

Team Name	Searching for a Deeper Meaning	5
Member 1	Will Garside	4
Member 2	Greta Ramaneckaite	3
Member 3	Rufus Cope	2
		1

What	What we evaluate	Value	Comments
Quality of Code		4	Good code quality, newer parts still need refinement
	Compilable & Executable. We must be able to run your code without any problems on standard MACs and using standard emulators	5	There are no issues with compiling or running the code. The node modules must be installed before running
	Code style, naming, format (consistent format regarding class and variable naming as well as indentation)	4	Following standard ES6 and Web Application development structure
	Exception (Proper use and handling of Exceptions in your code)	4	All major exceptions are handled, however some smaller ones (usually client-side) are still to be dealt with
	Code Organization (e.g. use of Java packages - dcs.shef.ac.uk.com4280.net, distribution of directories etc).	5	Code organisation is immaculate, neatly structures as specified in brief
Code: Details			
Querying:			
1.1.1. Searching for restaurants and displaying results	Ability to search by keywords	5	Handle case sensitivity, multiple word search
	Ability to search via schema.org tags	5	
	Displaying via list of restaurants	5	Display most important information about the restaurant, with linking to full pages
	Displaying via map	5	Users can see the nearest Restaurants or a map
	Quality of solution	5	
	Correctness of results	5	
	Documentation:		
	Issues	4	
	Design choices: Motivation	4	
	Design choices: Requirements	4	
	Design choices: Limitations	4	
1.1.2 The restaurant page	Displaying address, type of cuisine, and official photos	5	All required information is displayed with additional features
	Displaying photos uploaded by customers	5	Images can be expanded to view in fullscreen
	Displaying a global customer rating for the restaurant, reviews,	5	
	Displaying a small map	5	Google Maps has been embedded
	Quality of solution	5	All details are visilble to User
	Correctness of results	4	
	Documentation:		
	Issues	4	
	Design choices: Motivation	4	
	Design choices: Requirements	5	
	Design choices: Limitations	4	
1.1.3. Allowing restaurant reviews	Quality of solution	5	Only logged-in users can leave reviews
	Correctness of results	5	Reviews are displayed immediately thanks to socket.io implementation
	Documentation:		
	Issues	5	
	Design choices: Motivation	5	
	Design choices: Requirements	5	
	Design choices: Limitations	4	Cannot upload images from device's local storage

Team Name	Searching for a Deeper Meaning			5
Member 1	Will Garside			4
Member 2	Greta Ramaneckaite			3
Member 3	Rufus Cope			2
				1
1.1.4. Creating a new restaurant page	Quality of solution		5	High quality implementation, covering above and beyond spec
	Correctness of results		4	Images can only be uploaded from file. NOTE WebRTC used on Review submisison
	Documentation:			
		Issues	5	
		Design choices: Motivation	5	
		Design choices: Requirements	5	
		Design choices: Limitations	5	
1.1.5. The Server Architecture	Quality of solution		5	Code is high quality, directory structure is well laid out and easy to understand
	Correctness of results		5	
	Documentation:			
		Issues	4	
		Design choices: Motivation	5	Followed Java standards
		Design choices: Requirements	4	
		Design choices: Limitations	4	
1.1.6. The Database	Quality of solution		5	Successful deployment of a reliable database
	Documentation:			
		Issues	5	None
		Design choices: Motivation	5	Deployment reduces complexity
		Design choices: Requirements	5	MongoDB and Mongoose used successfully
		Design choices: Limitations	4	An active internet connection is required
1.1.6. Progressive web app	Quality of solution		4	Service worker implemented, however some corner cases still produce minor errors
	Documentation:			
		Issues	4	A couple of non blocking POST errors still exist
		Design choices: Motivation	5	Service Worker's range of different responses to different requests
		Design choices: Requirements	5	All requirements met
		Design choices: Limitations	4	A lot of time wasted on silly bug, otherwise good
1.1.8. Quality of the solution/keeping your solution manageable	Quality of solution		5	
	Documentation:			
		Issues	4	High quality solution, only missing minor parts of brief
		Design choices: Motivation	5	
		Design choices: Requirements	4	
		Design choices: Limitations	4	
Additional Features	describe any additional features you implemented	Contact Form	5	Email confirmation sent and contacts saved to DB
		User Management	5	Users can manage their details and see their submissions (Reviews or Restaurants)
		Advanced search and display ordering	5	Advanced Search controls allow better refinement for Users
Quality consideration	Use of JSON for communication of data		5	JSON has been used where appropriate
	Use of AJAX		5	AJAX has been used where appropriate

Team Name	Searching for a Deeper Meaning		5
Member 1	Will Garside		4
Member 2	Greta Ramaneckaite		3
Member 3	Rufus Cope		2
			1
	Quality of the interface (e.g. use of appropriate elements, no confusing views, etc.)	4	Interface is high quality, and layout of web pages is clear
	Use of WebRTC for image taking	5	WebRTC has been successfully implemented to allow the Uer to take images on review submission
	Use of http and API error codes for server communication	4	
	appropriate sync mechanism with the server	4	
	any other	4	
Documentation	Code documentation: JavaDoc like /doc dir (in-line commenting, javadoc)	5	JSDocs have been generated using the in-file commenting. Comments have been made according to JavaDoc standards
	Introduction, Conclusions, Extra information (if needed), Bibliography (if relevant), present, penalty if not!	5	All document report requirements have been met