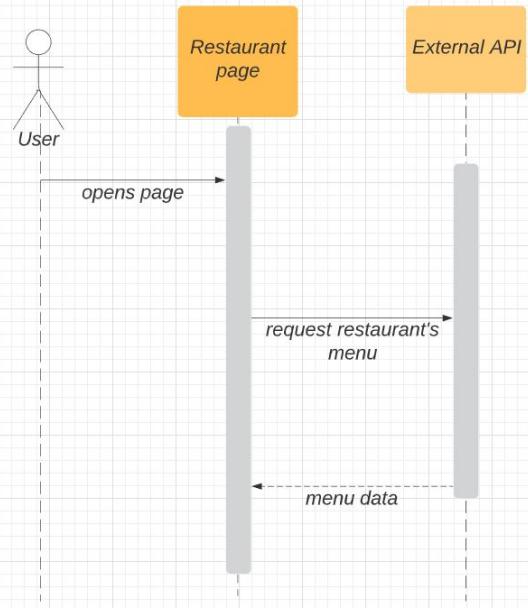
Find good restaurants (see review ratings)



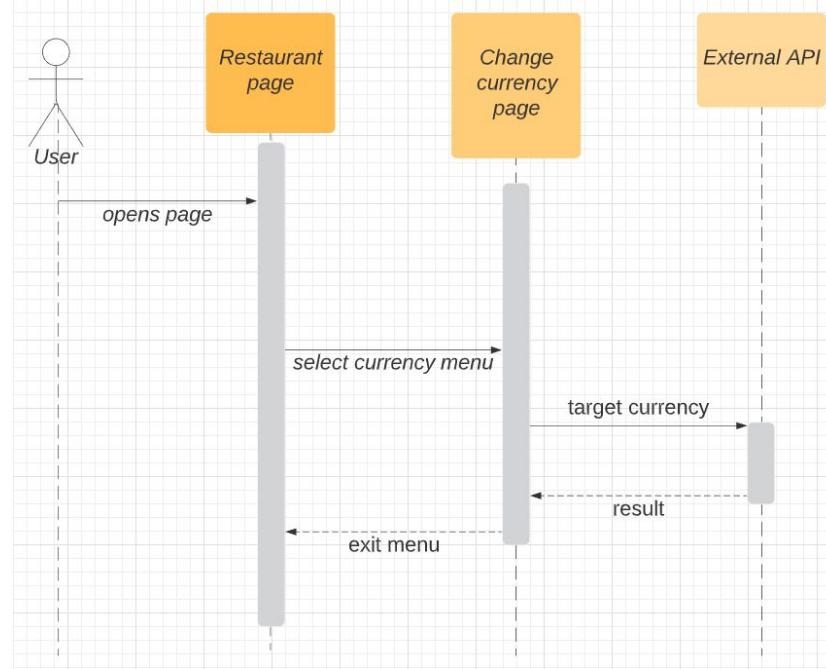
Find food (find specific food matching search query)



Compare prices among different services



Change currency to foreign currency

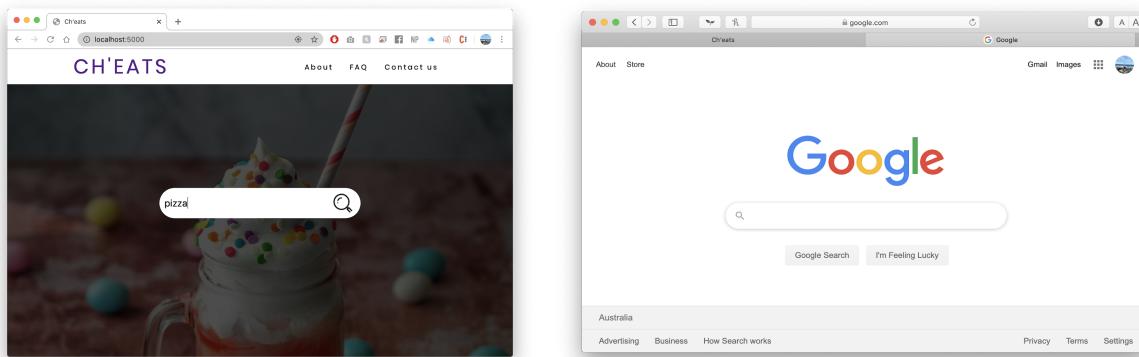


DESIGN DECISIONS

Landing page

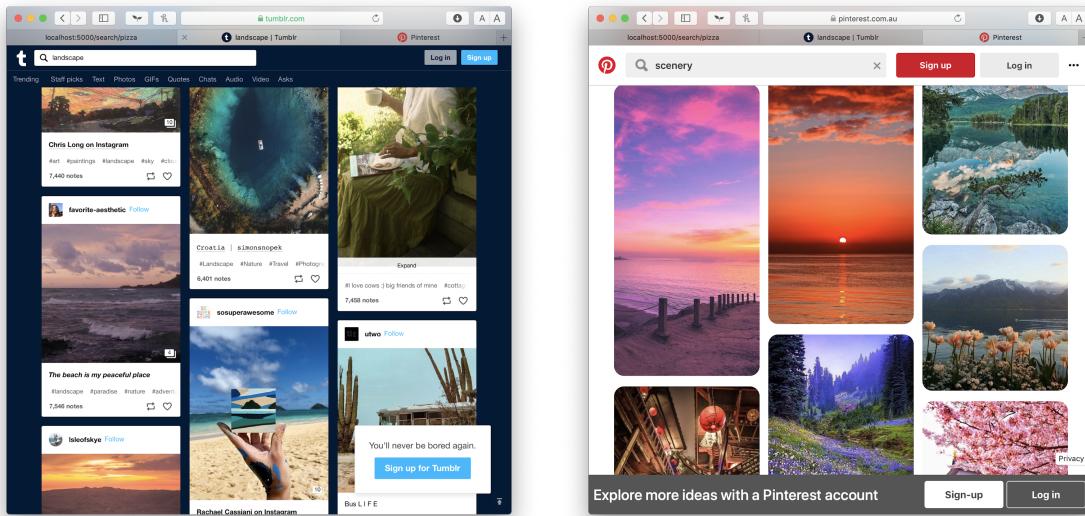
Our aim was to create a simple and easy to use landing page in order to help each user with each decision they have to make. Following Google's standards for modern and minimalistic websites, we have one search bar in the centre to prompt the user to make their first choice in their decision making process - deciding what type of food to eat. There are no other elements in the body of the website to reduce clutter and confusion of the user.

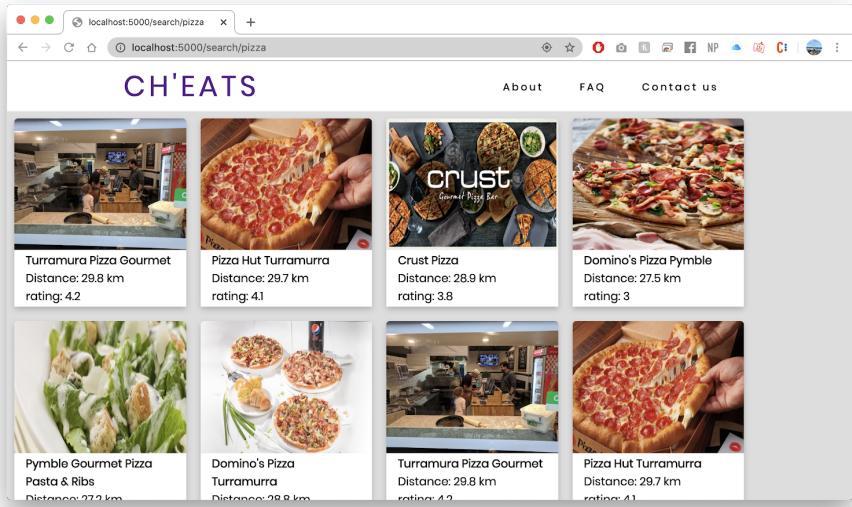
The **Unsplash API** is used to generate a dynamic background - it requests and receives food-related photos from the API, then chooses a random photo to be used as the background. These images improve the look of our website and keep it dynamic and fresh.



Search results page

The search results page is responsible for communicating to the user what restaurants are available as well as its ratings and distance away from the user. To do this in an efficient way, we used cards to group this information for each restaurant, making it intuitive to understand. This follows the card aesthetic used by Tumblr and Pinterest, making it easy to see options at a glance.





We've also extracted images from the restaurant and ratings via **Google Places API** to continue the modern aesthetics of our website. A study carried out by PR News found that online content with good images get 94% more views than those without⁶, so our website aims to highlight each restaurant and assist users in choosing the right one for them. The distances are calculated using the **Google Distance Matrix API**.

Search results are generated by cross-referencing the search terms with the cached SQL database. Our Python scripts return a JSONified dictionary of restaurants that the front end extracts and displays on the page. We decided this would be the best course of action as it was the simplest and most conventional way of doing this, which would reduce development time and ensure that we wouldn't run into any sudden surprises by taking an entirely novel approach.

Restaurant page

Once users have chosen a restaurant, the user can identify items they would like to order and see their corresponding prices across food delivery services at a glance. Hence we've opted for a gridview where rows are menu items and columns are colour coded for each food delivery service. This table format is used by Netflix and World2Cover travel insurance to compare options when users must make a choice. It is a clear and efficient way to see the differences among the options.

⁶ <https://www.prdaily.com/visual-content-receives-94-percent-more-views-than-text-only-marketing/> (June 2017)

	Uber Eats	DOORDASH	MENULOG	deliveroo
Northern Style Dumplings	7.03	8.61	2.31	17.44
Southern Style Dumplings	19.19	11.89	13.85	4.25
Chicken Fried Rice	14.38	1.65	13.42	15.96
sushi	11.6	11.07	3.38	17.13

On the right side of the restaurant page is a map showing the location of the restaurant implemented through the **Google Maps API**. This informs the user how far away it is and can therefore gauge a rough estimate for a delivery time.

These menu items will be retrieved from the food delivery services' APIs when the page is loaded. This is to ensure that the prices are as accurate as possible. We felt it was unnecessary to store them in our database as the user would have to refresh it anyways when they load the page.

We made the decision to display every price instead of only displaying the most affordable cost since there was a possibility a user could have a voucher for a certain service, such as Uber Eats, and could be using our site to decide which menu item to use the voucher on regardless if it was the cheapest cost or not. Furthermore a user could be collecting points for a food delivery service and we did not want to restrict other uses of our product.

SUMMARY

Overall, we believe our product has an intuitive design that naturally assists the user in their decision making process. First the user decides what time of food they would like to eat, and this corresponds to their first encounter on our site - the search bar. Next the user must choose which restaurant to order from, which they can do by selecting a restaurant from the result page. Once on the restaurant page, they can determine which food delivery service they would like to use by comparing the prices on the page for the menu item they want.

HTML5/CSS3 used in conjunction with Javascript is in the frontend because it is widely supported meaning it is functional both on desktop and mobile. It is also a robust and modern standard for interactive web apps. A Python 3 Flask server is used as its framework handles all the security and memory issues, and is a portable architecture. Python 3 also allows us to interface easily with the SQLite3 backend database, which meets the modern SQL standards with its fast load times.

TEAM REFLECTION

RESPONSIBILITIES AND ORGANISATION OF THE TEAM

Thomas was the leader of the team. He was responsible for organising meetings and ensuring all members of the team were delegated a task that was to be completed for the next. Thomas assisted development by leading the comparison functionality that was the essential to our project. He also integrated our product with the exchange rates API.

Cameron and Haodong were the main developers of the project. Once we realised we had to create an API that emulated the food delivery services, Cameron was responsible for its creation. Cameron also linked the three Google APIs with our project that was used across the whole site. Haodong was responsible for populating and maintaining the SQLite3 database our frontend retrieved data from.

Hayes was responsible for the documentation of the team. He was responsible for creating high-end fidelities, presentations and reports. Hayes also assisted in the development of the project when needed such as integrating the Unsplash API and the development of the search results page.

ISSUES AND LIMITATIONS

Unfortunately for this project, we were not granted access to any of the delivery services' APIs because our product was not a commercial product. After consulting with our mentor Decker, we agreed that an alternative solution would be to build an API that emulated Uber Eats, Deliveroo, Menulog and Doordash. As such, we've emulated the data the same way we would fetch it from these food delivery services, formatted in the same way as if fetched from the API.

The current climate of COVID-19 also made it difficult for our team to work together. The circumstances all impacted us in different ways personally and there was a period of time where none of us were able to work together. Almost all members of the team had to change their location of residence in response, so anxiety and stress were common throughout the team. Despite this, however, we are glad to have completed our project satisfactorily in the current climate.

Furthermore, the cost of using Google's APIs restricted the amount of testing we could do. The screenshot below is the running bill we received towards the end of the project. It was essential to make queries during development to ensure correct algorithms were being designed which in

total amounted to over \$150. As a result, we were only able to complete unit testing but not as much comprehensive testing as we would have liked due to financial constraints. Unfortunately we were not made aware of this before the project and we believe clarity should be given to future students who take SENG2021.

Apr 1 – 30, 2020	Places API Places Photo: 4087 Counts [Currency conversion: USD to AUD using rate 1.649]	A\$47.18
Apr 1 – 30, 2020	Places API Places - Nearby Search: 368 Counts [Currency conversion: USD to AUD using rate 1.649]	A\$19.42
Apr 1 – 30, 2020	Places API Contact Data: 368 Counts [Currency conversion: USD to AUD using rate 1.649]	A\$1.82
Apr 1 – 30, 2020	Places API Atmosphere Data: 368 Counts [Currency conversion: USD to AUD using rate 1.649]	A\$3.03
Apr 1 – 30, 2020	Maps API Dynamic Maps: 248 Counts [Currency conversion: USD to AUD using rate 1.649]	A\$2.86
Apr 1 – 30, 2020	Distance Matrix API Distance Matrix: 17455 Counts [Currency conversion: USD to AUD using rate 1.649]	A\$143.92

PERSONAL REFLECTIONS - How did the project go in your opinion? Would you do it any differently now?

Thomas

My personal journey during this project was mostly characterised by steep learning curves. First it was learning about how to start to manage a team. Second was learning the programming languages through youtube clips. Lastly was becoming educated on how API's work and how companies do not give them out that easily, as well as the trouble of CORS.

This project forced me to learn a lot very quickly - a positive and a negative. For example I was learning basics of javascript like learning that console.log() printed out on the console, whilst writing a complicated asynchronous function with three await API calls. However I did learn a lot of new things which is a positive.

There are a few other positives I would like to make note of. These include the role delegation throughout the project and how relevant our idea was or became throughout the project span. The role delegation initially started out as two teams, frontend and backend. Then this changed to more individual delegations, report, connect to exchange API, write our external food delivery API. This worked very well as we got our deliverables in on time and at a quality approved by our team.

Now to the negatives. The online nature of the team meetings and presentations did factor into some communication errors throughout our project. Personally my WIFI is not of the best quality

so I missed some of my cues during the presentations. In addition to this, there were clear communication barriers which could have been solved very easily in person. Finally, mentor meetings would have been much quicker and more useful when they were in person.

There are many things that could have been improved within this project. These include a more aesthetic UI, having an API key to the food delivery websites, as well as using many additional API's. Firstly having a more aesthetic UI would be down to having more experience with javascript, HTML and CSS. If we had an API key for all of the delivery websites, this would mean the application would be fully functional and have actual food delivery prices, in addition to this we would be able to order on site.

Completely new additions to the project include the additions of additional APIs. These include Edamam API which means we would be able to get nutritional data on all the meals, which means we could sort the menu items by different diets, nutritional value and also get recipes. Other APIs include google cloud translation API to add additional international functionality to the currency conversion, and rapidAPI which would have drastically improved our search keyword functionality via AI. Finally instead of relying on the API keys for the food delivery websites and if we started early enough we could have crowdsourced the data, which could have given us real data rather than having to fake it, but this process would still not have been reliable.

Overall, I did struggle a lot through this project, I did learn a lot and am proud of what we have achieved as a team.

Cameron

I felt the project went well considering that our initial idea was found to be impossible as a result of the data we required no longer being publically available. I found the programming difficult as I had to teach myself javascript which was very taxing and time consuming. The scope and complexity required by the course is not reflected in the prerequisite, making it much more time consuming than the standard 6 UOC workload that i have come to expect.

In regards to the programming it was very difficult as key members of our team and myself spent hours trying to debug code which we didn't fully understand as well as keep up with our other courses. With some features even being prevented by new browsers in the name of security, exemplified by protocols such as CORS which we had never encountered or been taught about were a major issue. This was only made harder by COVID-19 as our in person meet ups went online which made it even harder to communicate issues or seek help within our team. Overall, my personal understanding of the coding languages hampered my progress on some of the modules I had to write despite having done personal projects involving these languages, demonstrating the required complexity of the work required in this course.

A key external concern I would like to express is the fact that, in the name of privacy, many companies no longer publicly provide rest-apis which makes it nearly impossible to complete the requirements of this project as there is a direct specification that 4 apis must be used. With this abstract number completely distant from the other requirements whereby we need to make a commercially viable product. It overall, makes the whole course seem confused when arbitrary values are specified in conjunction with broad contextual requirements. In absolution, this often prevented us from receiving the marks I would have expected considering the technical complexity of our idea as well as real life relevance. It often felt as though this course was confused, and relied heavily on the individual talents of our group members.

The culmination of these inadequacies in the course and over reliance on individuals to learn and research new languages and techniques whilst dealing with the developing circumstances of COVID-19 caused me a huge amount of additional stress. In this project i ended up putting over 200 hours of programming and debugging alone not including the time taken for meetings, both with our mentor and team members. Other issues arose as many team members had additional commitments as can be expected, with all of us having jobs and family commitments which consume a large amount of our day, with this often resulting in late nights and frustrating debugging sessions amalgamating in the abandonment of features and hence penalisation by the markers.

Overall, I feel this project was done to the highest standard and capacity given the external variables we had to overcome. I feel personally as though this course was designed for a world where apes were publically available and often unregulated. However, as companies ‘regulate’, themselves they also conveniently block our ability to use their technology and data in our projects. Creating a greater task for the students in the course, as we are left to stitch together a “unique and contextually relevant” idea with very little data and technical knowledge to work with. In conclusion, i feel that we could have been more prepared coming into the course as well as rewarded more compationality for the amount of work we did, as the marking was often broad and the marking criteria poorly specified and i felt that although we were making progress towards developing our product we were hampered by arbitrary rules and small points in the marking criteria someone would consider rudimentary.

If i was to change a few ways we went about the project, it would mainly be the idea. With the contextual viability of the idea seeming to be merely a user use case. I would select a more data heavy application such as a graphing, trading or maybe even a star map. Which would allow us to use more backend languages which we would have been more comfortable with as well as reduce the overall workload of the course. With there being little incentive to attempt to create a

truly commercially viable and useful product with instead a simple product being more easy to document and thus, increasing our overall marks for the course.

In terms of our group, I would have made sure that people with more experience in certain areas were more tasked towards their strengths. With a key example being the front end design and javascript async programming. It would also have enabled us to create more “features” as we would have been able to implement these more easily in a simpler application. I think our tasking could have also been improved to make sure that everyone always had something to work on and our deadlines were more heavily enforced by the group as best we could as often a key piece of the backend was still being developed as the front end team attempted to still progress without the necessary backend support. With this disconnect between the two teams often making the project less elegant and thus more error prone, which towards the end became a major issue.

Haodong

Considering the limitations placed on us by the lack of any actual data, I think we did as well as we could. Despite being the only team member with anywhere near thorough JS and CSS experience, I was initially overwhelmed by the technical scope of our project. Ultimately, however, I think we managed to overcome those issues and come out better in the end. If I had the opportunity to go back in time and advise my former self, I’d tell him that he should have picked a project that we would have been able to get real data for more easily - unfortunately, these days, more and more APIs are being stripped and locked down, and the Web 2.0 dream is quickly dying. I fear that future students who take this course may find it totally impossible due to this fact. The COVID-19 pandemic held us back from being able to work together in the way we initially wanted to, but thankfully, we managed to resolve this issue among ourselves.

Hayes

I believe we completed the project satisfactorily and the problem we decided to tackle as a team was a new and relevant one. However, if I had known about the onset of the global pandemic, perhaps I would have chosen another project that fit our time frames better. It was a shame we lost time while we all individually transitioned to a new lifestyle. Time management was further an issue for the team as we were all learning on the go (since 3 of the 4 members had zero experience with Javascript) and this made it hard to separate what was achievable and what was not until the end. Despite learning a little bit of Javascript, CSS and HTML5, it would have helped if we were given lessons throughout the course rather than relying on online tutorials. Nevertheless, I believe the team worked cohesively together, especially towards the end, and I am happy with our final product.