

**Agile Software Project [CM2020]  
Final Report  
Team 64**



Created by:	Student number:	Email:	Roles:
Aditya Vijindra Pandhari	200629016	<a href="mailto:pandhari001@mymail.sim.edu.sg">pandhari001@mymail.sim.edu.sg</a>	Scrum Master
Kevin Royce Thomson	200553179	<a href="mailto:kevinrt001@mymail.sim.edu.sg">kevinrt001@mymail.sim.edu.sg</a>	Product Owner
Nicole Ching Yun Wen	200618766	<a href="mailto:nywching001@mymail.sim.edu.sg">nywching001@mymail.sim.edu.sg</a>	Developer
Charlotte Chong Qian Le	200261104	<a href="mailto:glcchong001@mymail.sim.edu.sg">glcchong001@mymail.sim.edu.sg</a>	Developer
Nachiappan Kannappan	210201451	<a href="mailto:nachiapp004@mymail.sim.edu.sg">nachiapp004@mymail.sim.edu.sg</a>	Developer

[Word count \(excluding Appendix\): 9971](#)

## Table of Contents

<b>BACKGROUND.....</b>	<b>3</b>
ABSTRACT .....	3
OUTCOMES .....	3
INTRODUCTION .....	3
<b>PLANNING AND RESEARCH.....</b>	<b>4</b>
TIME AND RESOURCE ALLOCATION.....	4
TEAM BREAKDOWN .....	8
<i>Group Meeting Log.</i> .....	8
FRAMEWORKS CONSIDERED .....	10
PROJECT MANAGEMENT TOOLS.....	13
<i>ClickUp.</i> .....	13
<i>Burndown Chart</i> .....	13
<i>Gantt Chart.</i> .....	14
RESEARCH – SUMMATIVE (INITIAL REPORT).....	15
RESEARCH – CURRENT .....	16
<b>PROTOTYPING AND ITERATION .....</b>	<b>17</b>
LOW FIDELITY WIREFRAME (CONCEPT VS PROOF-TO-CONCEPT EVALUATION) .....	17
MEDIUM FIDELITY WIREFRAME (CONCEPT VS PROOF-TO-CONCEPT EVALUATION) .....	17
HIGH FIDELITY MOCK UP (CONCEPT VS PROOF-TO-CONCEPT EVALUATION) .....	19
FEATURES TO IMPLEMENT.....	20
<b>DESIGN .....</b>	<b>22</b>
USE CASE MODEL.....	22
<i>Use Case Diagram</i> .....	22
USER TESTING.....	23
<i>Guerrilla Usability Testing</i> .....	23
<i>User testing – 1<sup>st</sup> iteration</i> .....	24
<i>User Testing – 2<sup>nd</sup> iteration</i> .....	27
<i>User Testing – 3<sup>rd</sup> iteration</i> .....	32
DESIGN CHANGES – BETWEEN PROTOTYPES.....	34
<b>SYSTEM DEVELOPMENT .....</b>	<b>42</b>
CODE DEVELOPMENT .....	42
VERSION CONTROL PROCESS.....	45
UNIT TESTING.....	48
CHALLENGES IN IMPLEMENTATION.....	51
SPRINT RETROSPECTIVE.....	52
EVIDENCE OF MODULE COUPLING.....	53
<b>ANALYSIS .....</b>	<b>54</b>
PROJECT ANALYSIS.....	54
<b>EVALUATION.....</b>	<b>56</b>
PROJECT EVALUATION .....	56
PRODUCT EVALUATION.....	58
<b>CONCLUSION.....</b>	<b>61</b>
PROJECT CONCLUSION .....	61
PERSONAL REFLECTION (NAME) .....	61
<b>APPENDIX .....</b>	<b>63</b>
APPENDIX A .....	63
APPENDIX B .....	64

## Background

### Abstract

The foundation of this project is to create a unique, consolidated food delivery platform applicable to all users of popular food delivery services (e.g. Grab Food, Foodpanda etc.). In contrast to using a specific food delivery service provider, users can now view food options, delivery services and fees from multiple companies collectively within our software. This will aid all users of essential food delivery services in making more cost-effective options while saving time. Also, this creates healthy competition in the industry for the different food delivery companies and opens doors for new economic growth in the business.

Based on results from our User Testing research, our team has successfully run the fundamental integrity of the web application with the intention to progress the software further in the near future.

### Outcomes

The overall goal for our team at the current milestone is to enable users to create an account and log in to an account on our desktop site, where they will be automatically redirected to the web application's home page. From there, users should be able to navigate to a myriad of restaurant and categorical options, make their selection of food items and view the price comparison for it with the different delivery partners and then proceed to a checkout page. Ergo, users can access a payment methods page, track their orders, and view their existing order histories. We are also looking to include a profile page with options for users to access multiple basic account-altering services such as a change in name, password or email.

In its entirety, we hope that our team can successfully deliver the functions mentioned above while ensuring that navigation between these pages runs smoothly with no issues or complications.

### Introduction

At the current stage, our team has deployed various research and planning efforts to facilitate our web application's first fully functioning iteration, which we will discuss further in the next section. We have structured our team with three front-end developers and two back-end developers working on the project concurrently. Our front-end team are tasked with presenting our content in the most user-friendly manner, closely abiding by the research we had done in our initial report. Our back-end team will be creating the basic functionality and foundation for the web application we aim to accomplish at this stage. Our initial conceptual report will be the guiding principle in this project's developmental stage. We will refer to all the research, planning timelines and prototypes that were done prior.

This report will consist of thorough research on the actual web application development, documentation of how our team adopted the scrum methodology, user testing and unit testing processes, and the tools that were considered and utilised in the process. We will conclude with an analysis and evaluation of our project and product.

## Planning and Research

### Time and Resource allocation

#### *Front-End*

Initially, the team developed iterations closely mimicking the high-fidelity mock-ups we drew up in the initial report using vanilla JS. However, after reviewing with our back-end developers, we have decided to convert to using the React.js framework. Hence, the initial front-end task in our 1st Sprint was familiarising ourselves with certain libraries and frameworks, the foremost being React.js. A user testing research portion was also created to analyse the designs, which can be found in this report's 'Design' section.

Also, part of the front-end team's focus was setting up the git repositories on GitHub to allow for a mutual codebase collaboration between the front-end and back-end teams. We have also organised the web pages we are delivering at this stage of the project into cascading difficulty levels (Figure 1.0 for front-end & Figure 1.2 for back-end). This is to facilitate a more productive stage-by-stage development beginning from the simplest to the most complex pages during implementation. This will reduce the odds of being held back on more complex implementations for a long duration.

#### *Front-end list of complexity in terms of implementation*



Figure 1.0 Complexity ranking for front-end implementation

*Front-end timeline and progress checklist*

Front-end timeline and progress checklist				
Start Date	Duration	Task	Completed	Incomplete
1st July 2022	Two days	Research and familiarisation with React.js framework and essential libraries	Successfully mastered the necessary skill set for React including libraries such as formik, yup and axios	
3rd July 2022	One day	Initialisation of Git repositories via GitHub Installation of essential libraries List web pages into various complexity levels	Successfully created a new repository for the project Completed necessary installations List of web pages	Actual code implementation
4th July 2022	Eleven days (Till end of Sprint 1)	Use of React to begin code implementation for easy level web pages: Sign in, Sign up, Categories & Home page	Successfully implemented code for easy level web pages mentioned	Code implementation for Search bar functionality (moved to Sprint 2)
14th July 2022	Two days	Conduct user testing research and analyse findings	User testing results collected and analysis were done	Changes in some user interface features for the completed web pages (easy) based on user testing findings
16th July 2022	Twelve days (Till end of Sprint 2)	Complete code implementation for remaining easy level (Reset password, Order history) and intermediate level web pages: Restaurants and Individual Restaurant	Successfully implemented code for intermediate level web pages mentioned	Code implementation for remaining intermediate level web pages; Data preload
28th July 2022	Two days	Conduct user testing research and analyse findings	User testing results collected and analysis were done	Changes in some user interface features for the completed web pages (intermediate) based on user testing findings
30th July 2022	Twelve days (Till end of Sprint 3)	Complete code implementation for remaining intermediate level web pages( My cart, Track my order, Order history & Help page) & some complex level web pages (Comparison & Profile page)	Successfully implemented code for web pages mentioned	Code implementation for remaining complex web pages
13th August	Two weeks (Till end of Sprint 4)	Complete code implementation for remaining complex level web pages & implementation of media queries	Successfully implemented code for remaining intermediate level web pages mentioned  Completed media queries implementation	-

Figure 1.1 Checklist for front-end team

## *Back-End*

Before commencing back-end development, the initial task in our 1st Sprint was dedicated to studying specific skills and concepts that were highly relevant to the back-end processes that we will be attempting at this stage of the project. During this time, the focus was put on familiarising with Node.js, a back-end JavaScript run time environment and MySQL database management systems to store our databases. To facilitate communication, the back-end and front-end teams will organize a weekly meeting to make any progress updates and checklist evaluations. This helped ensure that the front-end and back-end development were cohesive.

As our Sprints commence, both our back-end and front-end teams would organize individual online discussions daily to work on the different segments of the web application. The entire team would come together during our weekly meetings and sprint retrospectives to discuss the project's current status. The meetings would include Sprint reviewing, troubleshooting, and communicating any important updates along the way.

### *Back-end list of complexity in terms of implementation*

Simple
Home page
Categories page
Help page
Intermediate
Restaurants page
Individual restaurant page
Reset password page
Order history page
Payment method page
Complex
Comparison page
My cart page
Sign in page
Sign up page
Profile page
Track order page

Figure 1.2 Complexity ranking for back-end implementation

*Back-end timeline and progress checklist*

Back-end timeline and progress checklist				
Start Date	Duration	Task	Completed	Incomplete
1st July 2022	Two weeks	Research and familiarisation with Node.js, MySQL and Git	Successfully mastered the foundation and basic knowledge of these concepts	Lacked certain in-depth command over these tools
15th July 2022	Two days	Creating the database	Successfully created SQL tables and insertion of data	
17th July 2022	One week	Organising databases and linking them to the MySQL server	Databases were organised and querying of data was successful	
24th July 2022	Four weeks	Recognise and apply access to databases from Node.js code implementation	Access to databases was created successfully	Minor issues faced with certain data input
21st August 2022	Two weeks	Ensuring code consistency, troubleshooting to fix bugs and final review	Team has completed final clean up of code and bug fixes	

Figure 1.3 Checklist for back-end team

## Team Breakdown

The team consists of five members mentioned on the title page at the beginning. We divided the team further into a back-end and front-end team, each with its sub-focus discussed above. By doing so, we can split our manpower to work on the many different segments required in this project. Besides meeting online daily to work on the individual front-end and back-end objectives, the entire team would also meet at each sprint retrospective to discuss progress, request assistance if necessary and collaborate on the project. Below is a table we used to document the notes and summary of each group meeting, including our sprint retrospectives that were organised. It is collated on ClickUp docs so that every team member has access to it when required.

### Group Meeting Log

(Disclaimer: We did not document all of our in-person meetings, but the majority of them were recorded in this log. This was mainly for our own reference, but it is relevant in showing some of our thought processes and discussion contents.)

No.	Date	Attendance	Contributions	Notes & Summary
1	01/07/22	5/5	Kevin, Aditya, Kannappan, Charlotte & Nicole	<ul style="list-style-type: none"> <li>team roles are assigned: 3 main front-end developers and 2 main back-end developers</li> <li>front end team initially proposed to cast the web application using vanilla <code>js</code>, <code>html</code> &amp; <code>css</code> however upon discussion <code>react.js</code> has been decided to be a better fit for the project</li> <li>brainstorm and familiarisation of the tools we will require for this project:  <code>-node.js</code>  <del>-mysql</del>  <code>-react.js</code>  <code>-git branching</code>  <code>-conscious web designs</code>  <del>-clickUp</del> for project managing  <code>-timeline log + objectives checklist</code> </li> <li>completed the installations and set up of necessary software, tools, libraries and git repositories</li> </ul>
2	07/07/22	4/5	Kevin, Aditya, Kannappan & Charlotte	<ul style="list-style-type: none"> <li>back end team has commenced implementation for database systems, APIs, architecture and servers hence one front end team member will render support on this matter</li> <li>2 remaining front end team member will carry on making a draft timeline &amp; objectives log as well as questionnaire for user testing once the first few web pages are implemented</li> </ul>
3	14/07/22 (Sprint Retrospective)	5/5	Kevin, Aditya, Kannappan, Charlotte & Nicole	<ul style="list-style-type: none"> <li>all team members are comfortable with the final decision of which tools we will be using for both the front-end and back-end development</li> <li>team will commence guerrilla user testing 1<sup>st</sup> iteration</li> </ul>
4	28/07/22 (Sprint Retrospective)	5/5	Kevin, Aditya, Kannappan, Charlotte & Nicole	<ul style="list-style-type: none"> <li>front end team will have to make minor changes in the UI design based on user testing feedback received</li> <li>back end team is still working on the database access</li> </ul>

					<ul style="list-style-type: none"> <li>decided to have one front end developer become a full stack developer to assist on both ends to ensure that objectives are cleared according to the timeline we have created</li> <li>full stack developer to assist in back-end development objectives</li> </ul>
5	04/08/22	3/5	Kevin, Aditya & Kannappan		<ul style="list-style-type: none"> <li>both front-end and back-end teams are working towards completion according to plan</li> <li>team to commence second iteration of guerrilla user testing</li> </ul>
6	11/08/22 (Sprint Retrospective)	5/5	Kevin, Aditya, Kannappan, Charlotte & Nicole		<ul style="list-style-type: none"> <li>full stack developer assisting in front-end development for the implementation of some complex functionality</li> <li>team experienced conflict on some design sentiments however upon a fair discussion we came to a conclusion</li> </ul>
7	20/08/22	5/5	Kevin, Aditya, Kannappan, Charlotte & Nicole		<ul style="list-style-type: none"> <li>front-end team on track to completion of remaining web pages and will commence on the written report</li> <li>back-end team on track to completion and will proceed with some troubleshooting of the web application and code clean up</li> </ul>
8	25/08/22 (Sprint Retrospective)	5/5	Kevin, Aditya, Kannappan, Charlotte & Nicole		<ul style="list-style-type: none"> <li>completion of deliverable</li> <li>whole team currently working on the written report and will be thoroughly finalising it before the submission</li> </ul>
9	04/09/22	5/5	Kevin, Aditya, Kannappan, Charlotte & Nicole		<ul style="list-style-type: none"> <li>meeting held online to do a final review of our completed web application and written report</li> <li>project submitted</li> </ul>

Figure 1.4 Summary of group meetings

## Frameworks considered

Here we will discuss in detail the frameworks, database and additional plugins/libraries we considered. It will detail our research and thought process that led to us selecting the best-suited tool for our project.

### The frameworks we considered:

1. React
2. Angular
3. VueJS
4. Ember

Initially, we briefly attempted utilising vanilla JS, HTML and CSS to test the scalability of implementing the actual web application. It was decided that it would have been too time-consuming and inefficient to do so, as we have plans to scale up the complexity of the web application in our future implementations. Hence, a duration of two weeks in our 1st Sprint to research the various frameworks listed above to handle our web application build to enable us the freedom to execute more complex logic into our web applications.

There were some pointers which we have listed down to help us in our decision-making.

### Points to note:

- 1. What is the user popularity of the tools we are considering?**
  - Considering this factor will ensure that the tool we select will consist of a well-established development community proving its reliability as any bugs or errors we might face are likely to have a thread with multiple solutions.
- 2. Does the tool allow for build to be done in separate components?**
  - This was also an important factor as being allowed to build our application in separate components can help maintain the construction process in a simple and more manageable manner. Building in components will be much less complicated when we carry out unit testing since the components are independent of each other. It will also allow our website to be easily updated or maintained as components can be easily replaced or replicated.

Based on the first point, React, Angular and VueJS will be placed at the forefront of our considerations. Since Ember falls short on user popularity, it becomes a risk to use this tool for our build as we might encounter problems and find no solution. Based on our second point of consideration, React, Angular, and VueJS are also top in our pick as it allows for our build to be pieced together with various separate components. Hence, we will first eliminate Ember as a contender and then carry on by listing out some pros and cons of each of the remaining frameworks still in consideration.

### Pros of React:

React is a very flexible tool with many accompanying libraries that extend its functionality. It is maintained by a community of individual developers and Meta, formerly Facebook. With this level of flexibility, developers have free reign over which third-party tool is needed for the task at hand.

As we are developing this web application, we have plans to expand our application into mobile applications, and React-Native is the best option for cross-platform mobile

applications. Taking into consideration that not all users will be using the same operating system on their mobiles.

#### Cons of React:

One of our concerns with react is that the javascript library is evolving quickly, and we will have to be constantly aware of any updates on the most recent changes. This also means that any documentation might soon become irrelevant.

Although React provides flexibility over the features we intend to create, we will still have to maintain any third-party applications independently.

Refactoring in React will be an issue if there is any legacy coding.

#### Pros of Angular:

Angular is a framework, implying that all the features will come built-in, unlike in React, where we have to maintain each third-party application independently.

Google developed Angular and offers Long-Term Support(LTS), which means that this framework will not go obsolete, and many Google applications also utilize the Angular framework. Google also has plans to scale the Angular ecosystem further; hence this promising aspect can be considered by investing in Angular.

As Google is a giant entity of Search Engines, Angular itself is therefore optimized for search engines. The Angular Progressive Web Application is a cost-friendly solution that allows websites to function like mobile applications.

Angular uses Typescript, which is a superscript of JavaScript. This provides higher security which helps programmers catch and eliminate errors in the early stages of the writing process or task maintenance.

#### Cons of Angular:

SEO (Search Engine Optimisation) works best when the web application tool uses server-side rendering (RubyGarage, 2022). It is, therefore, much more accessible to set that up in React than with Angular. Angular has limited SEO options and poor accessibility for search engine crawlers.

#### Pros of VueJS:

Vue might be preferred as it uses HTML templates. Developers also have the option to write in JSX. It is one of the preferred tools to use as it is easy to learn and implement. As this tool uses HTML templates, it can be progressively integrated into existing backend applications without the need for significant refactoring. No extra framework is also required to integrate this code with legacy technologies.

#### Cons of VueJS:

VueJS is not used for large-scale projects as it does not have strong support, and large companies do not financially support the framework; therefore, it lacks scalability. VueJS also has much fewer plugins compared to React or Angular.

While VueJS allows for code flexibility, it can become a double-edged sword in programming. It can lead to irregularities and errors in code as different programmers take different approaches to the same problem.

Based on our thorough research, we have decided to rely on React to implement our web application as it also has the highest rating on StackOverflow (StackOverflow, n.d.). Angular was eliminated from our consideration as it would require us to familiarize ourselves with Typescript on top of what we are already familiar with. Our web application also relies heavily on search engine functions as users will likely require this feature to navigate our web application; hence using React would make the most sense. While acknowledging that VueJS would be a great tool, as it is excellent for refactoring, we are building this web application from scratch. Hence we decided to start with React as it allows our project to be easily scaled into more complex applications in the future.

#### Database considered: MySQL2

Upon heavy consideration, our team will be using MySQL2 to store our databases as it is what we are all already familiar with and would reduce the time taken to brush up on skills required with MySQL2.

## Project Management Tools

### ClickUp

ClickUp is a cloud-based project management tool that strikes a delicate balance between providing the correct number of features while ensuring affordability and user-friendliness (The Daily Egg, 2022). It is a communicative workspace that allows us to assign tasks to specific/multiple members (Figure 1.5), assign comments, create a shared document, set reminders and many more. As seen above, we utilised ClickUp docs to document notes from our group meetings (Figure 1.4). This was one of the many features of ClickUp that we used frequently and has proven to be extremely helpful in our project planning and management. All members could access and assign tasks or set up reminders for other members. This has allowed us to maintain communication and collaborate even while we were working on our tasks individually, keeping the productivity levels of our team high.

The screenshot shows the ClickUp interface with a 'TO DO' section containing one task: 'create access to database' (status: In Progress, due date: Aug 7). Below it is a 'COMPLETE' section containing two tasks: 'complete mysql database querying' (status: Complete, due date: Jul 23) and 'research on database systems, APIs & architectures to use' (status: In Progress, due date: Jul 11). The interface includes a sidebar with 'back-end to do list' and '+ NEW TASK' buttons, and a top right corner with a 'HIDE CLOSE' button.

Figure 1.5 Back-end to do list in ClickUp

With the consolidated list of tasks on ClickUp, we were able to make a burndown chart depicting the number of tasks to complete against our timeline (Figure 1.6, 1.7). The “Ideal” line in the burndown chart is depicted by the optimal rate of progression based on our Gantt Chart created in the initial report. This enabled us to have a very clear gauge on our productivity level as we progress in this stage and to also decide if the need to include more sprints to speed up task completion is necessary. Fortunately, we had planned our timeline well such that we only had minor delays in certain task fulfilments and did not engage the need to include any new sprints which may not have been viable given the limited timeline. To fulfil the delays we experienced any incomplete tasks were extended over to the next sprint which allowed to achieve completion without any major delays in our planned timeframe. This is documented in detail under our deliverables and sprint log (Figure 1.9, 2.0).

### Burndown Chart

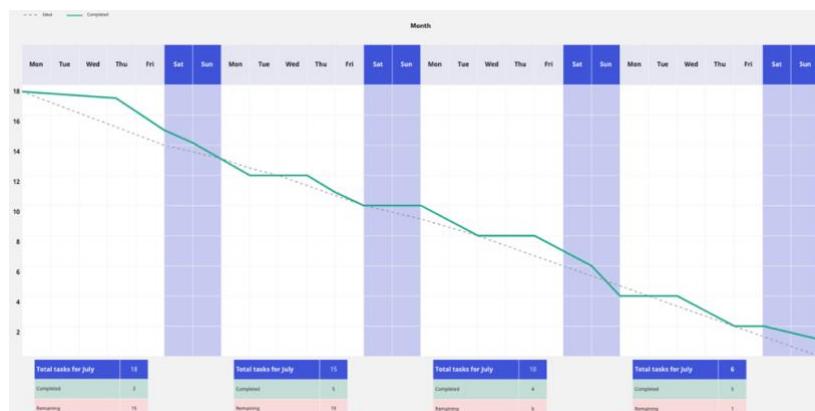


Figure 1.6 Burndown chart for month of July 2022. For a clearer image, visit this [link](#).

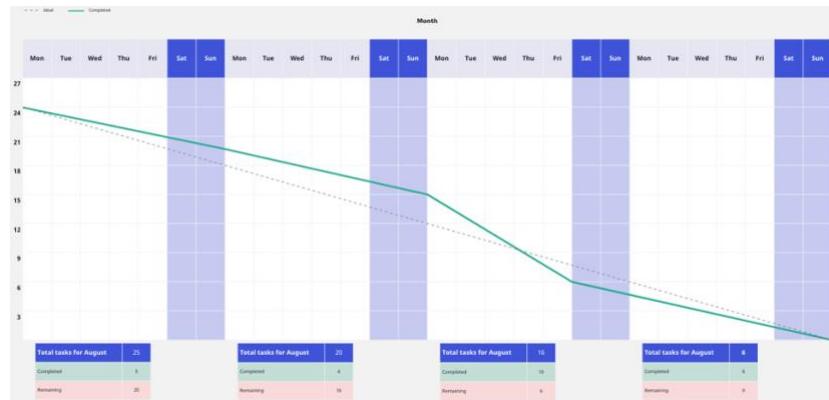


Figure 1.7 Burndown chart for month of August 2022. For a clearer image, visit this [link](#).

## Gantt Chart

In our initial report, we mentioned that we would be using multiple Sprints to develop the intended features in our web application. This would allow our team to have an iterative developmental process to complement the design process mentioned in our initial report (Takeout, 2022). It was extremely helpful to refer to our initial sprint backlog that was planned for in our initial proposal. However, there were some changes due to delays in completing certain tasks during the actual developmental stage, which we will present below. We also documented the new and more accurate sprint timeline in the modified Gantt chart (Figure 1.8) and a new log of our team's sprints with the notable changes (Figure 2.0), which will assist us better in our future developments. Below is our finalised Gantt Chart for this stage of our project.

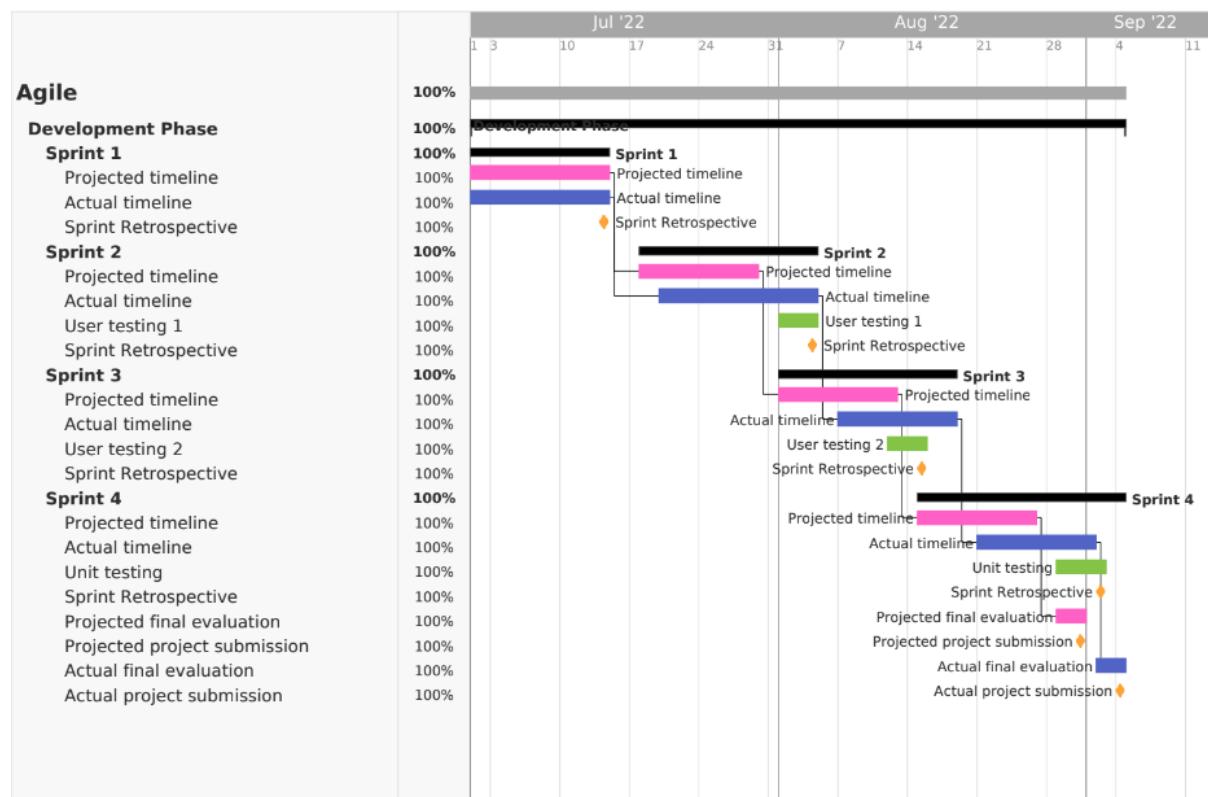


Figure 1.8 Modified Gantt chart for the current development stage. For a clearer image, visit this [link](#).

### *Deliverables and Sprint log*

<b>Deliverables and Sprint log</b>			
<b>Sprint 1</b>	<b>Sprint 2</b>	<b>Sprint 3</b>	<b>Sprint 4</b>
Research on required back-end, front end tools and GitHub integration	Begin database creation	Database querying and access	Track order page (front-end + database linking)
Select database management system and install front end development environment/interface/libraries	Individual Restaurant page (front-end + database linking)	Payment methods page (front-end + database linking)	Implement media queries (front-end)
Home page (& navigation bar & search bar)	Restaurants page (front-end + database linking)	Comparison page (front-end + database linking)	Complete remaining functionalities
Categories page (front-end + database linking)	Reset password page (front-end + database linking)	My cart page (front-end + database linking)	Unit testing
Help page (front-end + include help page content)	Order history page (front-end + database linking)	Comparison page (front-end + database linking)	Include evidencing (tables, logs, charts, code screenshots) in the final report
Sign in page (front-end)	User testing 1 and evaluation	Profile page (front-end + database linking)	Final review and evaluation of project as a whole
Sign up page (front-end)	Implement improvements based on user testing findings	Sign in and sign up page (database linking only)	-
-	-	User testing 2 and evaluation	-
-	-	Implement improvements based on user testing findings	-
-	-	Implement media queries (front-end) and commence working on the final report	-

Figure 1.9 Log documenting task to be fulfilled at each sprint

Below are some of the features we had trouble fulfilling in its original sprint designation. To accommodate the additional time required, they were extended over to the next sprint to ensure that we did not face any significant delays in our timeline to completion.

<b>Delayed functionality</b>	<b>Planned sprint → New sprint</b>
Home page search bar functionality	Sprint 1 → Sprint 2
Categories page database linking only	Sprint 1 → Sprint 2
Restaurants page database linking only	Sprint 2 → Sprint 3
Individual restaurants database linking only	Sprint 2 → Sprint 3
Implementation of media queries	Sprint 3 → Sprint 4
Comparison page database linking only	Sprint 3 → Sprint 3, 4
Order history page database linking only	Sprint 3 → Sprint 3, 4
Sign in and sign up page database linking only	Sprint 3 → Sprint 3, 4

Figure 2.0 Table documenting changes seen in our sprints

### Research – Summative (Initial report)

#### Project.

The document above shows the previous user, market research we conducted and our various findings.

## Research – Current

To further substantiate our project at this point, we will conduct a SWOT analysis of the functionalities that similar competing applications have. This will allow us to take an analytical approach to identify their strengths and weaknesses and identify any possible risks we can avoid. With this research, it will be essential when we decide on the functionalities we will implement in our product (discussed in detail under ‘Features to implement’ in the prototyping and iteration segment).

### Strengths:

Many applications that are similar in functionality to ours rely highly on curated advertisements to boost their popularity amongst users. Such advertisements are often seen on popular social media platforms such as Tiktok, Instagram or Youtube. By targeting these popular platforms among the younger demographics, these are also the same users who are more comfortable with the use of technology and, therefore, more likely to consider making their food orders online through an application. One thing in common that all our competing applications offer would be discounts or promotional codes for new users of the application. This is a sure way to gather a new crowd of users who are not already familiar with the brand to give it a chance to impress with its services. The research on these strengths will provide us with better insights on how to improve our product in the future.

### Weaknesses:

We noticed that users of similar brands and applications are subject to unavoidable hikes in delivery fees, especially during peak hours such as lunch, dinner or during bad weather conditions when food delivery seems to be the best option. This creates a negative view of the application where users become less likely to make orders during these peak periods to avoid the high cost of delivery and this, in turn, drives down the revenue of these brands. During such opportunities (e.g. lunchtime, dinner time), these companies can receive high volumes of food delivery orders.

### Opportunities:

We considered that the one thing these applications do not offer their users is the freedom of choice since there are no other options than to pay for the high delivery prices to have food delivered to their doorsteps. Hence, we thought of a solution that could improve the situation for users, which is to implement a comparison function where they can explore different similar services and perhaps settle on an option less costly than the others. Some may argue that the prices will remain competitive as these businesses are, in fact, competitors. However, we believe that allowing users to compare the different price points will create an understanding amongst users that they are not being unfairly overcharged.

### Threats:

A common threat for users that stands while using such online delivery services is security concerns. Users are at risk when they provide personal details such as card details for payment or personal addresses for deliveries. We believe that a priority to implement security measures should be put in place for all brands that provide such personal services to protect their users from any vulnerability.

## Prototyping and Iteration

### Low Fidelity Wireframe (Concept vs Proof-to-concept evaluation)

This was one of the low fidelity wireframes we created, for the full wireframes please refer to this [document](#) of our initial report.

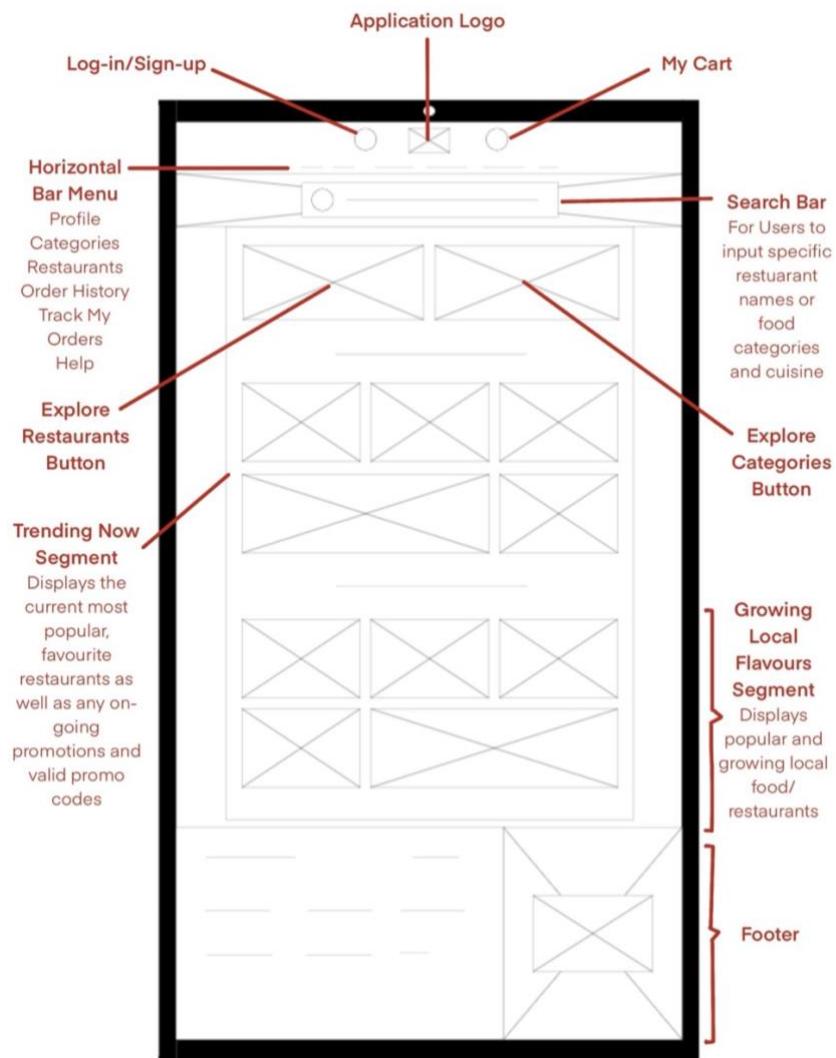


Figure 2.1 Home page low fidelity wireframe

This was the first concept of the web application we had drawn out. It gave us a much clearer view of the web application we were going to build. The low fidelity wireframes were used to eventually map out our medium fidelity wireframes and high fidelity mock-ups. How we created the low fidelity wireframes were based on our documentation of features to implement and how they should be organized on a web page. We studied various popular competitors' web applications to garner ideas on implementing our parts in the most user-friendly manner. The low fidelity wireframes are just a skeletal outline of how our navigation bar, header, banners and content should be placed and organized in our web design. This helped showcase the features we will implement and let the entire team know what our web application should look like. At this stage, there is still no documentation of any colour palettes or design sentiments as it is solely used as an outline template.

## Medium Fidelity Wireframe (Concept vs Proof-to-concept evaluation)

This was one of the medium fidelity wireframes we created, for the full wireframes please refer to this [document](#) of our initial report.

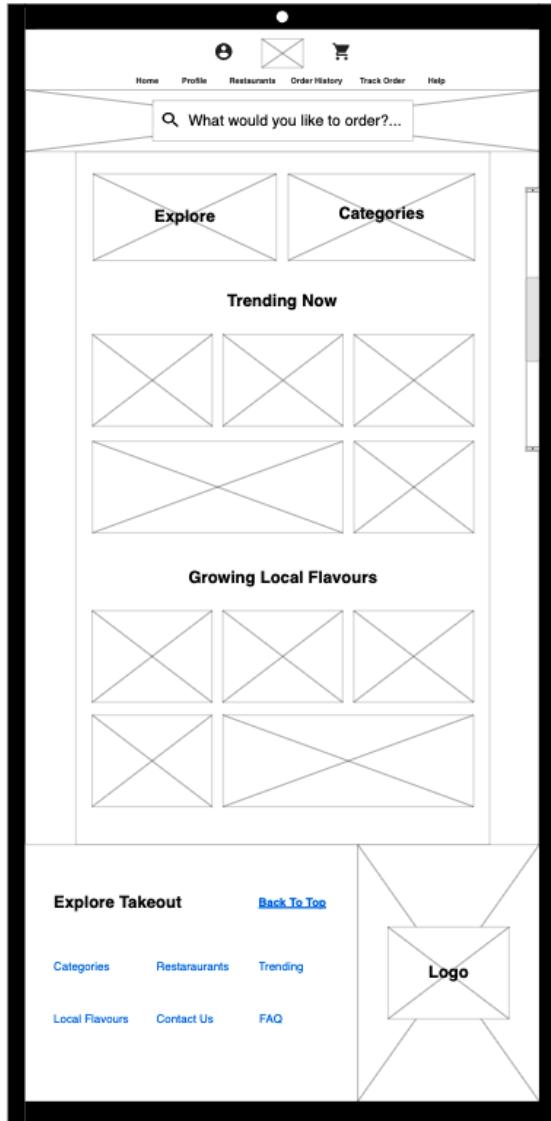


Figure 2.2 Medium fidelity wireframe of our home page

This is one of our medium-fidelity wireframes, which is an updated version of our low-fidelity wireframes with minimal changes. It showcases all the navigation links and buttons which was not identified in the low-fidelity version. This is a helpful guideline for our front-end team when it comes to code implementation as it contains all the different buttons, icons, header and navigation links placement. There is still no decision on colour palettes and user interface-centric considerations as we decided that it is not yet necessary.

## High Fidelity Mock Up (Concept vs Proof-to-concept evaluation)

This was one of the high fidelity mock ups we created, for the full mock ups please refer to this [document](#) of our initial report.

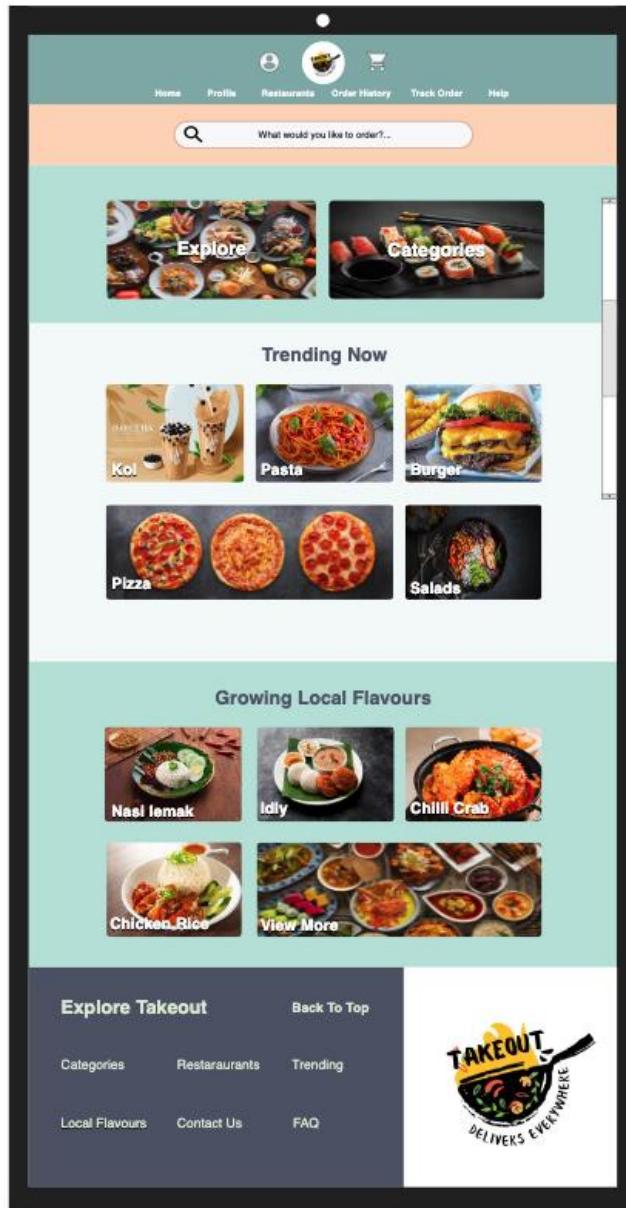


Figure 2.3 High fidelity mock-up of our home page

This was our final mock-up, which fully mimics how the actual web application should look. This version includes findings in our study telling us what functionality users would like to see. We have also done a [research](#) on colour psychology and how it would impact an individual's perception of food which affects the decision-making of our web application's colour palette. Our logo design decision was based on studying popular fast food chains/restaurants' sentiments on what made their logo so iconic so we could design it using similar principles. This mock-up consists of all our final design ideas and the colour scheme. The front-end team will frequently refer to this mock-up when implementing the web application.

In our initial report, we conducted a user feedback survey to ask users what were some of the features they would like to see improved in existing food delivery applications and what new features they want in food delivery platform services. For the complete survey and evaluation, visit this [link](#). Based on that study, we were able to consolidate a complete list of functionality we would like to include in our software at this stage of the project, as well as our future developments.

List of features we intend to implement on our online platform:

- CSS Responsive Web Pages
- Search Engine
- Navigation System
- User Account Registration System
- Sign In / Sign Out Verification System
- Create / Edit Customer User Profile (E.g Name / Email / Username / Password)
- Add / Edit / Remove Customer Delivery Address (E.g Street Name / Block Number / Postal Code)
- Compare Prices Between Delivery Apps & Delivery Timings
- Compare Discounts Between Delivery Timings
- Advance Order System
- Add To Cart System
- Redeemable Reward Points System
- Redeemable Promo Codes System
- Add / Remove Customer Payment Methods (E.g Credit / Debit Card / E-Wallet)
- E - Wallet Payment Method
- Payment Systems
- Track Delivery Order System
- Order History

Figure 2.4 List of features we intend to implement adapted from our [initial report](#).

## Features to implement

What features are we implementing at this stage and why?

1. CSS Responsive Web Pages
  - This was one of the first considered features as we believe it is an essential criterion for a launched web application to have responsive features. Since this will be the first iteration of our fully established web application, it should be as fully responsive as possible for users to begin interacting with our program.
2. Search Engine
  - A search engine within the web page is one of the most used features; hence we figured it would be necessary first to implement this. 62% of consumers turn to search engines when they want to learn more about a new business, product, or service — and 41% use them when they're ready to buy (Zheng, 2018). As this will only be our demo launch, there will only be a single restaurant data. Hence this functionality will be fully functional when we gain new APIs from our partnering food delivery brands.
3. Navigation System
  - A fully responsive webpage can only be possible with a functioning navigation system. Individuals are also more inclined to return as web application users when they find the web page smooth and easy to use. Navigation is one of the most critical aspects of product design as they rely on the navigation to find content and features; therefore, helping users navigate should be a top priority for every app or website. (Babich, 2020).

4. User Account Registration
  - This feature will allow users to utilise our web application's primary function, which is to make food orders and compare the prices for various delivery options; hence, it is also a must to implement at this stage.
5. Sign In/Sign Out Verification System
  - This functionality will go hand in hand with the feature mentioned in point 4. It is for Users to be able to log in to an account before proceeding with making any orders.
6. Compare Prices Between Delivery Apps & Delivery Timings
  - This is the primary function and core of our web application. We will begin implementing this feature to its best and perform user testing post-launch to see what improvements can be made. Our users must understand the purpose of using our web application; hence we will be implementing this at this current stage of the project.
7. Add To Cart System
  - This is a function that allows our users to add their options to their cart, after making their selection and comparisons. By implementing this function, users can then proceed to add payment methods or make a payment if they already have any existing ones.
8. Add / Remove Customer Payment Methods (E.g Credit / Debit Card / E-Wallet)
  - This feature allows our users to add payment methods to make their orders on our web application. If this function is not implemented, we will not be able to collect any actual data to see if users are using our program. Hence, we will be implementing this as well.
9. Order History
  - This feature will require us to store data that users have input. It should not take up too much effort to implement. Hence it was decided that we could include this feature in this iteration of the web development. Users can view any orders they have made with this functionality.
10. Track Delivery Order System
  - We plan to implement this feature as most of our similar competing applications have this function. However, we also understand that it may be too difficult to accomplish at this project stage since it requires linking up multiple databases.

It is our goal to implement as many features as possible while maintaining a reasonable timeline for the launch of our web application. Our team believes that by implementing more features, we will be able to gather more valuable data from users when they test out the web application. This will give us plenty of opportunities to improve our program and hopefully ideate some new development for the future. The implementation of the said functions above was taken into massive consideration in planning our timeline to ensure that we do not overestimate the time and effort we will require since we are working against a fixed deadline. For the features listed, which we did not implement at this stage of the project, we decided that it is not crucial to have at this point; hence will be planned for in our future developments.

From here, we will discuss in detail the design sentiments of our web application and what was taken into consideration during this process.

# Design

## Use Case Model

With the new list of features which our team will be implementing in the current timeline, we came up with a new use case diagram (Figure 2.5) to help us identify the possible interactions our users should have access to with and without logging in to an account. It summarizes the details of the system's users, allowing us to understand the system's developers better. Using a use case model, we hope to define and organize the requirements making it easier for our team to refer to during our developmental process. With this, we can proceed with code implementation to produce the first iteration of our web page and conduct user testing, which we will discuss further in the next segment.

## Use Case Diagram



Figure 2.5 Use case diagram for logged in and non-logged in users

## User Testing

Based on our high-fidelity mock-ups from the initial report, the front-end team began the development on the first three web pages: The home page, the Categories page and the Help page. The decision to first implement these pages was based on the complexity level required to build each web page, which was documented in the table above (Figure 1.0). Our team has decided that it would be more efficient first to implement the pages with more straightforward code implementations and then proceed with the first iteration of our user testing. This will ensure that we minimize the subsequent effort required to make improvements on the web pages and hence do not exceed the designated timeline for each task keeping the development on track.

At each milestone of the web pages we implement, we will conduct an iteration of user testing to ensure the overall quality of the web application.

## Guerrilla Usability Testing

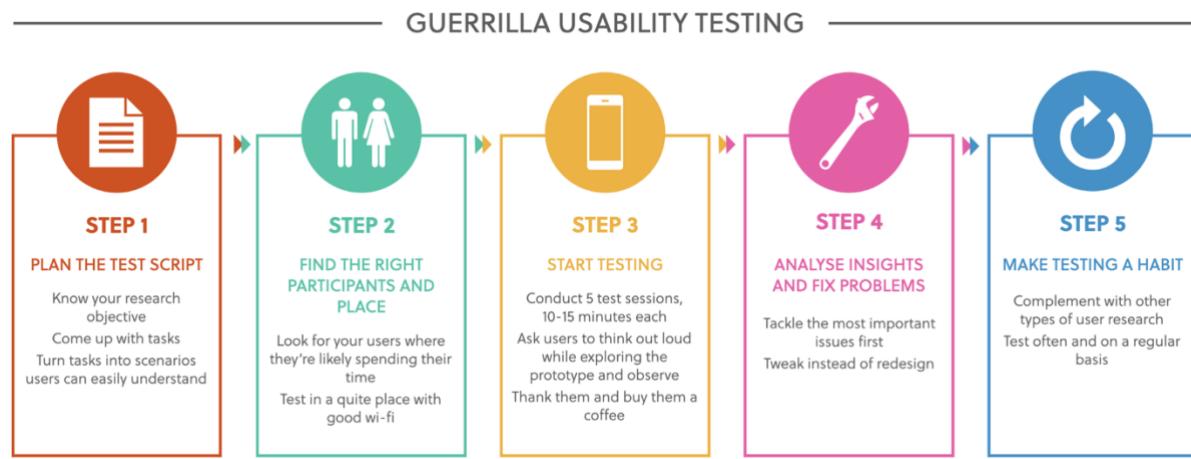


Figure 2.6 Guerrilla Usability Testing

Our team has concluded by using guerrilla usability testing (Figure 2.6) to evaluate our web application development. Guerrilla testing gathers user feedback by taking a design or prototype into the public domain and asking passers-by for their thoughts (Chesters, 2017). Guerrilla testing allows us to garner a response from randomly selected individuals effortlessly. From there, they can provide us with a fresh perspective on the design and usability of our web application which we can use to make further improvements. While considering that some participants may not be our targeted users, we also believe chances are that everyone will be a user of a certain web application and can hence offer us a variety of insights based on their unique experiences. The goal of such testing is to inform design decisions for ongoing projects and identify usability issues rather than assess the usability of an already existing interface (Adiseshiah, n.d.) which is the case for our project. Based on google statistics research, 85% of core usability problems can be found by observing five people using the application (Google Developers, 2015). Therefore, we have discovered guerrilla testing to be highly suited to the current stage of our product development.

## User testing – 1<sup>st</sup> iteration

For the first iteration of Guerrilla usability testing, it was conducted in our school, where six individuals were randomly recruited to test the first three web pages we have implemented (Home page, Categories page & Help page). This was done during Sprint 2 (refer to Figure 1.9) of our project timeline plan.

### Takeout user testing (Iteration 1)

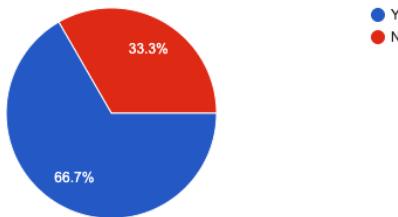
Please navigate between the home page, categories page and help page to provide us with some user feedback.

Figure 2.7 Title for the 1<sup>st</sup> iteration of guerrilla user testing

1. I enjoy the design of the home page.

6 responses

 Copy



If you answered "No" to question 1, why?

2 responses

The icons beside the logo was hard to notice

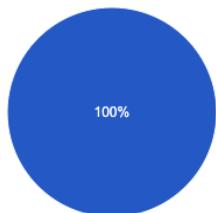
It was hard to read 'Explore' and 'Categories' in the first two images on the page

Figure 2.8 66.7% of the participant users appreciated the design of the home page and out of the 33.3% of users who answered 'No' have also provided their feedback to why they did not feel the same.

2. I find the layout of the home page easy to navigate.

6 responses

 Copy



If you answered "No" to question 2, why?

0 responses

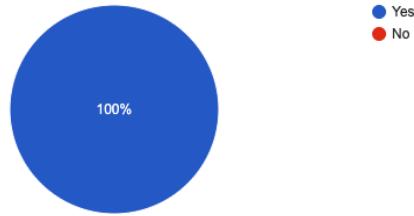
No responses yet for this question.

Figure 2.9 100% of participant users found the home page easy to navigate

3. The options found on the navigation bar is sufficient.

[Copy](#)

6 responses



If you answered "No" to question 3, why?

0 responses

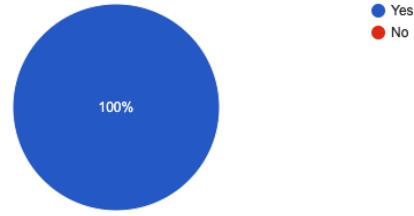
No responses yet for this question.

Figure 3.0 100% of users found the functionality in the navigation bar to be sufficient

4. The design of the categories page was easy to navigate.

[Copy](#)

6 responses



If you answered "No" to question 4, why?

0 responses

No responses yet for this question.

Figure 3.1 100% of users found the categories page easy to navigate

6. The help page was easy to locate.

[Copy](#)

6 responses

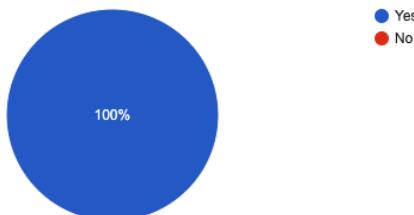


Figure 3.2 100% of participant user agreed that the help page easy to locate

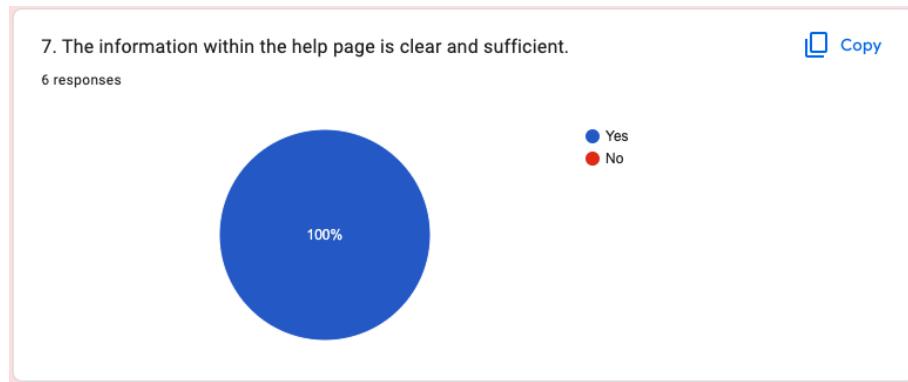


Figure 3.3 100% of participant users felt that the information we have provided in the help page was conclusive and sufficient

#### Conclusion of the 1<sup>st</sup> iteration

To conclude this iteration of testing, the majority of the participating users have appreciated the designs of these three web pages, with two exceptions of the home page, as seen in Figure 2.8. This data allows us to rule out the exceptions to implement improvements but still conclude that the design sentiments in the high-fidelity mock-ups from our initial report were successful. The testing has also concluded that navigation within these pages was of good user-friendliness, implying that any users of our web page will have little to no issue, even if they are new to our web application. This information allows us to substantiate that the current formatting of the web design has cleared the mark for user-friendliness and that we will not need to make any improvements in this segment. We have also asked participants if they could locate the help page (Figure 3.2) quickly and if the information within the page was concise and sufficient (Figure 3.3) for them as users. The functionality of the help page within our web application assists users in cases where they require assistance with any issues involving the web application or its functionality. 100% of participants agreed that it was sufficient, and we can gather that the web application will be able to serve the users in this manner.

Some issues were raised during our sprint retrospective at this point by a few of our members, which is that the current number of participants may not be entirely sufficient to substantiate the accuracy of the data. Also, although two constructive feedback were received regarding the design of our home page, they were unique and would not be substantiated enough to probe a change in our design. Hence, as a team, we have decided to conduct the 2nd iteration of testing, which will mimic the 1st iteration but with added questions to further probe users if the mentioned issues were notable or significant. This will be conducted in Sprint 3 (refer to Figure 1.9) of our project timeline plan.

## User Testing – 2<sup>nd</sup> iteration

As mentioned above, we have improved our user testing methods to gather better data for our web pages. In this 2nd iteration, we repeated some of the questions from the 1st iteration, asking if the users enjoyed the web page's design, found that the web pages have sufficient clarity and if they also found the functionality within that page to be adequate. There were some new questions included, and we also increased the number of testers to 13, which is double the 1st iteration of testing. By doing so, we can scale down the testing into a few main criteria for the users to answer against. This prevents us from overcomplicating the testing while still being able to gain essential feedback from our users.

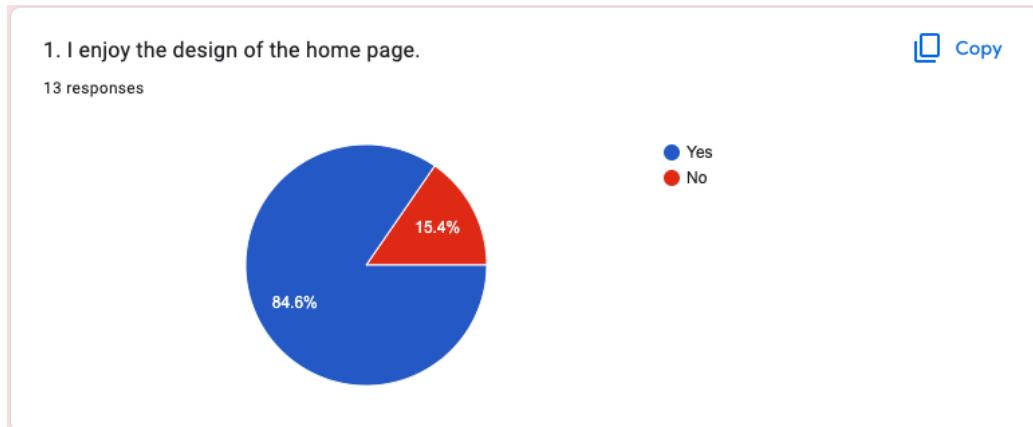


Figure 3.4 84.6% of users enjoyed the design of our homepage

2. Are the icons and navigation bar at the top of the home page clear for you as a user?  
7 responses

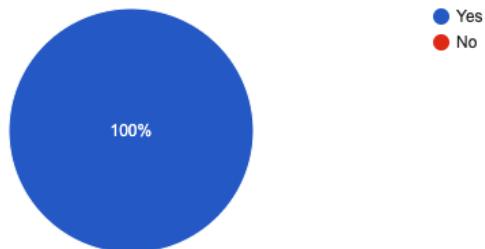
- The icons did not look like buttons
- Okay
- I could not tell the icons next to the logo was buttons
- The icon beside logo looks weird but the nav bar is clear
- OK
- Not really
- Looks fine

Figure 3.5 Responses to clarity of icons, buttons and navigation bar

3. I find the layout of the home page easy to navigate.

 Copy

13 responses



If you answered "No" to question 3, why?

0 responses

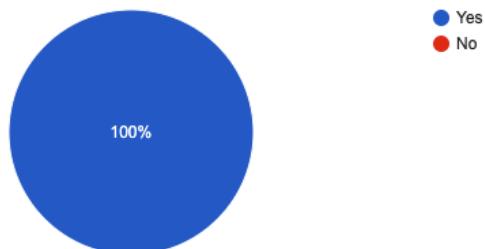
No responses yet for this question.

Figure 3.6 100% of users found the layout of the home page easy to navigate

3. The options found on the navigation bar is sufficient.

 Copy

13 responses



If you answered "No" to question 3, why?

0 responses

No responses yet for this question.

Figure 3.7 100% of users found the navigation bar to be sufficient

**4. Are there any text, buttons in the home page that you find difficult to see?**

7 responses

I could not see the words 'explore' and 'categories' clearly

No

No its fine

The explore and categories word blends into the image and is hard to see

Some of the texts lack contrast against the images, it is hard to see

Yes some of the words in the images cannot be seen clearly

Some words are hard to read against the pictures

Figure 3.8 Responses to clarity of texts in the home page

**5. The design of the categories page was easy to navigate.**

 Copy

13 responses



● Yes

● No

If you answered "No" to question 5, why?

0 responses

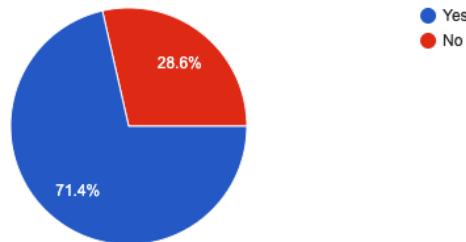
No responses yet for this question.

Figure 3.9 100% of users found the categories page easy to navigate

7.Did you enjoy the design of the help page?

7 responses

 Copy



If you answered 'No' to question 7, why?

2 responses

The design looks very simple and not blended

No the help page boxes does not look very nice

Figure 4.0 71.4% of users enjoy the design of the help page but 28.6% of users did not

7. The help page was easy to locate.

13 responses

 Copy

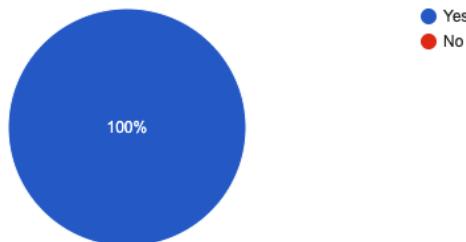


Figure 4.1 100% of users found that the help page was easy to locate

8. The information within the help page is clear and sufficient.

13 responses

 Copy

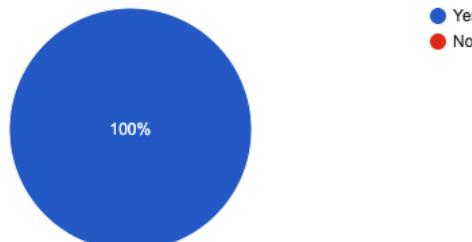


Figure 4.2 100% of users found the information within help page was sufficient

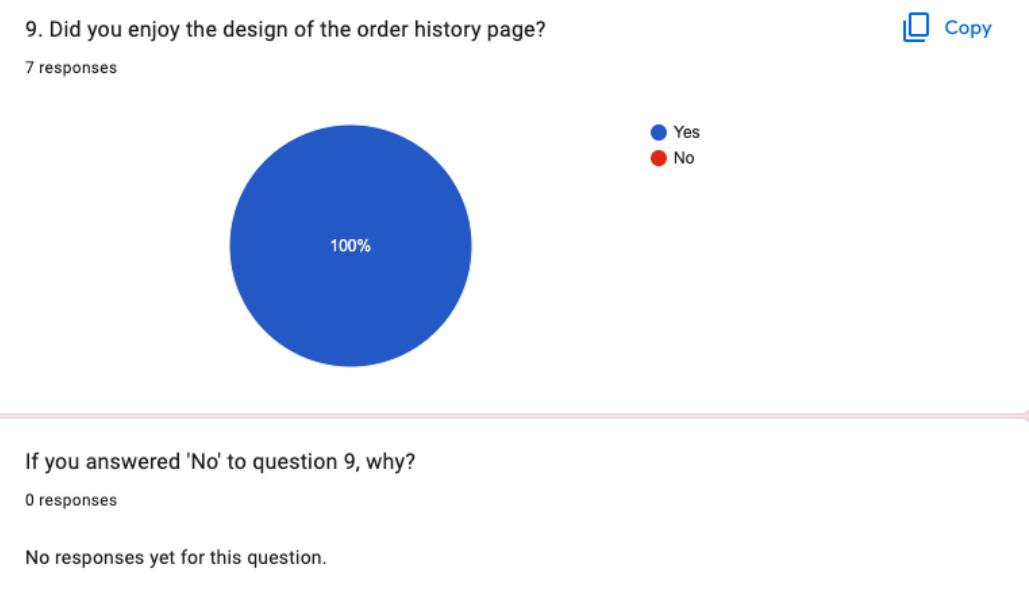


Figure 4.3 100% of users enjoyed the design of the order history page

#### Conclusion of the 2<sup>nd</sup> iteration

To conclude our 2nd iteration of testing done in Sprint 3, we identified some concerning issues we needed to rectify in our design. The first concern was that users could not identify the icons at the top of the home page as buttons. This creates a massive problem for us as users will then struggle with navigating between the different functions of our webpage. Another concern was that on the home page, some of the images containing text titles have clashing colours, reducing the contrast and making it difficult for users to read.

## User Testing – 3rd iteration

This 3rd iteration of testing was not planned for in our initial Sprint. However, our team collectively decided that we could afford to organize this 3rd iteration in our 4th Sprint; hence, we would like to garner more feedback for the additional web pages we have finished implementing.

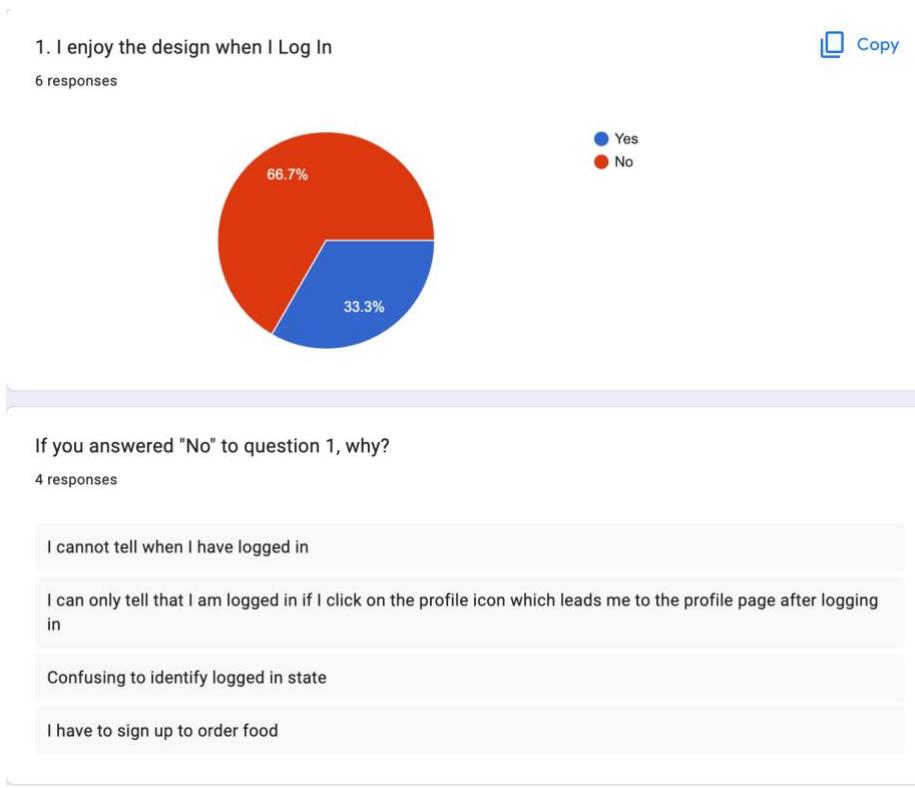


Figure 4.6 There were some concerns regarding the lack of a log-in state for users

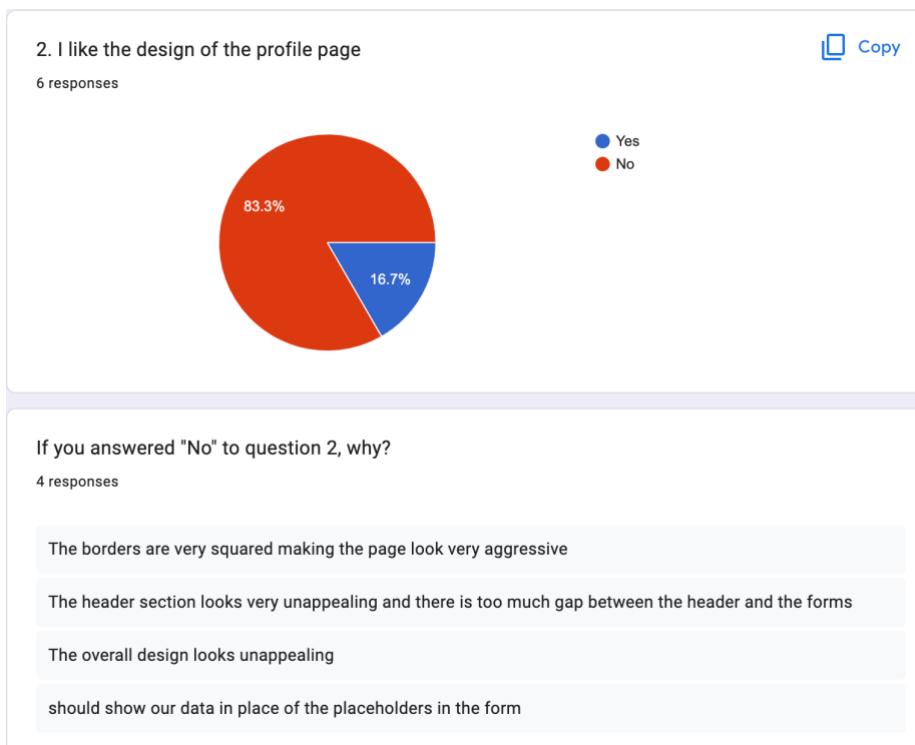


Figure 4.7 There were some user concerns regarding the design of our profile page

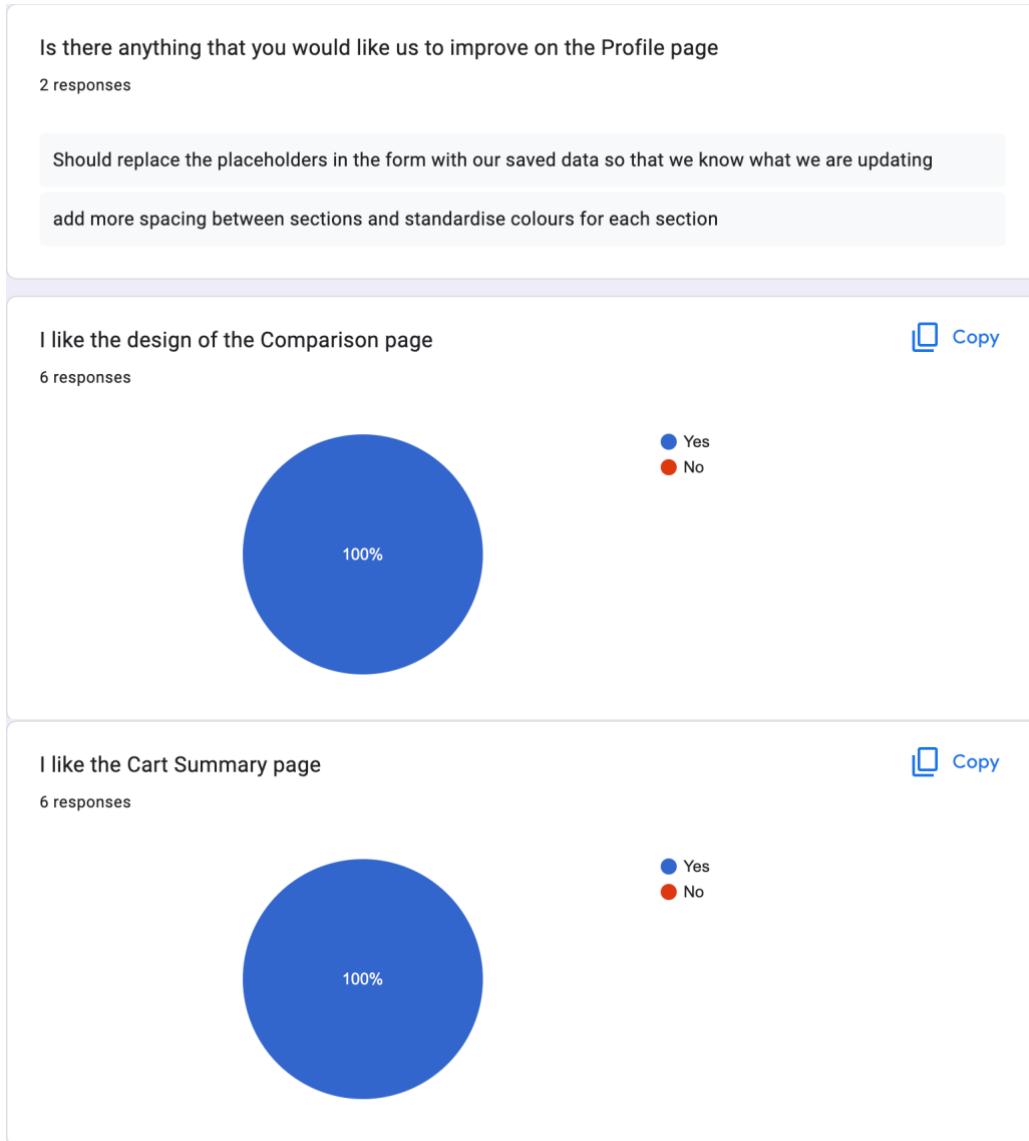


Figure 4.8 User feedback for improvements to be made to the profile page and 100% responses for the design of our comparison and cart summary page

#### Conclusion of the 3<sup>rd</sup> iteration

Based on this 3rd iteration, we received much constructive feedback on how to improve the design and functionality of the profile page in our web application. There was a lack in some design aesthetics and essential user interface criteria (lack in displaying user data on the profile page even when users are already logged in). However, we are glad that most users appreciate our web application's design sentiments on the remaining web pages. Therefore, we decided not to implement any further testing at this point as the information we have currently received is sufficient. There is also the consideration that we are working against a deadline and did not pursue further usability testing. We do hope to be able to carry out different methods of usability tests in our future developments to enhance our program further. We will be documenting the changes we made to our design based on the various feedbacks received from all three counts of user testing.

## Design Changes – between prototypes

### Changes based on User Testing feedback:

On the home page, icon buttons beside the logo were unclear for some users (Figure 3.6).

Before:

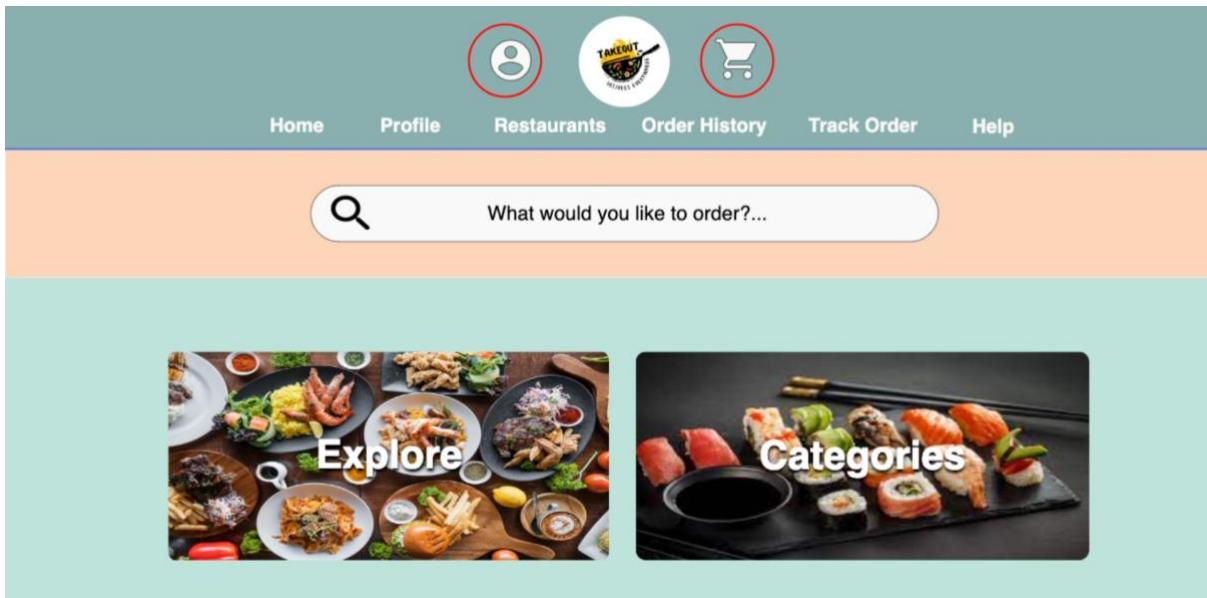


Figure 4.9 Icon before changes

After improvements:

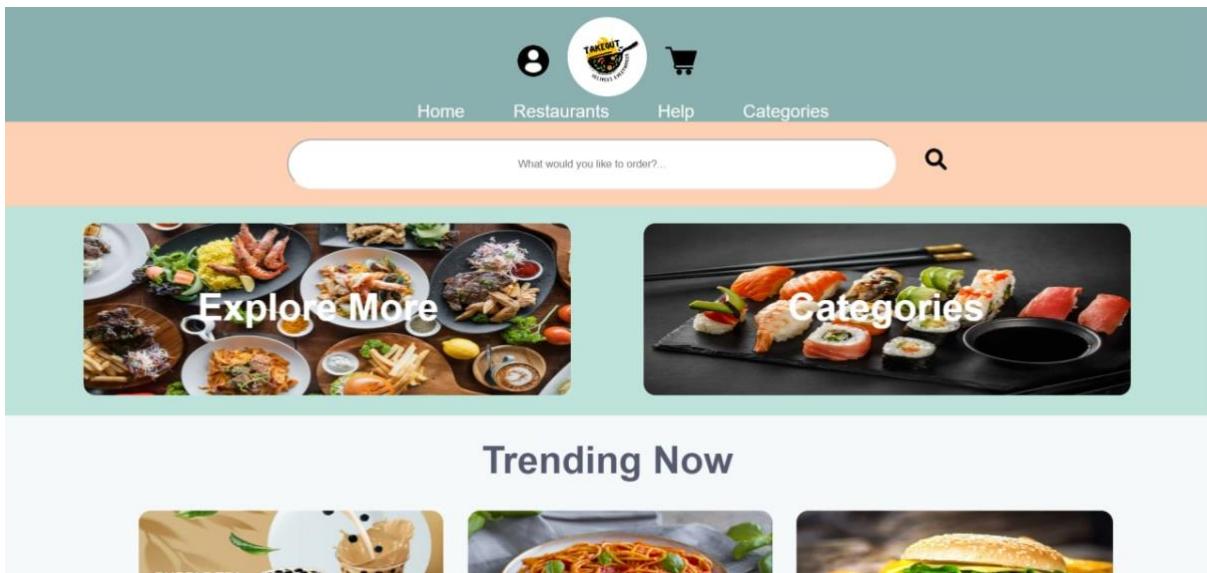


Figure 5.0 Newly designed icons

We simply switched up the colours of the icons from white to black to create a contrast from the logo's colour. The new design helps the icons to stand out more such that users are able to recognize them as clickable buttons.

On the home page, some texts on the images are not clear for the users (Figure 3.9).

Before:

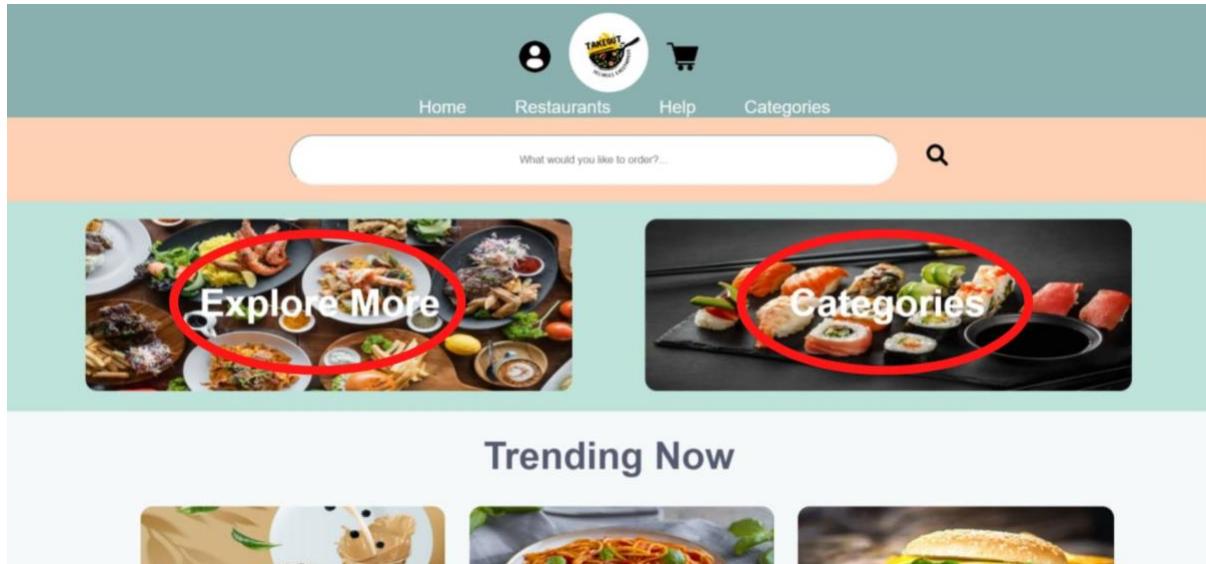


Figure 5.1 Unclear texts on the home page

After improvements:

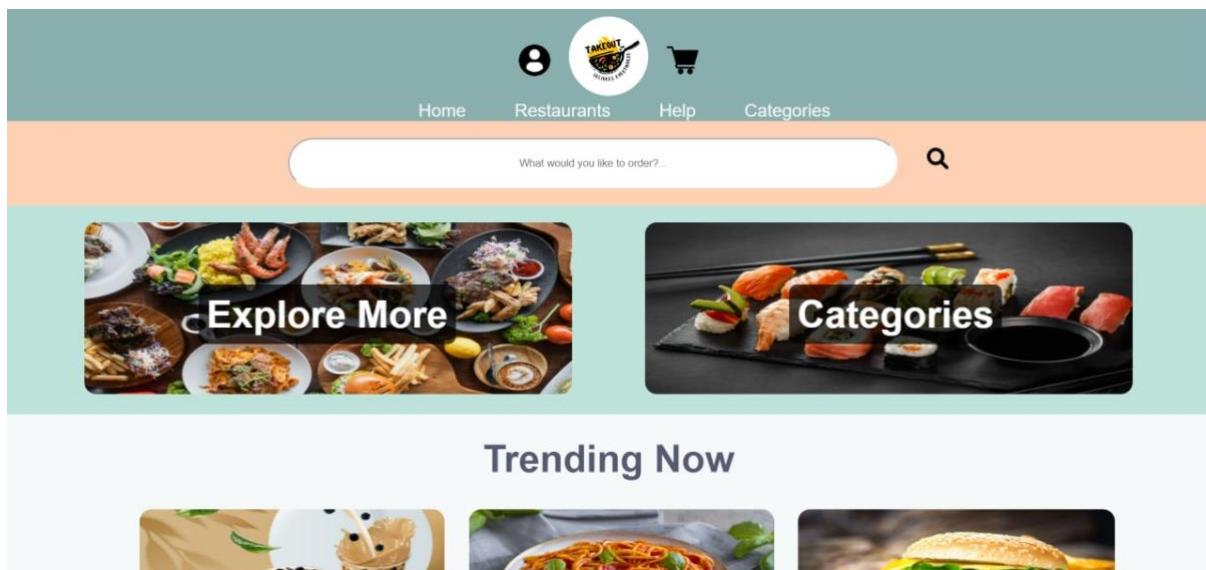


Figure 5.2 Added overlay on text to create contrast for a clearer view

We added a translucent black overlay to the text such that there is a contrasting background against the images so that users are able to see the texts much clearer.

A percentage of users found that the design of the help page was too simplistic (Figure 4.2). Some of our team members also raised the same concerns during our sprint retrospective; hence we decided to implement some changes to the design.

Before:

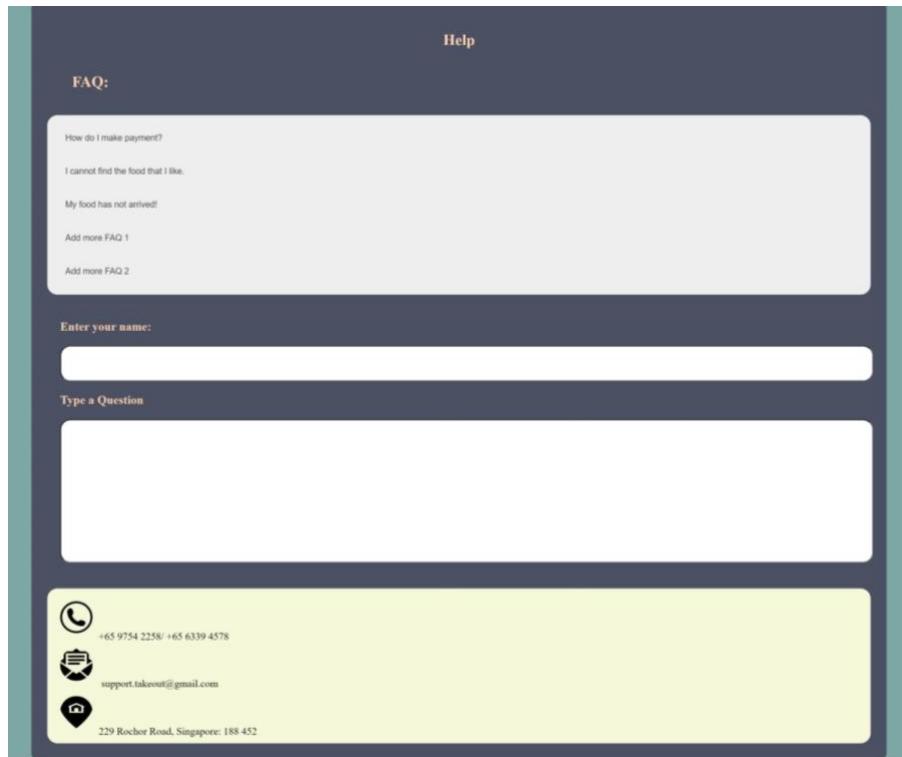


Figure 5.3 Original design of help page

After improvements:

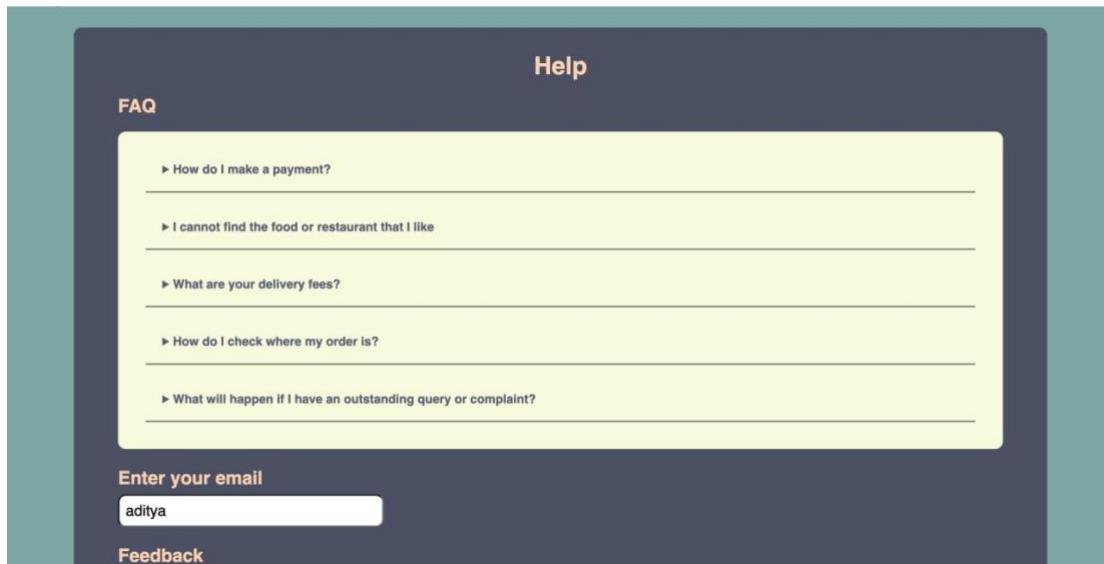


Figure 5.4 New and improved design of the help page

We tried to modify the look of the help page as some user feedback and internal feedback, found that the original design lacked some depth. Hence, the new creation has a much more cohesive colour scheme than the rest of the page, making it more enjoyable to look at.

Previously, when users were logged in to their accounts, the details of the form on the profile page would still be empty. We realised that this is highly inconvenient to all users who want to view their profiles; hence we implemented for these forms to be pre-filled with the user's data when logged in. Another issue was that the profile page's aesthetics lacked some depth. Hence we have rounded the borders and coordinated the colour scheme of the elements on the page to make it more soothing for users to view.

Before:

The figure consists of four panels arranged in a 2x2 grid. The top-left panel shows the 'Profile' section with fields for Name, Username, Password, Confirm Password, Email, and Phone Number, all of which are currently empty. The top-right panel shows the 'Address' section with fields for Street Name, Block Number, Building Name, Level Number, Unit Number, and Postal Code, also all empty. Both of these panels are circled in red. The bottom-left panel shows the 'Add New Card' section, which includes fields for Card Number, Expiry Date, CVC/CVV, and Card Holder Name, all of which are empty. The bottom-right panel shows the 'E-wallet' section, which includes fields for Amount (\$0.00), Balance (\$1), and Auto Top-Up, also all empty.

**Profile:**

Name  
Username  
Password  
Confirm Password  
Email  
Phone Number

Security Question (What was the first school that you studied in?)  
Security Answer

[Update Profile](#) [Log Out](#)

**Address:**

Street Name  
Block Number  
Building Name  
Level Number  
Unit Number  
Postal Code

[Save](#)

**Add New Card:**

Card Number:  
0000 0000 0000 0000

Expiry Date:  CVC/CVV:

Card Holder Name:  
Enter Card Holder's Name

[Add Card](#)

**E-wallet:**

Visa

Amount: \$0.00 [Top-Up](#)

Balance: \$1

Auto Top-Up:

Figure 5.5 Empty forms in profile page and rough aesthetics

After improvements:

The screenshot displays a mobile application interface with a light teal header and footer. The main content area is divided into several sections:

- Profile:** Contains fields for Name (Kevin Royce Thomson), Username (KevinRoyze), Password (123456), Email (kevinrt001@mymail.sim.edu.sg), and Phone Number (93973129). Below these is a Security Question field asking "What was the first school that you studied in?", with the answer "Clementi Primary".
- Address:** Contains fields for Street Name (Clementi Ave 3), Block Number (555), Building Name (Clementi Crest), Level Number (1), Unit Number (#39-210), and Postal Code (126555). A "Save" button is located at the bottom right of this section.
- Add New Card:** A form for adding a new card, including fields for Card Number (0000 0000 0000 00), Expiry Date, CVC/CVV, and Card Holder Name.
- E-wallet:** Shows a balance of \$100, an Auto Top-Up checkbox, and a VISA logo.
- Reward Points:** Displays a balance of 50 points.
- Promo Codes:** Lists five promotional codes: TK123, TK456, TK789, TK987, TK654, and TK321.

Figure 5.6 New profile page with pre-filled user data and improved design elements

Design changes based on team's opinion (during Sprint retrospectives, Figure 1.4):

As a team, we have identified a couple of conceptual design problems and decided to rectify them. One of the problems was the lack of depth in the design of our help page mentioned above, which was also backed up by user testing feedback (Figure 4.2). The second problem we identified by ourselves was that in our home page's navigation bar, there was a duplicate of the same navigation link to the profile page. One is an icon, and one is an option in the navigation bar; hence, we decided to remove the option from the navigation bar to make it less confusing for any users.

Before:

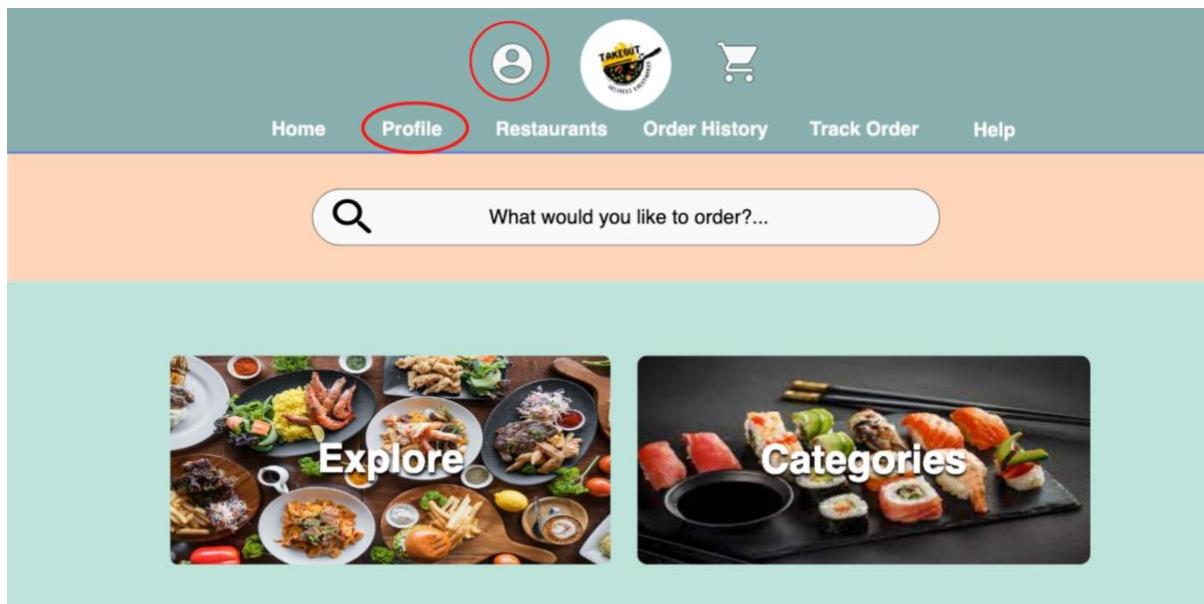


Figure 5.7 Duplicating navigation links to profile page

After improvements:

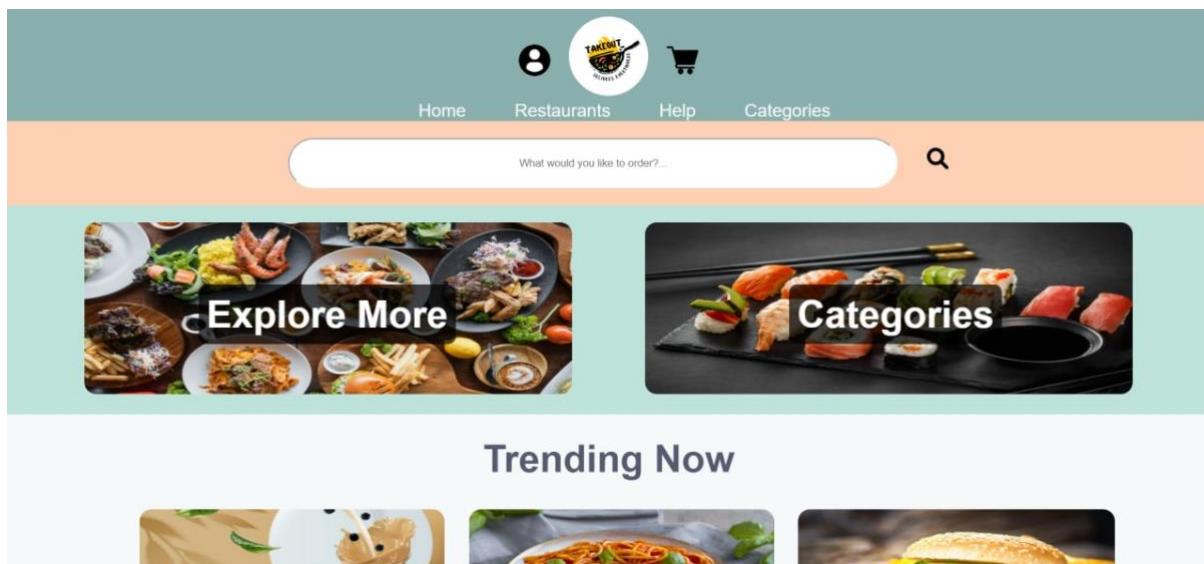


Figure 5.8 'Profile' option in navigation bar removed

During our third Sprint retrospective, some team members identified that our web page lacks a design sentiment that tells users that they are already logged into an account. Hence, we decided to add a design element that says “Welcome -username-“on our home page for the user’s awareness. Before this implementation, we realized that there was no way for users to know if they were logged in unless they navigated to the profile page, which was not very user-friendly and inconvenient.

Before:

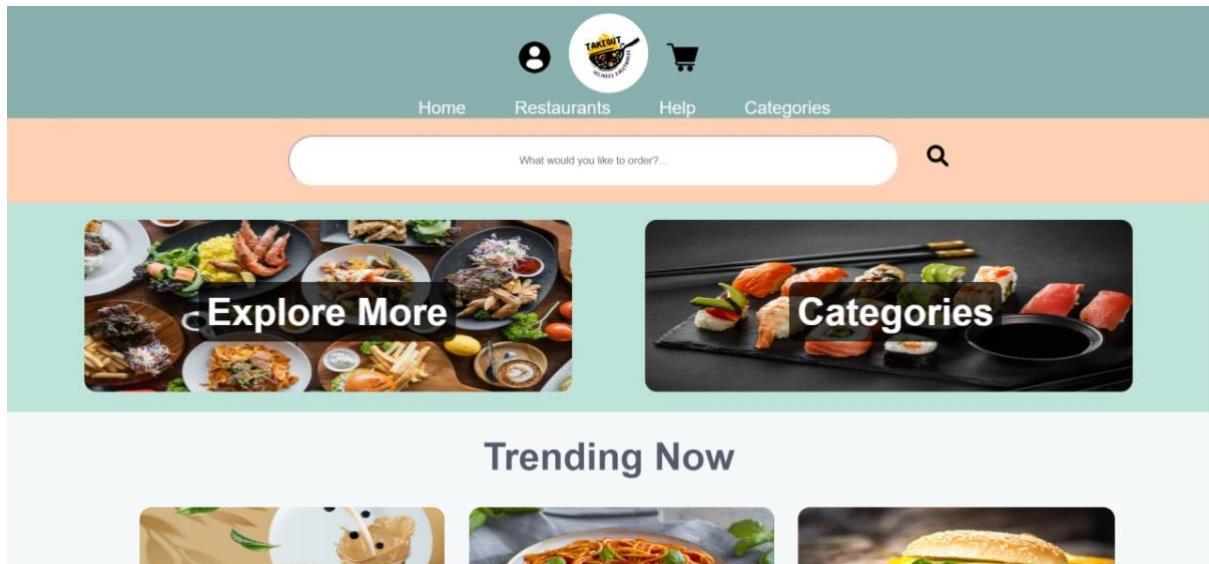


Figure 5.9 No design element to let users know of their log in status

After improvements:

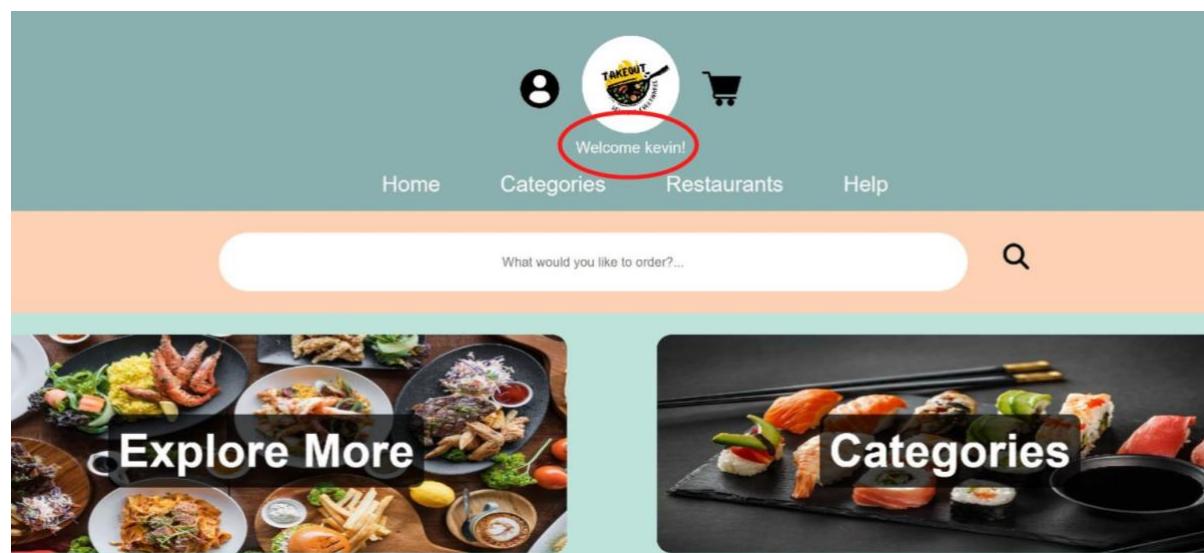


Figure 6.0 New design element showing the logged in status for an account with the username “kevin”

*Some additional libraries utilised*

Here is the list of the libraries we utilised in this entire design process.

**3<sup>rd</sup> party applications installed:**

**Npm packages installed:**

1. init
2. express
3. cors
4. body-parser
5. MySQL2
6. nodemon

**Npm packages (from React) installed:**

1. React-router-app
2. Axios
3. React-toastify
4. Body-parser
5. FontAwesome
6. Formik
7. Yup

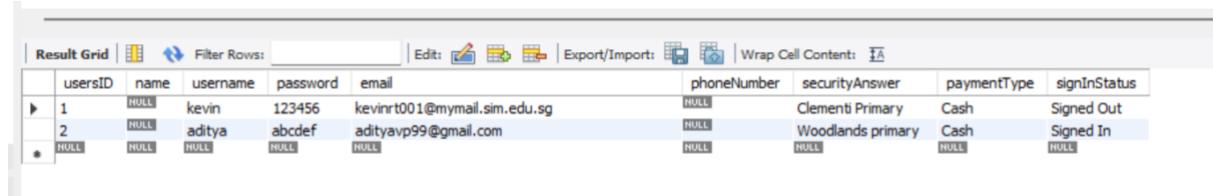
# System Development

In this segment, we will present the code that implements the multiple functionalities within our software. It will include explanations on how data is stored in the specific databases and how we process data queries. We will also mention the version control process, GitHub, which our team adopted to code together concurrently. During this system development process, alternating team members would also take turns to develop and do unit testing on the software at the same time. Our unit testing results are also documented within this segment of the report.

## Code Development

### Sign-in page

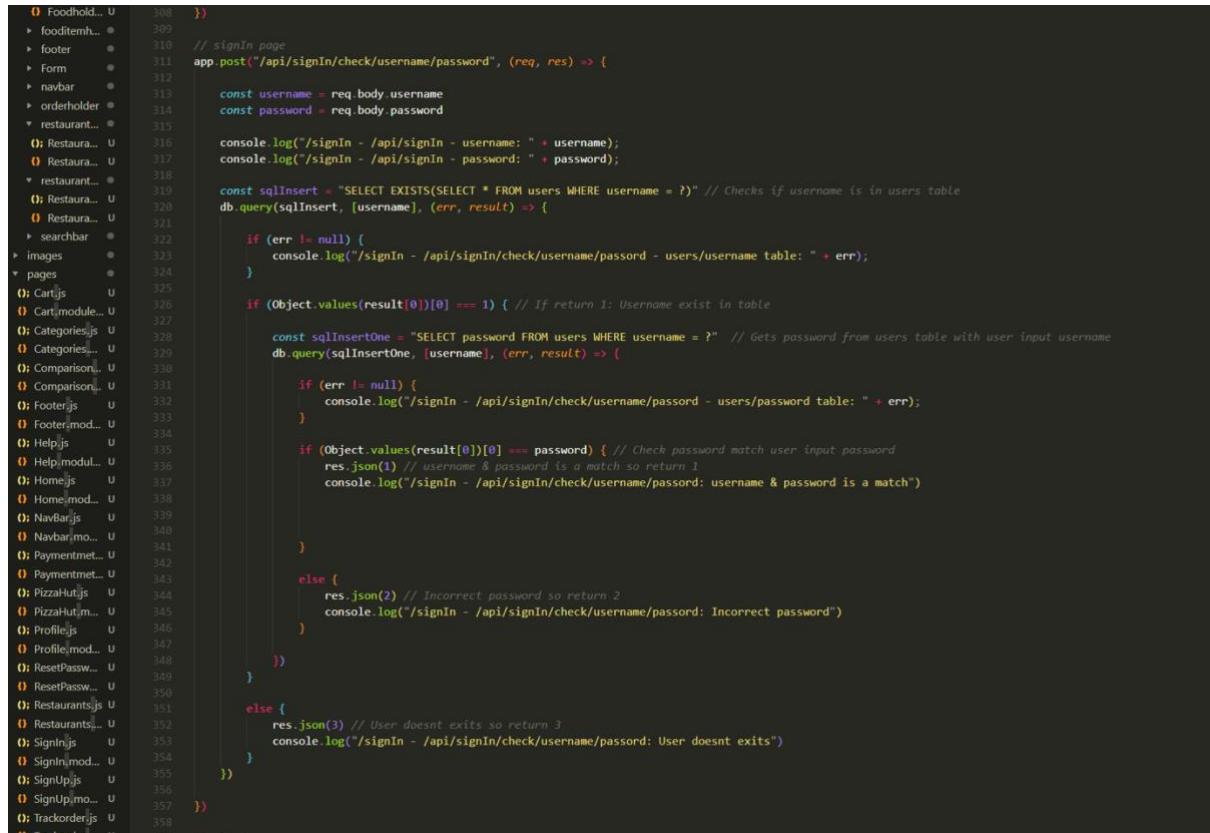
When users input their username and password, post request method will query into users table in the database and check if the username already exists with the correct password. If it exists and the password is correct, users will be logged in however if it does not exist they will be prompted to sign-up instead.



The screenshot shows a MySQL Workbench interface with a result grid titled 'Result Grid'. The grid displays data from a 'users' table with the following columns: usersID, name, username, password, email, phoneNumber, securityAnswer, paymentType, and signInStatus. There are three rows of data:

usersID	name	username	password	email	phoneNumber	securityAnswer	paymentType	signInStatus
1	kevin	kevin	123456	kevinrt001@mymail.sim.edu.sg	NULL	Clementi Primary	Cash	Signed Out
2	aditya	aditya	abcdef	adityavp99@gmail.com	NULL	Woodlands primary	Cash	Signed In
3	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 6.1 users table in MySQL database

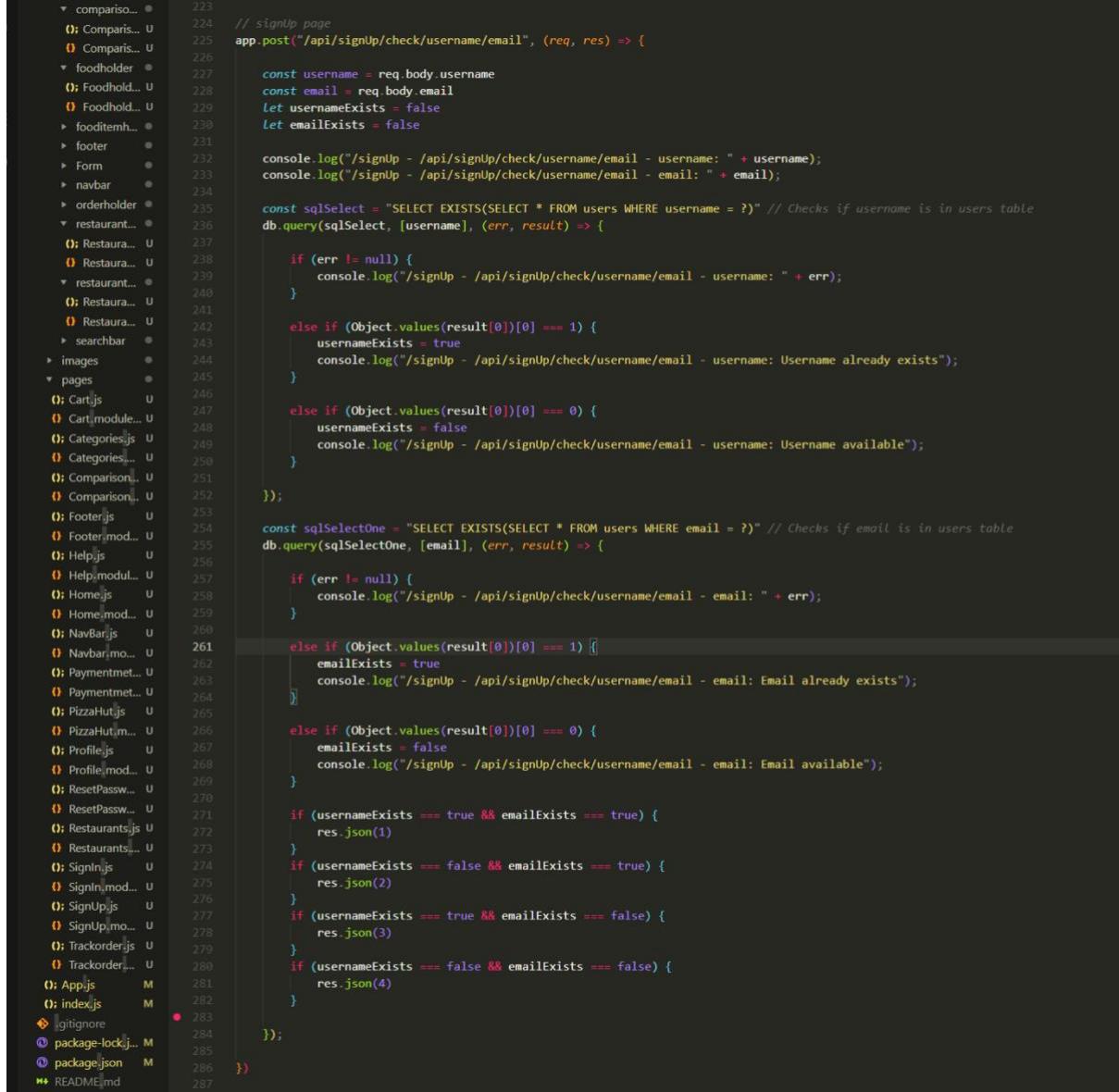


```
 308     })
 309
 310     // signin page
 311     app.post("/api/signIn/check/username/password", (req, res) => {
 312
 313         const username = req.body.username
 314         const password = req.body.password
 315
 316         console.log("/signIn - /api/signIn - username: " + username);
 317         console.log("/signIn - /api/signIn - password: " + password);
 318
 319         const sqlInsert = "SELECT EXISTS(SELECT * FROM users WHERE username = ?)" // Checks if username is in users table
 320         db.query(sqlInsert, [username], (err, result) => {
 321
 322             if (err != null) {
 323                 console.log("/signIn - /api/signIn/check/username/password - users/username table: " + err);
 324             }
 325
 326             if (Object.values(result[0])[0] === 1) { // If return 1: Username exist in table
 327
 328                 const sqlInsertOne = "SELECT password FROM users WHERE username = ?" // Gets password from users table with user input username
 329                 db.query(sqlInsertOne, [username], (err, result) => {
 330
 331                     if (err != null) {
 332                         console.log("/signIn - /api/signIn/check/username/password - users/password table: " + err);
 333                     }
 334
 335                     if (Object.values(result[0])[0] === password) { // Check password match user input password
 336                         res.json(1) // username & password is a match so return 1
 337                         console.log("/signIn - /api/signIn/check/username/password: username & password is a match")
 338
 339                     }
 340
 341                 }
 342
 343             } else {
 344                 res.json(2) // Incorrect password so return 2
 345                 console.log("/signIn - /api/signIn/check/username/password: Incorrect password")
 346             }
 347
 348         })
 349
 350     }
 351
 352     else {
 353         res.json(3) // User doesnt exits so return 3
 354         console.log("/signIn - /api/signIn/check/username/password: User doesnt exits")
 355     }
 356
 357 })
 358 }
```

Figure 6.2 Post request method for sign-in functionality

## Sign-up page

When users input their preferred username and email, post request method will query into users table in the database and check if the username or email already exists. If it already exists, Users will have to use a different username or email to sign-up. If it does not exist, the data will be stored in users table (Figure 6.1) as a new account (Figure 6.4).



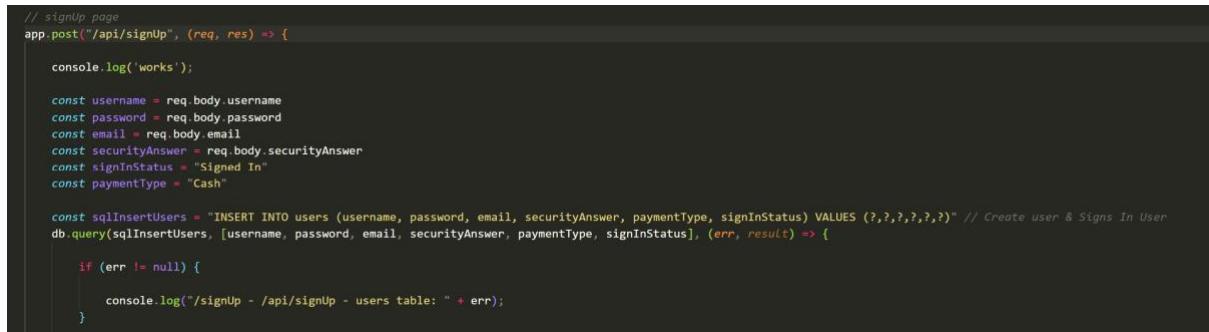
```

// sign up page
app.post("/api/signUp", (req, res) => {
  console.log('works');

  const username = req.body.username;
  const password = req.body.password;
  const email = req.body.email;
  const securityAnswer = req.body.securityAnswer;
  const signInStatus = "Signed In";
  const paymentType = "Cash";

  const sqlInsertUsers = "INSERT INTO users (username, password, email, securityAnswer, paymentType, signInStatus) VALUES (?, ?, ?, ?, ?, ?)" // Create user & Signs In User
  db.query(sqlInsertUsers, [username, password, email, securityAnswer, paymentType, signInStatus], (err, result) => {
    if (err != null) {
      console.log("/signUp - /api/signUp - users table: " + err);
    }
  });
}
  
```

Figure 6.3 Post request method for sign-up functionality



```

// sign up page
app.post("/api/signUp", (req, res) => {
  console.log('works');

  const username = req.body.username;
  const password = req.body.password;
  const email = req.body.email;
  const securityAnswer = req.body.securityAnswer;
  const signInStatus = "Signed In";
  const paymentType = "Cash";

  const sqlInsertUsers = "INSERT INTO users (username, password, email, securityAnswer, paymentType, signInStatus) VALUES (?, ?, ?, ?, ?, ?)" // Create user & Signs In User
  db.query(sqlInsertUsers, [username, password, email, securityAnswer, paymentType, signInStatus], (err, result) => {
    if (err != null) {
      console.log("/signUp - /api/signUp - users table: " + err);
    }
  });
}
  
```

Figure 6.4 Function that stores new account data into users table

## Add to cart page

When users sign-up, a cart table is created (Figure 6.5) and tagged to their account. Once users add their selections into the cart, the data will be updated to their cart table and it will be displayed in their comparison page and cart summary page. The function below is a PUT request method that helps to update the data to the cart table each time the user makes a change in their selection (Figure 6.6).

cartID	restaurantName	restaurantImage	foodItem1	foodItemPrice1	foodItemImage1	foodItemQuantity1	foodItem2	foodItemPrice2	foodItemImage2	foodItemQuantity2
1	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL
2	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

Figure 6.5 cart table in MySQL database

```
+ orderholder # 747
+ restaurante... # 788
  ① Restaura... U 789 // add to cart
  ① Restaura... U 790 app.put("/api/foodholder/add/to/cart", (req, res) => {
  ① Restaura... U 791   restaurante... # 792
  ① Restaura... U 793   const carts = req.body.carts
  ① Restaura... U 794   const restaurantName = req.body.restaurantName
  ① Restaura... U 795   const restaurantImage = req.body.restaurantImage
  + searchbar # 796   const fooditem = req.body.fooditem
  images # 797   const fooditemPrice = req.body.fooditemPrice
  pages # 798   const fooditemImage = req.body.fooditemImage
  Cart.js # 799   const quantity = req.body.quantity
  CartModule..U 800
  Categories.js # 801   const foodItemNumber = req.body.foodItemNumber
  Categories..U 802   const foodItemPriceNumber = req.body.foodItemPriceNumber
  Comparison..U 803   const foodItemImageNumber = req.body.foodItemImageNumber
  Comparison..U 804   const foodItemQuantityNumber = req.body.foodItemQuantityNumber
  Comparison..U 805
  Footer.js # 806
  FooterModul..U 807   const sqlUpdate = `UPDATE ${carts} SET restaurantName = ?, restaurantImage = ?, ${foodItemNumber} = ?, ${foodItemPriceNumber} = ?, ${foodItemImageNumber} = ?, ${foodItemQuantityNumber} = ? WHERE cartID = ?`;
  Help.js # 808   db.query(sqlUpdate, [restaurantName, restaurantImage, fooditem, fooditemPrice, fooditemImage, quantity, usersID], (err, result) => {
  HelpModul..U 809   // const sqlUpdate = `UPDATE cart1 SET restaurantName = ?, fooditem1 = ?, fooditem1price1 = ?, fooditem1image1 = ?, fooditem1quantity1 = ? WHERE cartID = ?`;
  Home.js # 810   // db.query(sqlUpdate, [restaurantName, fooditem, fooditemPrice, fooditemImage, quantity, usersID], (err, result) => {
  HomeMod..U 811
  NavBar.js # 812   if (err != null) {
  NavBarMod..U 813     console.log(`Add to cart - /api/foodholder/add/to/cart: ${err}`);
  PaymentMod..U 814
  PaymentMod..U 815   else if (usersID === null) {
  PizzaHut.js # 816     console.log(`Add to cart - /api/foodholder/add/to/cart: Not Signed In`);
  PizzaHut..U 817   }
  Profil.js # 818   }
  ProfilMod..U 819
  ProfilMod..U 820
  ResetPassw..U 821
  ResetPassw..U 822   else {
  Restaurants.js # 823     console.log(`Add to cart - /api/foodholder/add/to/cart: Added to cart`);
  Restaurants..U 824
  Signin.js # 825
  SigninMod..U 826   }
  Signin.js # 827   }
  SigninMod..U 828
  SignInMod..U 829   })
  SignInMod..U 830
  SignInMod..U 831
  TrackOrderer..U 832
```

Figure 6.6 PUT request method to update individual cart table

## Version Control Process

### Using Version Control:

During our project, our team decided to document our code on the version control system GitHub and linked our work directly to Visual Code. Hosting our project on the internet in the cloud meant everyone had easy access to it and could always see the most updated version. GitHub is also very suitable for undergoing Agile processes which is the case for this project. During the initial sprint, the front-end team helped set up the Git repository and created code templates for the client and server sides.

 adityavp99	Delete node_modules from home outer directory	dfa9bc3 23 minutes ago	⌚ 115 commits
 client	Updated pizzahut link, dropdown and totalcost	22 hours ago	
 server	updated packages to work with formik and yup	12 days ago	

Figure 6.7 Our Git node modules

As shown in the screenshot above, the client-side team had updated some changes in the front end, whereas the back end had updated packages. Using GitHub made it easy for both teams to seamlessly work hand-in-hand with independent testing.

The screenshot below shows the different branches created whilst embarking on the project. We had created individual branches for each page/component/feature, such as pizza hut, search bar, navbar, media-queries etc.

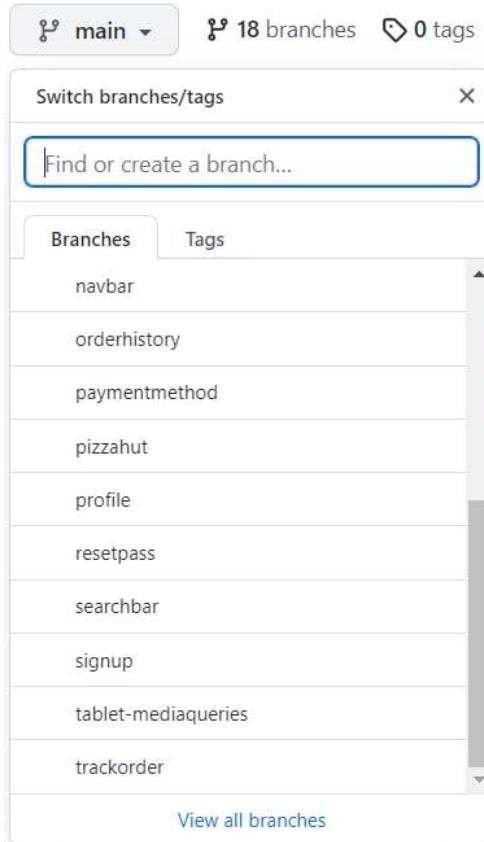


Figure 6.8 Git branches in our repository

We used branches to build individual pages/ features and simplified unit testing as we can test within our dedicated branches. Using git branches, we did not need to worry about messing up the main file. After our sprint reviews, we merged each feature to the home branch.

The screenshots below shows multiple commits made during sprints to the profile page branch (E.g. adding reward points, adding links, fixing CSS issues).

removed duplicate images from profile merge adityavp99 committed 2 days ago	<a href="#">f827e4c</a>
Final CSS adjustments nachiapp004 committed 2 days ago	<a href="#">04369f6</a>
Fixed footer, navbar links and profile page css nachiapp004 committed 2 days ago	<a href="#">d8035ad</a>
merged profile page with home page adityavp99 committed 2 days ago	<a href="#">f9212f7</a>
added reward points and promo codes adityavp99 committed 2 days ago	<a href="#">b3d65e9</a>
fixed error message styling adityavp99 committed 2 days ago	<a href="#">d95a674</a>
added links between pages adityavp99 committed 2 days ago	<a href="#">e29e66c</a>
merged cart page with home page adityavp99 committed 2 days ago	<a href="#">48e2cbf</a>
Fixed address and promo segment css nachiapp004 committed 2 days ago	<a href="#">5ec49fa</a>
Fixed styling in trackorder and footer adityavp99 committed 2 days ago	<a href="#">0869b43</a>
merged trackorder page with home page adityavp99 committed 2 days ago	<a href="#">3385366</a>

Figure 6.9 Git commits

We built the pages in stages and created commits for each significant step to allow us to revert back to the previous commit in the case of any discrepancies.

The screenshots below show commits made to merge branches to the home branch. (E.g. Comparison page, Sign In page, track order page etc)

Merge branch 'mobile_mq' into home nachiapp004 committed 23 hours ago	<a href="#">830cf50</a>
merged tablet media queries with home adityavp99 committed 3 days ago	<a href="#">2689c9c</a>

Figure 7.0 Git commits

trackorder done		f34910b			
mushhall committed 5 days ago					
merged signin page with home page		ac8eeb6			
adityavp99 committed 5 days ago					
Finished orderhistory page with styling		278a725			
nachiapp004 committed 5 days ago					
added delete icon and security question field		7a18c5d			
adityavp99 committed 5 days ago					
fixed comparison and pizzahut page issues		f28a210			
nachiapp004 committed 5 days ago					
Merge branch 'comparison' into home		96de225			
nachiapp004 committed 5 days ago					
Finished Comparison Page with styling		1020074			
nachiapp004 committed 5 days ago					
trackorder done		f34910b			
mushhall committed 5 days ago					
merged signin page with home page		ac8eeb6			
adityavp99 committed 5 days ago					
Finished orderhistory page with styling		278a725			
nachiapp004 committed 5 days ago					
added delete icon and security question field		7a18c5d			
adityavp99 committed 5 days ago					
fixed comparison and pizzahut page issues		f28a210			
nachiapp004 committed 5 days ago					
Merge branch 'comparison' into home		96de225			
nachiapp004 committed 5 days ago					
Finished Comparison Page with styling		1020074			
nachiapp004 committed 5 days ago					
merged profile page with home page		f9212f7			
adityavp99 committed 5 days ago					

Figure 7.1 Git commits

After our sprint reviews, we merged branches to the home branch and resolved merge conflicts if any occurred. This gave our team an idea of the completed features and progress of the whole in the home branch.

## Unit Testing

Unit testing aims to validate that each unit or component of the software code performs as expected (Hamilton, 2022). The developers should organise this testing during the development phase of the software. A unit may be an individual function, method, procedure, module, or object (Hamilton, 2022). Hence, our team decided to conduct a version of unit testing in sprint 4 of our project as we firmly believe this will set up our software on a solid foundation. With this, future developments will be less likely to encounter errors or failures, which is much more productive in the long run.

Case ID	Test functionality	Intended Outcome	Remarks	Test Result
<b>Navbar</b>				
1.1	Profile Icon	Supposed to link to signin page (when logged out) Supposed to link to profile page (when logged in)	Completed	Pass
1.2	Logo Picture	Supposed to link to homepage	Completed	Pass
1.3	Cart Icon	Supposed to link to cart (only when logged in)	Completed	Pass
1.4	List items	Supposed to link to the respective pages	Completed	Pass
<b>Footer</b>				
2.1	Logo Picture	Supposed to link to homepage	Completed	Pass
2.2	List items	Supposed to link to the respective pages	Completed	Pass
<b>Searchbar</b>				
3.1	Search field	Supposed to search and display items searched for user	Unable to complete functionality	Fail
<b>Homepage</b>				
4.1	CSS Responsive for Desktop/Laptop (1440px)	Page is supposed to be compatible for a device having a screen-size of 1440px	Completed	Pass
4.2	CSS Responsive for Tablet (768px)	Page is supposed to be compatible for a device having a screen-size of 768px	Completed	Pass
4.3	CSS Responsive for Mobile (425px)	Page is supposed to be compatible for a device having a screen-size of 425px	Completed	Pass
<b>Categories page</b>				
5.1	CSS Responsive for Desktop/Laptop (1440px)	Page is supposed to be compatible for a device having a screen-size of 1440px	Completed	Pass
5.2	CSS Responsive for Tablet (768px)	Page is supposed to be compatible for a device having a screen-size of 768px	Completed	Pass
5.3	CSS Responsive for Mobile (425px)	Page is supposed to be compatible for a device having a screen-size of 425px	Completed	Pass

Figure 7.2 Unit testing outcome for home page & categories page, for the full document, visit this [link](#).

<b>Help page</b>				
6.1	CSS Responsive for Desktop/Laptop (1440px)	Page is supposed to be compatible for a device having a screen-size of 1440px	Completed	Pass
6.2	CSS Responsive for Tablet (768px)	Page is supposed to be compatible for a device having a screen-size of 768px	Completed	Pass
6.3	CSS Responsive for Mobile (425px)	Page is supposed to be compatible for a device having a screen-size of 425px	Completed	Pass
6.4	FAQ Section	Each question is supposed to have a working drop-down to display answers	Completed	Pass
6.5	Submit button	Submit button is supposed to return an alert stating feedback received	Completed	Pass
<b>Restaurants page</b>				
7.1	CSS Responsive for Desktop/Laptop (1440px)	Page is supposed to be compatible for a device having a screen-size of 1440px	Completed	Pass
7.2	CSS Responsive for Tablet (768px)	Page is supposed to be compatible for a device having a screen-size of 768px	Completed	Pass
7.3	CSS Responsive for Mobile (425px)	Page is supposed to be compatible for a device having a screen-size of 425px	Completed	Pass
7.4	Individual Restaurant Links	Supposed to link to the respective restaurant pages	Completed	Pass

Figure 7.3 Unit testing outcome for help page & restaurants page, for the full document, visit this [link](#).

Pizza Hut Page(Individual Restaurant)				
8.1	CSS Responsive for Desktop/Laptop (1440px)	Page is supposed to be compatible for a device having a screen-size of 1440px	Completed	Pass
8.2	CSS Responsive for Tablet (768px)	Page is supposed to be compatible for a device having a screen-size of 768px	Completed	Pass
8.3	CSS Responsive for Mobile (425px)	Page is supposed to be compatible for a device having a screen-size of 425px	Completed	Pass
8.4	Add to cart button	Supposed to register quantity of food item to the database	Incomplete	Incomplete
8.5	Quantity buttons(+/-)	Supposed to increase or decrease the quantity of the food item	Completed	Pass
8.6	Quantity of food item (label)	Should only display values between 0 and 10	Completed	Pass
8.7	Continue button	Supposed to link to signin page (when logged out) Supposed to link to comparison page (when logged in)	Incomplete	Incomplete
Order Summary Page				
9.1	CSS Responsive for Desktop/Laptop (1440px)	Page is supposed to be compatible for a device having a screen-size of 1440px	Completed	Pass
9.2	CSS Responsive for Tablet (768px)	Page is supposed to be compatible for a device having a screen-size of 768px	Completed	Pass
9.3	CSS Responsive for Mobile (425px)	Page is supposed to be compatible for a device having a screen-size of 425px	Completed	Pass
9.4	Order Item	Supposed to display food quantity, item name, delivery partner and delivery timing	Incomplete	Incomplete
9.5	Save button	Should store the address in the database	Incomplete	Incomplete
9.6	Apply button	Apply promo code discount to the order	Incomplete	Incomplete
9.7	Redeem button	Redeem points to offset cost	Incomplete	Incomplete
9.8	Go back button	Redirects user back to comparison page	Completed	Pass
9.9	Proceed to pay button	Redirects user to paymentmethod page	Completed	Pass

Figure 7.4 Unit testing outcome for individual restaurant page & order summary page (cart page), for the full document, visit this [link](#).

Payment method page				
10.1	CSS Responsive for Desktop/Laptop (1440px)	Page is supposed to be compatible for a device having a screen-size of 1440px	Completed	Pass
10.2	CSS Responsive for Tablet (768px)	Page is supposed to be compatible for a device having a screen-size of 768px	Completed	Pass
10.3	CSS Responsive for Mobile (425px)	Page is supposed to be compatible for a device having a screen-size of 425px	Completed	Pass
10.4	Add card button	Supposed to store card details in the database	Incomplete	Incomplete
10.5	E-wallet drop-down	Supposed to display saved cards	Incomplete	Incomplete
10.6	Top Up button	Supposed to top up amount in the user's database	Incomplete	Incomplete
10.7	Pay button	Should redirect user to the trackorder page after payment	Completed	Pass
Track order page				
11.1	CSS Responsive for Desktop/Laptop (1440px)	Page is supposed to be compatible for a device having a screen-size of 1440px	Completed	Pass
11.2	CSS Responsive for Tablet (768px)	Page is supposed to be compatible for a device having a screen-size of 768px	Completed	Pass
11.3	CSS Responsive for Mobile (425px)	Page is supposed to be compatible for a device having a screen-size of 425px	Completed	Pass
11.4	Order History label	Should redirect user to order history page	Completed	Pass
Order History page				
12.1	CSS Responsive for Desktop/Laptop (1440px)	Page is supposed to be compatible for a device having a screen-size of 1440px	Completed	Pass
12.2	CSS Responsive for Tablet (768px)	Page is supposed to be compatible for a device having a screen-size of 768px	Completed	Pass
12.3	CSS Responsive for Mobile (425px)	Page is supposed to be compatible for a device having a screen-size of 425px	Completed	Pass
12.4	Order Item	Supposed to display food quantity, item name, delivery partner, delivery timing, points used, address, promo code used and order number	Incomplete	Incomplete

Figure 7.5 Unit testing outcome for payment method page, track order page & order history page, for the full document, visit this [link](#).

Sign In page				
13.1	CSS Responsive for Desktop/Laptop (1440px)	Page is supposed to be compatible for a device having a screen-size of 1440px	Completed	Pass
13.2	CSS Responsive for Tablet (768px)	Page is supposed to be compatible for a device having a screen-size of 768px	Completed	Pass
13.3	CSS Responsive for Mobile (425px)	Page is supposed to be compatible for a device having a screen-size of 425px	Completed	Pass
13.4	Username field	Should validate input against the database	Completed	Pass
13.5	Password field	Should validate input against the database	Completed	Pass
13.6	Sign In button	Should sign the user in if credentials are correct and redirect them to the home page	Completed	Pass
13.7	Forgot password label	Redirects user to the forgot password page	Completed	Pass
13.8	Sign up label	Redirects user to the sign up page	Completed	Pass
Sign Up page				
14.1	CSS Responsive for Desktop/Laptop (1440px)	Page is supposed to be compatible for a device having a screen-size of 1440px	Completed	Pass
14.2	CSS Responsive for Tablet (768px)	Page is supposed to be compatible for a device having a screen-size of 768px	Completed	Pass
14.3	CSS Responsive for Mobile (425px)	Page is supposed to be compatible for a device having a screen-size of 425px	Completed	Pass
14.4	Username field	Should validate input against the database	Completed	Pass
14.5	Password field	Should validate input against the database	Completed	Pass
14.6	Confirm password field	Should validate input against the password input	Completed	Pass
14.7	Email field	Should validate email format	Completed	Pass
14.8	Security Answer	Should store the answer in the database	Completed	Pass
14.9	Sign Up button	Should sign the user up and store the credentials in the database and redirect them to the home page	Completed	Pass
14.10	Sign In label	Redirects user to the sign in page	Completed	Pass

Figure 7.6 Unit testing outcome for sign in & sign up page, for the full document, visit this [link](#).

Reset Password Page				
15.1	CSS Responsive for Desktop/Laptop (1440px)	Page is supposed to be compatible for a device having a screen-size of 1440px	Completed	Pass
15.2	CSS Responsive for Tablet (768px)	Page is supposed to be compatible for a device having a screen-size of 768px	Completed	Pass
15.3	CSS Responsive for Mobile (425px)	Page is supposed to be compatible for a device having a screen-size of 425px	Completed	Pass
15.4	Email field	Should validate email format and against the database	Completed	Pass
15.5	Security Answer	Should validate the answer stored in the database	Completed	Pass
15.6	New password field	Should update the password in the database	Completed	Pass
15.7	Confirm password field	Should validate input against the password input	Completed	Pass
15.8	Sign In label	Redirects user to the sign in page	Completed	Pass
15.9	Register now label	Redirects user to the sign up page	Completed	Pass
Profile page				
16.1	CSS Responsive for Desktop/Laptop (1440px)	Page is supposed to be compatible for a device having a screen-size of 1440px	Completed	Pass
16.2	CSS Responsive for Tablet (768px)	Page is supposed to be compatible for a device having a screen-size of 768px	Completed	Pass
16.3	CSS Responsive for Mobile (425px)	Page is supposed to be compatible for a device having a screen-size of 425px	Completed	Pass
16.4	Name field	Should update name into database	Completed	Pass
16.5	Username field	Should update the username in the database	Completed	Pass
16.6	Password field	Should update the password in the database	Completed	Pass
16.7	Email field	Should validate email format	Completed	Pass
16.8	Security Answer	Should update the security answer in the database	Completed	Pass
16.9	Phone Number	Should update the phone number in the database	Completed	Pass
16.10	Update Profile button	Should update all the above fields in the database	Completed	Pass
16.11	Log Out button	Should sign the user out	Completed	Pass
16.12	Save button	Should store the address in the database	Completed	Pass
16.13	Card Number field	Should validate number input	Completed	Pass
16.14	Expiry Date field	Should validate Month/YYYY input	Completed	Pass
16.15	CVV	Should validate number input for maximum 3 digits	Completed	Pass
16.16	Add card button	Supposed to store card details in the database	Incomplete	Incomplete
16.17	E-wallet drop-down	Supposed to display saved cards	Incomplete	Incomplete
16.18	Top Up button	Supposed to top up amount in the user's database	Incomplete	Incomplete

Figure 7.7 Unit testing outcome for reset password page & profile page, for the full document, visit this [link](#).

## Challenges in implementation

### Some issues encountered:

- 1) One of the major challenges we faced was struggling to link the multiple tables in our database to each unique user. The multiple tables comprise of a users table (consisting user's profile data), cart table (consisting user's data selection), credit card table (user's credit card data) etc.
- 2) We encountered difficulty in implementing a function that can add up the total cost of all the food items (including respective quantities) that users selected. The implementation of this function was essential to create the respective data for our cart summary page.

### Solutions:

To resolve the first issue, we used the primary key which is the user ID in the users table as a foreign key in all the other tables to link them together to one specific user. To resolve the second problem encountered, we queried into the cart table to retrieve all the necessary data (food items, price & quantity) and extracted non-empty food item fields to compile the total cost amount. These were some of the methods which we used to resolve said challenges that we faced.

## Sprint Retrospective

This is a sprint retrospective table which we hope can help us plan more ways to increase the quality and effectiveness of any future developments that we will have. It is widely considered a crucial gathering that improves the potential of the Scrum Team and gives room for improvement in future Sprints (PremierAgile, n.d.).

Criteria	Sprint 1	Sprint 2	Sprint 3	Sprint 4
What can be done to improve quality and effectiveness?	Organize more in-person meetings than online to better communicate ideas.  Ensure that everyone in the team is clear about the objectives in each sprint.	Structure our manpower in a more efficient manner.  Be more realistic with what can be accomplished in a sprint to avoid over planning resulting in the need to compensate and causing the team unnecessary stress.	Plan for an additional day to fix complications or accommodate any unforeseen delays in task completion.	Plan for an additional day to fix complications or accommodate any unforeseen delays in task completion.
What was done well?	We had many forms of timeline/progress checklist (e.g. tables, Gantt charts etc.) which was extremely helpful to refer to before commencing the actual development.	The entire team was very productive in trying to complete tasks within the sprint to avoid delays.	Each team member was very initiative with assisting other members to avoid delays as we are approaching the deadline.	Our team came together to review the software and report which was essential as we could pin point any mistakes or missing elements and correct them on the spot.
What was not done well?	We did not ensure that everyone in the team was clear of the sprint objectives hence there were some miscommunications.	We were a little over confident and over planned the tasks in this sprint and resulted in some of our team members being extremely overwhelmed.	-	-
What are we committing to improve for our future developments?	We will co-ordinate our individual schedules and organize in-person meetings in advance. This will help with turn up rate as well as the planning process.	We will try to plan our sprints more realistically in future developments.	-	We wish to plan less in the last sprint to create space to accommodate any last minute changes or fixes.

Figure 7.8 Sprint retrospective table for the current development phase

## Evidence of Module Coupling

We would like to briefly touch on module coupling (Figure 7.9), a concept that exists in our system development. Module coupling measures the degree of interdependence between two things and how closely these two things are connected or represent the strength of the relationship between them (GeeksforGeeks, 2020). Here, we would like to discuss some examples of module coupling present in our development: data coupling and control coupling.

One of the more critical user data that is stored in our MySQL2 server contains our user's account details. This data will be the input for allowing users access to pages that require them to be logged in, for example, our price comparison. This is very representative of module coupling as the data input stored in our database (back-end) serves as an input that allows our users to access certain web pages/functionality, which is the output.

Another type of common module coupling, known as control coupling, can be observed in the interaction between our reset password page and the user's account. The data input here is when users change their account's password, which controls the original user account database by overwriting the original data. The output will allow users to replace their previous password with the new input they entered.

Understanding the concept of module coupling was extremely helpful for our team since it allowed both our back-end and front-end teams to better understand how the system works together. Thus, we can better grasp the concept of building a fully responsive software together.

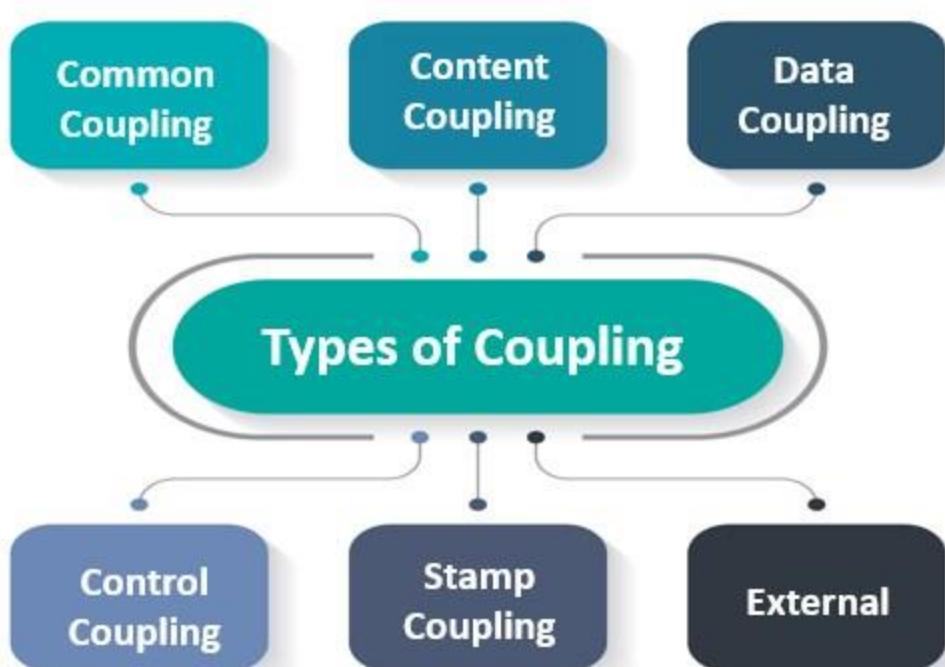


Figure 7.9 Different types of module coupling

## Analysis

### Project Analysis

To analyse this project as a whole, our team will be using a wide range of resources to determine the successes and failures in our Agile methodology and product.

#### *Scrum Methodology: Success Metrics Analysis*

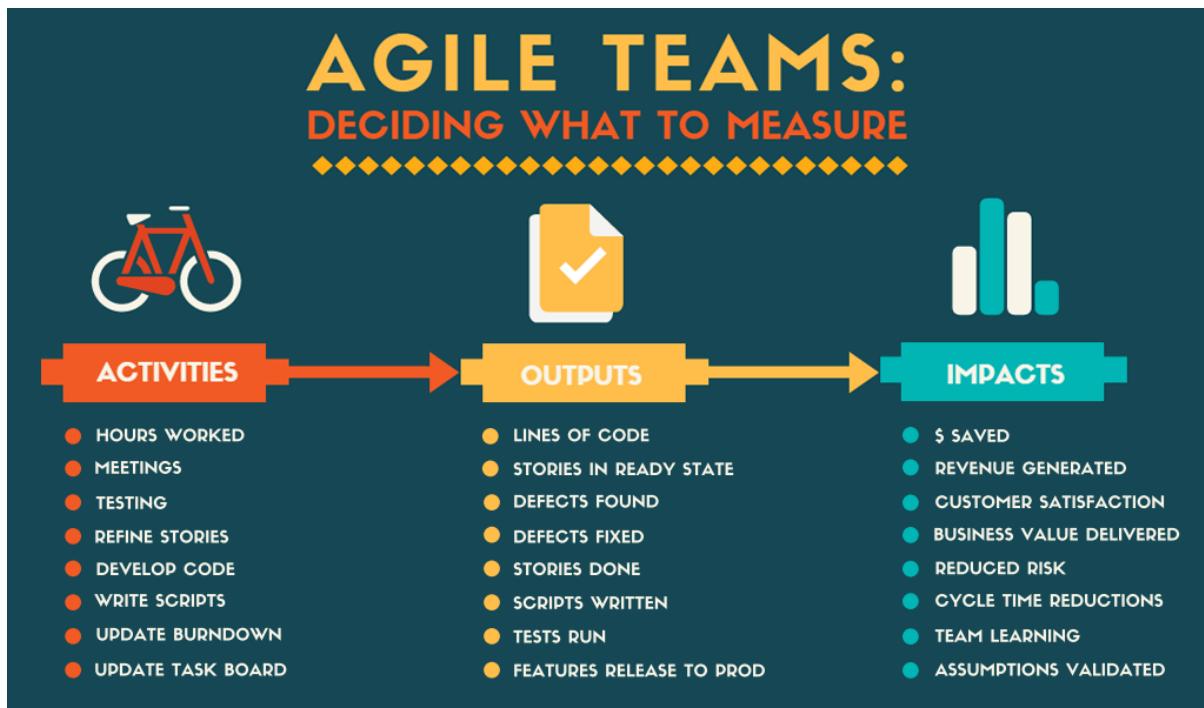


Figure 8.0 Image of the many different success measures in Scrum methodology

Here we will be using Scrum success measures to analyse our project. To measure the success of Scrum, it will be determined based on specific tracking metrics, whichever is relevant. "Success with Scrum is measurable by whatever metrics you would normally use. Those might be velocity, however you define it; improved efficiency, such as fewer cycles in the production process; and improved quality, such as fewer defects — whatever key indicators you hope to see improvement in." (Scrum Alliance, n.d.)

Based on resources (SeaLights, n.d.), our team has narrowed down to 3 main metrics which we believe to be the most significant factors in determining the success of our project. The **3 Scrum success metrics** are: **Sprint Goal Success**, **Customer Satisfaction** and **Daily Scrum & Sprint Retrospective**.

#### Measuring Deliverables – Sprint Goal Success: Failure

Based on the tables in Figure 1.8 and Figure 1.9, it shows our Gantt Chart and Sprint tasks log respectively. The Gantt Chart is indicative of the initial projected timeline and our actual timeline, this gives us a very detailed comparison of how much we strayed from what was initially planned. Based on the Sprint tasks log, our team has completed an estimated 97% of the tasks that were planned. However, we experienced multiple delays in some of the tasks as we had over planned the timeline and overestimated the time and effort we required for some of the tasks. As seen in Figure 2.0, there were a handful of tasks which we had to carry over to the next succeeding sprint to accommodate the delay. Our team felt that even

though we were able to complete the project before the deadline, we would like to acknowledge the struggles we faced that constituted our ineffective planning.

#### Measuring Effectiveness – Customer Satisfaction: Failure

To measure the effectiveness of our project, we refer to the Guerrilla User Testing we have done in sprint 2 and sprint 3. We feel that there is some lack in the quality of our user testing as our plan to implement it in sprints 2 and 3 was very premature. This resulted in us having insufficient data since the testing occurred in the early stages of our development. Due to this, our team lacks sufficient customer satisfaction information on our final deliverable. In the future development, our team will be taking more considerations into planning for user testing and unit testing in sprints. We hope that our testing procedures can be planned more wisely such that there exists an iterative process that depicts a trend in the level of user satisfaction from the start to the end of the project.

#### Measuring the Scrum Team – Daily Scrum & Sprint Retrospective: Success

Despite facing struggles in the entire development process, our team has managed to document our Group Meeting Logs with Sprint Retrospectives (Figure 1.4). This is indicative that our team has worked with high measures of teamwork and collaboration. Even though there were inconsistencies in our deliverables, each member has given their best in reviewing each other's progress and tasks. We are happy with the consistency in our Scrum team, and we hope to maintain the same levels in our future developments.

We have failed two out of three of our chosen Scrum success metrics. Even though we managed to deliver a complete product before the deadline, since we are planning for future developments, it is more in our best interest to acknowledge failures than successes. This allows us the opportunity to learn from our mistakes and strive to produce deliverables of higher quality in future. There are many metrics scales in Scrum success analysis. However, we felt that the three mentioned above were very applicable to the nature of our project and its development. Using these three metrics to analyse our project allows us to see what we did well and lacked. In this case, we can look further into improving our customer satisfaction and sprint goal success. In our evaluation segment below, we will use a Scrum methodology to evaluate our project. It will consist of some solutions we hope to adopt in future developments to enhance our Scrum work ethos and deliverables.

## Evaluation

### Project Evaluation

To conduct a critical evaluation of our project, we have decided to use the Scrum methodology.

#### Scrum methodology evaluation:

Prospect	Benefits	Challenges	Solutions	Improvements
<b>Daily Scrum meetings (both online &amp; in-person)</b>	This was essential for the team as it sets the foundation for us to work together while avoiding any miscommunications.	Timings for each meeting are hard to organise as we have to consider the individual schedules of all five members.	We have adopted giving our individual availability in advance such that we can avoid last minute meetings and low turn up rate.	Each member should take accountability to inform the group in advance if they are unable to make it for any meetings.
<b>Sprint Planning meetings</b>	This gave our team a very clear guideline to follow, there were little to no confusions about the task to complete and the given timeline for doing so.	-	-	-
<b>Sprint Duration</b>	This was essential in ensuring that we complete each task in the given concise duration and do not create any major delays in the timeline.	There were a couple of more complex tasks that exceeded the planned duration.	To accommodate the minor delays we moved the tasks into the succeeding sprint.	Team members experiencing difficulty in completing the tasks within the given duration should voice it out during our meetings so that other members may assist if available.

<b>Sprint Review</b>	This was useful in allowing each member to demonstrate their completed tasks and allows our team to document and track each progress made.	-	-	-
<b>Sprint Retrospectives</b>	This was a vital aspect in our project as it allowed us to feedback on each other's work. The feedback was important in helping us ensure that our project is constantly evaluated at every checkpoint.	Not all members were proactive in providing constructive feedback.	All members should be encouraged to have an opinion on the project be it positive or negative.	In future, we implement a guideline for each member to provide at least one opinion on the work. It can be a positive reinforcement or a constructive feedback.

Figure 8.1 Scrum methodology evaluation on our project

## Product evaluation

To critically evaluate our final product, our team will use the same SWOT analysis, which will be essential in developing marketing strategies and the PESTLE business analysis to gain valuable information about the macro-environment around our project/product.

### SWOT analysis on TakeOut:

<b>Strengths</b>	<ul style="list-style-type: none"> <li>Successfully implemented <b>user authentication in our forms</b> to verify legitimacy of data</li> <li>Web application designed based on <b>user-centred design elements</b></li> <li>Unique <b>comparison feature</b> that allows our users to compare price points</li> <li>Update profile functionality for users to <b>customise their account details</b></li> <li><b>Secured functionality to retrieve account</b> if users forget their password via validating their email and answering their security question</li> <li><b>E-wallet feature</b> which users can choose to top-up to such that they do not need to save their card details in the application if they prefer not to</li> </ul>
<b>Weaknesses</b>	<ul style="list-style-type: none"> <li>TakeOut's track order feature was unable to be implemented as it requires a database which we <b>do not have access to</b> at this point</li> <li>Some functionalities are too complicated to implement in this given limited timeline so there may be a <b>lack in complex depth</b> in certain aspects of the application (e.g. internal search engine)</li> <li><b>Lack in variety</b> of food options for our users (this is due to our launch being a demo version so we still do not possess the APIs from our partnering food delivery brand at this stage but we will be rectifying this post-launch)</li> </ul>
<b>Opportunities</b>	<ul style="list-style-type: none"> <li>Our web application was built on a structure of multiple combination of components hence it is <b>non-restrictive</b> and we can improve or implement more features in any future developments</li> <li>Our product reserves large <b>potential for advertisement partnership</b> if we can attract a large base of users</li> <li>We have plans to incorporate food options that are <b>inclusive of all cultures and dietary restrictions</b> as we do not believe in exclusivity</li> </ul>
<b>Threats</b>	<ul style="list-style-type: none"> <li>Users that are <b>loyal to their existing choice of food delivery platforms</b> might not be convinced to try out TakeOut</li> </ul>

Figure 8.2 SWOT analysis of TakeOut

# PESTLE ANALYSIS



Figure 8.3 PESTLE analysis

## PESTLE analysis:

Political	<p>✓: TakeOut is launched in an era of change, during this COVID-stricken times, the local Singapore government has launched <b>additional fundings</b> to every individual. These fundings are to sustain the new lifestyle of working at home hence cooking at home or ordering food deliveries for consumption especially during nationwide lockdown situations are extremely common. Our government also strongly advises against any close contacts hence <b>food delivery service</b> that minimises interactions are <b>highly encouraged</b>.</p> <p>✗: For people living in countries with <b>low political stability</b>, they are unlikely to receive any additional fundings let alone the individual ability to maintain basic necessities hence the <b>luxury of being able to afford food delivery services</b> are <b>very unlikely</b>.</p>
Economic	<p>✓: For countries with a <b>strong economy</b>, the chances of individuals having the ability to afford such food delivery services are therefore much higher.</p> <p>✗: Due to the recent COVID-19 pandemic, times are trying and economies are highly impacted. Many people have experienced a <b>loss of income</b> hence will be deterred from paying for the luxury of food delivery.</p>
Social	<p>✓: In our current society technologies have become extremely advanced, people are constantly striving to find ways to increase convenience and efficiency in their daily livelihoods. Food delivery services are therefore extremely common as it helps to save time especially for individuals who have very packed schedules.</p>

Technological	<p><b>✓:</b> Statistically, 63.1% of the world's population possesses <b>internet connectivity</b> (DataReportal, 2022) and anyone with such <b>access</b> can readily become users of our software.</p> <p><b>✓:</b> With the support of highly evolving technologies, we will possess <b>unlimited resources</b> to implement any of our <b>future developments</b> (i.e. new features, artificial intelligence integration).</p>
Legal	<p><b>✗:</b> As our software directly stores user's sensitive personal data (e.g name, contact number, address, emails &amp; card details) we will be held <b>highly responsible for adhering to privacy laws and regulation</b>. Any offences can result in <b>legal liabilities</b> and <b>cost our users basic rights to safety</b>.</p> <p><b>✗:</b> There are <b>legal concerns</b> to consider when we look for food delivery companies to partner with for access to their APIs.</p>
Environment	<p><b>✓:</b> Our software runs <b>digitally</b> hence there is <b>no direct impact</b> on the environment.</p> <p><b>✗:</b> There are some <b>areas of environmental considerations</b> that impacts our jurisdiction for example, we should eliminate working with brands (i.e. restaurants) and companies whom businesses are harming our environment or takes part in greenwashing. Partnering with such brands/companies will <b>negatively impact the reliability and accountability</b> of our brand.</p>

Figure 8.4 PESTLE analysis for TakeOut

## Conclusion

### Project Conclusion

To conclude our project, we would like to summarise our outcomes and achievements at this stage and explain briefly on how it will impact our community of stakeholders.

#### **Outcomes summary:**

Technically-wise, the outcome we intend to achieve is to garner users for our web application. Gaining traction in our software will provide us with tangible user data, which is essential for some of our un-launched features. For example, our order tracking functionality which we understand as a necessary feature for a food delivery platform such as ours. As it is not applicable to use any dummy data or non-real-time data resources for this specific function, it becomes a conundrum when we currently do not have access to such real-time data. We hope to be able to begin implementation of the said feature once we gain valuable data upon launching our product. Ultimately, the goal our team wishes to achieve is to impact any individuals with our product positively. Food consumption is a basic necessity, and any method to increase convenience and access to this necessity is our priority.

We look forward to seeing the positive changes we can incur in our users' daily living and consumption. In terms of our future work, we have plans to incorporate machine learning elements into our project. This will involve including recommender systems for a more personalised search result.

#### **Impact on our community of stakeholders:**

As mentioned in our outcomes, we would like to elevate the convenience and access to food consumption for individuals that utilise our product. In a larger perspective, we hope to enhance the food industry by driving friendly competition between companies offering food delivery services. Our platform focuses on keeping these companies accountable for the affordability of their services. These brands will also be pushed to innovate and elevate their businesses for more growth by satisfying more needs of the users. With such competition, we hope our application can bring back healthy growth in the food industry and the economy, as the COVID-19 pandemic has largely impacted it. Especially in times like these, economic development to gain back what we lost will benefit everyone.

We managed to launch 95% of the features we intended to produce in this demo version of our application. We ensured that our project was implemented based on a dynamic workflow mindset as this will allow us to change or add on to the project in future when needed. This will be our first demo launch of TakeOut, so we hope to gain the outcomes we have predicted and move towards the next part of our future developments.

### **Personal Reflection (ADITYA VIJINDRA PANDHARI)**

As a Scrum Master, I helped to setup meetings and discussions in a timely and convenient manner for all members to ensure that regular and effective communication was being carried out between all team members. I also moderated discussions to ensure that differences in opinions were raised in a healthy and constructive manner so that all views were brought to light but did not hurt or discourage other members in the process.

As a back-end developer, I had to familiarise myself with MySQL queries, RESTful API calls such as 'get', 'post' and 'put' to retrieve, insert, and update data into the database by using a library called Axios in the client side and Express in the server side. Through this process, I was constantly in touch with my other database team member to split work fairly, raise any impediments, discuss solutions and other useful tools and information that I learnt along the way to speed up the learning process and decrease the probability of mistakes being committed.

**Word count (excluding Appendix): 9971**

## Appendix

### Appendix A

*Table of Contributions*

	Kevin	Kannappan	Aditya	Charlotte	Nicole
<b>Background</b>					
Abstract	✓				✓
Outcomes		✓		✓	
Introduction	✓		✓		✓
<b>Planning and Research</b>					
Time and Resource allocation			✓		✓
Team Breakdown	✓	✓		✓	
Frameworks considered		✓	✓		✓
Project Management Tools		✓			
Research – Summative (Initial report)	✓			✓	
Research – Current	✓				✓
<b>Prototyping and Iteration</b>					
Low Fidelity Wireframe (Concept vs Proof-to-concept evaluation)	✓			✓	✓
Medium Fidelity Wireframe (Concept vs Proof-to-concept evaluation)	✓			✓	✓
High Fidelity Mock Up (Concept vs Proof-to-concept evaluation)	✓			✓	✓
Features to implement	✓			✓	✓
<b>Design</b>					
Use Case Model				✓	✓
User Testing	✓	✓	✓	✓	✓
Design Changes	✓			✓	✓
<b>System Development</b>					
Code Development		✓	✓		✓
Version Control Process	✓	✓	✓	✓	
Unit Testing	✓	✓	✓	✓	✓
Challenges in implementation		✓	✓		
Sprint Retrospective		✓	✓		✓
Evidence of Module Coupling		✓	✓		✓
<b>Analysis</b>					
Project Analysis	✓	✓	✓	✓	✓
<b>Evaluation</b>					
Project Evaluation	✓	✓	✓	✓	✓
<b>Conclusion</b>					
Project Conclusion	✓	✓	✓	✓	✓
Personal Reflection	✓	✓	✓	✓	✓
<b>Appendix</b>					
Appendix A	✓				✓
Appendix B	✓	✓	✓	✓	✓

Figure 8.5 Report contribution table

	<b>Kevin</b>	<b>Kannappan</b>	<b>Aditya</b>	<b>Charlotte</b>	<b>Nicole</b>
<b>Back-end</b>	✓		✓		
<b>Front-end</b>		✓		✓	✓
<b>Full stack</b>		✓			

Figure 8.6 Developer table

## Appendix B

### References

1. The Daily Egg, 2022, 'ClickUp Review', <<https://www.crazyegg.com/blog/clickup-review/>>.
2. Takeout, 2022, 'Agile Software Project Mid-term Proposal'.
3. Utkarsh Sidana, 22<sup>nd</sup> November 2021, 'What are the Advantages and Disadvantages of Angular', <<https://www.edureka.co/blog/advantages-and-disadvantages-of-angular/>>.
4. Swati Saini, Sakshi Arora, 24<sup>th</sup> November 2021, '8 Proven Reasons You Need Angular For Your Next Development Project', <<https://www.grazitti.com/blog/8-proven-reasons-you-need-angular-for-your-next-development-project/>>.
5. Michele Stieven, 21<sup>st</sup> November 2017, "React is better than Angular" and other nonsense', <<https://michelestieven.medium.com/react-is-better-than-angular-and-other-nonsense-6ff752161cf#:~:text=Angular%20doesn't%20have%20Virtual.they're%20both%20fast%20enough!>>.
6. Cue Logic Insights, 9<sup>th</sup> September 2020, 'Angular vs. React: A Complete Comparison Guide', <<https://www.cuelogic.com/blog/what-are-the-differences-between-angular-and-react>>.
7. RubyGarage, 2022, 'How to choose between Angular and React for your next project', <<https://rubygarage.org/blog/how-to-choose-between-angular-and-react>>.
8. DDI Development, December 2020, 'Pros and Cons of Vue.js Framework Programming', <<https://ddi-dev.com/blog/programming/the-good-and-the-bad-of-vue-js-framework-programming>>.
9. StackOverflow, n.d., 'Web Frameworks', <<https://insights.stackoverflow.com/survey/2021#most-loved-dreaded-and-wanted-webframe-love-dread>>.
10. Web Crunch, 24<sup>th</sup> January 2016, 'Should I use SVG, Icon Fonts or Images.', <<https://web-crunch.com/posts/should-i-use-svg-icon-fonts-or-images>>.
11. Beste Burcu Bayhan, 25<sup>th</sup> January 2022, 'React Form Libraries Comparison: Formik vs React Hook Form', <<https://apiumhub.com/tech-blog-barcelona/react-form-libraries-comparison-formik-vs-react-hook-form>>.
12. David Zheng, 23<sup>rd</sup> May 2018, '8 Reasons Why Your Website Needs Search Engine Optimization.', <<https://www.crazyegg.com/blog/why-search-engine-optimization/#:~:text=It%20helps%20your%20target%20audience%20find%20your%20site&text=In%20fact%2C%2062%25%20of%20consumers,they're%20ready%20to%20buy>>.
13. Nick Babich, 1<sup>st</sup> December 2020, 'Practical Tips For Creating Smooth Website Navigation Experience', <<https://uxplanet.org/practical-tips-for-creating-smooth-website-navigation-experience-3034ba9961b9>>.
14. Cacoo, 23<sup>rd</sup> December 2021, 'How a UML use case diagram can benefit any process', <<https://cacoo.com/blog/how-a-uml-use-case-diagram-can-benefit-any-process/>>.

15. Elizabeth Chesters, 9<sup>th</sup> May 2017, 'What is guerrilla testing and how do you use it?', <<https://www.userzoom.com/blog/what-is-guerrilla-testing-and-how-do-you-use-it/>>.
16. Emily Grace Adiseshiah, n.d., 'Guerrilla Usability Testing: How To Introduce It In Your Next UX Project', <<https://usabilitygeek.com/guerrilla-usability-testing-how-to/>>.
17. Google Developers, 23<sup>rd</sup> June 2015, 'Guerrilla Testing with Usability Cafe', <<https://www.youtube.com/watch?v=0YL0xoSmyZI>>.
18. Thomas Hamilton, 27<sup>th</sup> August 2022, 'Unit Testing Tutorial – What is, Types & Test Example', <<https://www.guru99.com/unit-testing-guide.html>>.
19. PremierAgile, n.d., 'Everything You Need to Know About Sprint Retrospective', <<https://premieragile.com/everything-you-need-to-know-about-sprint-retrospective/>>.
20. GeeksforGeeks, 16<sup>th</sup> June 2020, 'Module Coupling and Its Types', <<https://www.geeksforgeeks.org/module-coupling-and-its-types/#:~:text=Module%20coupling%20means%20to%20couple,Coupling%20is%20related%20to%20cohesion.>>.
21. Scrum Alliance, n.d., 'How to Measure Scrum Success', <<https://resources.scrumalliance.org/Article/measure-scrum-success#:~:text=%22Success%20with%20Scrum%20is%20measurable,hope%20to%20see%20improvement%20in.%22>>.
22. SeaLights, n.d., '11 Scrum Metrics and Their Value to Scrum Teams', <<https://www.sealights.io/software-development-metrics/11-scrum-metrics-and-their-value-to-scrum-teams/>>.
23. DataReportal, July 2022, 'Digital Around The World', <<https://datareportal.com/global-digital-overview#:~:text=A%20total%20of%205.03%20billion,12%20months%20to%20July%202022.>>.