Planning and Organisation

**Team Members:**

Aaron Nogal - a1725688

Barry/Zhi Tong - a1736023

Anas/Ansul Azam - a1680278

Robert Norris-Karr - a1731824

|  |
| --- |
| **Key:**  Needs to be done  Is partially done  Is fully done  Important consideration |

**TODO List**

* Review
  + Have another group review your website design
  + Have another group perform basic usability testing.
* Research
  + Record any pages that are a good example of what the site might look like
  + Analyse the recorded pages (what is good, what is bad)
* Data Plan
  + (see highlighted parts of part 3 below)
  + Reference ER Model, and see if any changes need to be made to it.
* UI design
  + Mobile design
  + Mark dimensions on desktop design
  + Note any responsive CSS
  + How does it minimise kinematic and cognitive load?
* Does it meet the standards & heuristics referenced in lectures and on MyUni?
* Implementation
  + All HTML and CSS
    - (see highlighted parts of part 2 below)
  + Some Vue.js
* Feature Plan
  + Feature per page
  + Needs more information per feature, for example, how it is going to be implemented and what should be checked on each page with javascript.
  + Choose one of the following
    - Find something nearby using the customer's location and preferences, choose and book a suitable nearby restaurant on the customer's behalf.
    - Customer calendar reminder; customers can have their bookings automatically added to their Google Calendar when making a booking, and removed if cancelling.

# 

# Highlighted Specification

# **Part 1 – Designing your website**

Think about and explore web pages. What might a restaurant booking website look like? What information might users be required to supply? How will they log in? How will the search work? What about the map? How should restaurants manage their information? **You are encouraged to think about additional features for this application; we have outlined the minimum.**

* Research existing websites to get ideas for your web application.
  + Record any pages you find that are good examples of what your site could look like. What parts work well. What parts don’t work well?
  + Also record any examples you find of features you might want to include in your site.

Once you decide what a user will be able to do, then you need to think about how they will do each task and how you will make those tasks visible on your site – **keeping in mind the articles and lectures on design and usability**.

* Draw (don’t code) what your site will look like.
  + It should be obvious from the drawing what each part of the site does.
  + Be sure to include any dimensions – if the window is resized, how do elements change?
  + How might the site look on a phone vs desktop?
* Your website will need to have the following features:
  + Users should be able to sign up and log in to make or manage table bookings and manage their information.
  + Restaurants should be able to sign up and log in to manage their profile, number of tables available, and individual bookings.
  + Users should be able to search for restaurants using appropriate criteria.
  + Users should be able to leave a review after their booking.
  + Users should be able to choose to link a social media/email/other account, allowing login via that platform, to make logging in easier.
* Plus one of the following special features should also be present:
  + Find something nearby; Using the user's location and preferences, choose and book a suitable nearby restaurant on the user's behalf.
  + Calendar Reminder; Users can have their bookings automatically added to their Google Calendar when making a booking, and removed if cancelling.
* Thinking about the above features, describe how each of these features of your site will function, plus any others that you might like to have.
  + What can the user do on each page?
  + What happens when a user performs one of these interactions?
  + If a change occurs to the page/site following an interaction, clarify what that will be and draw if necessary.

Once you have a basic design, you now need to review and tweak it for usability and accessibility.

* Review your design.
  + How does it minimise kinematic and cognitive load?
  + Does it meet the standards & heuristics referenced in lectures and on MyUni?
* Have another group review your website design and perform basic usability testing.
  + Which parts of your site were clear. Which weren't?
  + Record the feedback from the other group and any findings.
* Record any updates to your design made as a result of the above testing and review.

As you learn more about implementing web applications throughout this course, your design will likely change; that’s okay, but be sure that as those changes creep in, you continue to review your site for usability and accessibility. We will be assessing you for this in your final presentation.

**Your week 7 tutorial time is a great opportunity to ask questions, get feedback, and compare your design with another group before the deadline.**

# **Part 2 – Building your website**

Now that you have a solid design, it’s time to start coding it! You should now have a firm grasp on these languages and be able to implement ALL of the static elements of your website.

* Use HTML and CSS to implement all of the pages in your site.
* Follow best practices as outlined in lectures and exercises.
  + Ensure CSS is used appropriately to provide a consistent and maintainable style across all your pages.
  + Ensure your implementation passes validation.

As you build your website, begin to add interactive features using JavaScript and Vue.

* Use JavaScript and Vue to further build your design.
* Start to add client-side code for users to book a restaurant table & manage a booking
  + Bookings made should be able to be reviewed and modified after a user has made them, however it is not expected that those bookings are saved between uses of the site.
* Follow best practices as outlined in lectures and exercises.
  + Ensure JavaScript is used appropriately to provide a consistent and maintainable code across all relevant pages.
  + Ensure your implementation still passes validation after modifications.
  + Ensure your JavaScript and Vue passes linting.

Do not worry about making user interactions persistent (i.e. user changes remain after a refresh) at this stage, but do begin to think and make notes about which interactions will need server resources.

# **Part 3 – Planning your Routes and Database**

Now that you have a solid design and basic client side implementation, you can start thinking about the content that your web application will be dealing with, and how.

* For each of the features in your website, create a data plan that lists the different pieces of content/information that your web application will be dealing with & determine where it should be stored, what format, and where it should be processed.
  + Where does the information come from?
  + What form should it take?
  + If the information is on the server, what will the client need to send to retrieve that data?
  + If the information is on the client, how will it be sent to the server?
  + Does all information need to be stored on the server?
  + What processing needs to be done to make the data useful?

Once you have an idea of the different pieces of data that you'll need to be working with, you can start thinking about how you'll store that data on your server.

* Using the methods outlined in lectures and pracs, create an Entity-Relationship diagram for your web application's data.

# **Milestone 1 Submission:**

Submit all work done so far from Parts 0-3 to the MyUni Group Milestone Submission page before the deadline, **11:59pm Friday Week 9**. This should include:

* Your research, design, feedback, review and changes from Part 1
* Your basic client side implementation from Part 2
* Your data plan and E-R model from Part 3

Submit your group's work as **a single ZIP file;**

* Be sure to include all your work in the ZIP file.
* If you have your files organised in folders/directories, be sure to keep this structure when adding them to your ZIP file.
* Do not include your node\_modules folder in your submission, but do ensure you include any other server files.
* Unzip and test your site on a Lab PC with npm install then npm start before submitting to make sure nothing is missing/everything works as expected. **It is your responsibility to ensure that your code works in the environment it will be marked in before submission.**

This submission is worth 10% of your final course grade

* A detailed mark rubric will be available on the submission page closer to the deadline.
* Your mark will be available in the MyUni gradebook within 2 weeks of the deadline unless otherwise noted.
* **Late submission penalties apply!** (see course outline)