

YourFamily

'A web application for a family, by a family'

By MUHAMMAD AZAAN ZAFAR

EECS Third Year Project (BSc Computer Science) May 2022

Supervisor: Dr Paulo Oliva

School of Electronic Engineering and Computer Science

FINAL YEAR REPORT
THIRD YEAR PROJECT

DECLARATION

This report entitled

YourHome - A web based application to bring families together

Is written and composed by me and is based on my own work except where indicated in the text throughout the report.

Name: Muhammad Azaan Zafar

Signed: Azaan

Date: 04/05/2022

Abstract

The world has become digitalized and technology has become a prominent part of our lives. Children absorption of technology from texting to video games limits interaction and communication with their parents and other members of their household (Taylor, 2013). In a way, technology has taken its toll on many people. Families are physically communicating less, siblings are less aware of what's going on each others lives and this gap is increasing daily. Evidently, a study shows that a working parent returning home from work is only greeted 30% of the time where as 50% were totally ignored (Taylor, 2013). In conclusion, technology is creating a family divide.

This report will describe a solution to overcome the problem of family division and minimize the gap in a household between family members. The solution in this report will increase family communication and interaction through the help of a web based application providing a variety of tools, such as allocating tasks, reward system based on points, meal voting system and much more, that will increase involvement of members of the family, especially children.

Acknowledgments

First and foremost, I would like to thank Allah (SWT) for making this project possible and providing me with the knowledge and strength to complete this research through the times of ease and difficulty.

I would like to thank my supervisor Dr Paulo Oliva for his continuous support, advice, and guidance throughout this project. I have received a great amount of support from him as well as a great amount of beneficial knowledge which allowed me to complete this project.

Contents

List of Figures	vii
List of Tables	vii
1 Introduction	1
1.1 Background	1
1.2 Problem Statement	1
1.3 Aim	2
1.4 Objectives	2
1.5 Research Questions	3
2 Literature Review	4
2.1 Technology and its effect on communication and family division	4
2.2 Similar Systems	5
2.2.1 Apple Family Sharing	5
2.2.2 Family Wall	7
2.2.3 Cozi Family Organizer	8
2.2.4 Home Organizer	8
2.3 Comparison Of Similar Systems	9
2.3.1 Similar Features Across All Applications	9
2.3.2 Comparison Of Different Features In Similar Applications	11
2.4 Proposed Solution	13
3 Project Requirements and Project Plan	14
3.1 Functional Requirements	14
3.2 Non-Functional Requirements	15
3.3 Project Plan	16
4 Design and Model Methodology	18
4.1 High-Level Design Overview	19
4.2 Application Walkthrough	20
4.3 System Architecture, MVC Design Pattern	30

4.4	Model Methodology	31
5	Deployment	32
6	Implementation	33
6.1	Rationale for Platform and Technologies	33
6.1.1	The Django Framework	33
6.1.2	Bootstrap	33
6.1.3	AJAX	33
6.2	API's and Libraries used	34
6.2.1	GeoLocation Google Maps API	34
6.2.2	Places API	35
6.2.3	The Directions API	36
6.3	Core Functionalities and Models	37
6.3.1	Login Page	38
6.3.2	Add Task Page	39
6.3.3	Meal Planner Voting and Results Functionality	40
6.3.4	Chat Functionality	41
7	Testing	43
7.1	Unit testing by Django	46
7.1.1	URL Testing	46
7.1.2	Login Test Case	47
7.1.3	Adding Rewards	47
7.1.4	Summary of Unit Testing	48
7.1.5	Shared Location Google Maps API Selenium Test Case	48
7.1.6	Summary of Selenium Testing	51
7.2	Cross-platform Browser Compatibility Testing	51
7.3	Summary of Testing	51
8	Evaluation	53
8.1	Usability/Normal Evaluation Metrics	53
8.2	Results and Analysis	53
8.3	Summary of Evaluation	55

9	Conclusion	57
9.1	Achievements and Challenges	57
9.2	Future Improvements	59
9.3	Final Remarks	60
10	Appendix	61
10.1	Appendix A: Risk Analysis	61
	References	65

List of Figures

3.1	Gantt chart for semester 2 activities	16
4.1	Flow diagram of the system by a Parents point of view	19
4.2	Flow diagram of the system by a Child's point of view	20
4.3	State Diagram of system	21
4.4	Login Page	21
4.5	Register Page	22
4.6	Add Task Page/ View Your Task Page	23
4.7	Slider/options tab	24
4.8	Shared Grocery List - Personal todo list	24
4.9	Meal Planner Home Page	25
4.10	Parent view only: Creating/updating meals	26
4.11	Results page	26
4.12	Add/Claim Rewards page	27
4.13	Error message when claiming reward	27
4.14	Shared Location Web Page	28
4.15	Shared Gallery Main Page	29
4.16	View Details Page showing the picture in a bigger format as well as the map from current location to picture location	29
4.17	Shared Gallery Main Page	30
4.18	MVC Design Pattern (MVC Design Pattern - GeeksforGeeks, 2018)	31
6.1	code to get the users coordinates and send it to backend	34
6.2	code to fetch coordinates from backend and display on Google Maps with marker	35
6.3	Auto completing location using Place API	35
6.4	Code for Auto complete Place API	36
6.5	Showing the directions from Point A to point B	37
6.6	Code for the, The Directions API	37
6.7	Code for authenticating the user	38
6.8	code for the User model in models.py	39
6.9	Code for adding a task	39

6.10	Code for voting poll	40
6.11	Code for voting poll	42
6.12	Code for voting poll	42
7.1	Unit test for a given URL	46
7.2	Result of URL test	47
7.3	Unit test for a logging in	47
7.4	Result of Login test case	47
7.5	Unit test for adding rewards	48
7.6	Code for logging in	49
7.7	Code for map test	50
7.8	Output for map API test	50
8.1	Barchart displaying the features users use the most to reduce family division	54
8.2	Is this system useful to you and your family - does it reduce family division?	
	If so, will you recommend it to other families?	54
8.3	Can you use the API's efficiently without coming across any errors? . . .	55
8.4	Results for survey about location being accurate	56

List of Tables

1	Comparison of common features in similar applications	10
2	Comparison of the different features in similar systems	12
3	Functional Requirements	14
4	Non-Functional Requirements	15
5	Project Plan, Semester 2	16
6	Authentication testing	44
7	Unit Testing For All Functionalities	48
8	Selenium Testing	51
9	Risk Analysis	62

1 Introduction

1.1 Background

Technology is advancing faster than ever. An increase in telecommunication implementation throughout the world has substantially increased internet usage thus increasing internet penetration rate to 59% in January 2021 which is an increase of nearly 7% from 2020 (Bulao, 2021). Evidently, it is clear that the world is evolving around technology. Technology is an important factor of our daily lives. The most important factor that technology brings to our daily lives is communication from sharing images to your Whatsapp groupchat to having a Zoom conference meeting with your manager, everything is evolved around technology. Communication is a key part of our lives and technology and communication go hand-in-hand. The advancement of technology and an increase of other messaging platform such as, smartphones and video consoles, has sparked a decrease in physical communication and interaction.

Communication begins from your home. Talking, interacting, planning activities and much more are the different aspects of communication that takes place in a household. This is the easiest and most common use of communication. As the author of this report, personally speaking, there was a time in my home where members of the family took part of different activities together. This could be from eating together at the dinner table to watching a show together on every Sunday. However, these activities are very rare to be seen in a given home. This is due to the fact people have indulged themselves in technology and are unconsciously ignoring what goes around them. From texting applications to game applications, the marketers of these applications do not take into consideration the fact they are creating a family division.

1.2 Problem Statement

In the current day and age, interaction between family members has decreased. Each member is involved in their own activity in their own little bubble which mainly evolves around the use of smartphones. Family members interact with each other on a smartphone through a messaging application more than they do physically. These messaging

applications are good for communicating text across one device to another, however such systems do not implement any interactivity which would influence family bonding. Furthermore, during the pandemic many questions arose in relation to mental health and how it will affect a mass population. However, results from a study show that youths who were facing mental health problems prior to the pandemic, the symptoms were drastically reduced during the pandemic and the sole reason behind this was the increase in the level of quality time spent with their families (Penner, Hernandez and Sharp, 2020). Therefore, implementing YourFamily, a web based application to increase interactivity between family members, will help overcome family division and increase the level of quality time spent within a family.

1.3 Aim

This project aims to help allow and give the ability to families, specifically parents, to reduce or eliminate the communication gap that a family may have inside their home. Parents will be able to increase communication within their family by increasing participation through many features such as, assigning chores/tasks to children that they need to complete within a certain time frame, giving the freedom of choice to members of the family by allowing them choose and vote for their daily meals, keeping track of the location of each family member and much more. All these features, and many more, will conquer the disadvantages that similar systems may propose which are listed in the literature review below. Additionally, this project will overcome the Socio-digital division between the two generations, parent and child, in a family by creating an application that requires a limited amount of skills set to enable both generations, especially parents, to easily use the system.

1.4 Objectives

- To carry out in-depth research of similar existing systems, their features, advantages and limitations.
- To develop a system that gives parents the ability to reduce a family communication gap by implementing features that will increase interaction.

- To allow parents to assign tasks and to allow members to update the status of their given tasks to completed or pending.
- To develop a system that rewards members who complete a given task by allowing parents to assign points to a task where children can then redeem those points for rewards.
- To understand the concept of how an Application Programming Interface (API) works so that a system is developed to allow members in a family to locate each other.
- To develop a system where communication in a family becomes easier.
- To gather data through existing customers of existing similar applications so that an understanding of benefits and limitations become clear.

1.5 Research Questions

The following research questions are set to be answered in this project:

- How can you improve the organisation and communication between family members through a web application?
- How can you portray a web application, that keeps 'track' of every family member in all aspects, as an appealing rather than an intrusive application.
- How is a family division created and why?
- How can a family gap be reduced through a web application?
- What are families currently doing to reduce their family gap?
- What are the different things that can bring a family close together and so what features should be implemented for these different things?

2 Literature Review

This section aims to briefly discuss what communication is and the effect it has had because of technology and it will look into existing systems in the market. The data and information gathered will then be further used for gathering requirements for the proposed solution. Due to technology, the level of communication within in a family is reducing (Taylor, 2013). There is known to be a negative correlation between the amount of interpersonal communication in a family and the amount of smartphones being used by children. The level of interpersonal communication within a family decreases as the amount of smartphones being used by children increases. Data from another report suggests that 80-90% of all teenagers use their smartphones for purely texting (Haig, 2002). This is one of the main problems that is encouraging family division.

2.1 Technology and its effect on communication and family division

Communication is the process of interchanging information between individuals. Constant communication is the key to maintain a healthy relationship. Regular communication and interaction with eachother in a family promotes a healthy family relationship and prevents a family from growing apart. Many families are growing apart despite the advancements of communication technology. Members of a family, especially children, have wrapped themselves in their personal social lives, and forget to interact and put an effort in within their own family to build a better relationship (Hardy, 2021).

”Interpersonal communication is the process by which people exchange information, feelings and meaning through verbal and non-verbal messages; it is largely a face-to-face type of communication” (Ngunan and Regina, 2016). With an increase of communication technology being accessible to many people today, it is believed that interpersonal communication is on a continuous decline. ”Why is that?”- a very popular question asked in this day and age where communication has become very easy due to technology. A profound reason for this continuous decline in interpersonal communication is the socio-digital divide. Socio-digital divide is when the society you are in can cause and create dependency on technology which then can further create inequality in between

generations with different skills set in relation to technology. An estimated of 80-90% of children have smartphones and are dependant on them for communication in terms of texting rather than verbal communication (Haig, 2002). Members of the family, specifically children, come home from work or school and tend to indulge themselves in their own personal digital world. They do not tend to pay importance to what goes on in their family nor try to go sit down for an hour or two to relay how their day went to their parents, who are aging and urging to interact with their children. Additionally, the digital divide between the two generations may also be one of the reasons why a family grows apart. Parents may not be up-to-date with the latest technology that their second generation is using and so they are unable to 'fit' in with them. In essence, technology is the reason for an increase in family division.

To conclude, we can say that communication, which requires technology, has increased with the advancements of technology. However, despite technology being a reason of increase in 'online' communication, it is also the reason for family division. Evidently, as explained above, technology plays a huge role in everyone's lives, specifically teenagers and young adults. Parents are also exposed to technology, but they require a simpler system with a simpler interface to become confident with any given system. Even though technology is creating family division, if we use it in the correct way it can also reduce family division. For example, by developing an online web application, which increases interaction in a family and reduces family division, by targeting both groups; Parents Children. We will use 'technology' and 'smartphones', which children are already indulged in, to tackle the issue of family division.

2.2 Similar Systems

2.2.1 Apple Family Sharing

Apple Family sharing is a system, provided by Apple for users with Apple devices, that enables a family to easily share different Apple services with each other. For example, these include: iCloud, Apple Music, Apple TV, Apple calendar, shared location and many more Apple features. It's a relatively simple to use system that requires an adult

in the family to create a 'group' and then invite members to that group. Members of that group are now subscribed to the group's subscription and content that are eligible for sharing (What is Family Sharing?, 2021). A feature that is very useful is shared location. Members of the group in Family Sharing are able to see each other's location live. Additionally, you can share your gallery with any member, of your choice, in the group. The benefit of using Family Sharing in comparison to other applications such as, FamilyWall, is that it enables you to share the features on your Apple device with other members that you may use daily. These could be, Apple Music, Apple Books, Apple TV and much more. Whereas, FamilyWall is not device specific hence it does not have the ability to let users share their applications that they may have on their device.

However, Family Sharing does come with some limitations. The first and main limitation of Family Sharing, by Apple, is that it is device specific. This means that Family Sharing can only be accessed by users who have Apple Devices. This can be an issue for families who possess different smartphones. For example, an Android user will not be able to use Family Sharing as it's only for Apple devices. This limits the target market as not everyone will want to purchase Apple Devices just to solely use Family Sharing. Additionally, Family Sharing is not as simple to use. As Family Sharing can be accessed by all Apple Devices (Apples Ecosystem), therefore this means the process of setting up Family Sharing on all devices will differ. The following example shows how to turn on Shared Albums on the different Apple Devices:

- On your iPhone, iPad, or iPod touch, go to Settings >[your name] >iCloud >Photos, then turn on Shared Albums.
- On your Mac, open the Photos app. From the menu bar at the top of the screen, choose Photos >Preferences. Click the iCloud tab, then select Shared Albums.
- On your Apple TV, go to Settings >Users and Accounts >iCloud and turn on Shared Albums.

(How to share albums in Photos on your iPhone, iPad, and Mac, 2020)

As seen above, users who have different Apple devices may find difficulty when setting up Family Sharing as there are so many different steps to follow for different devices. There should be one system, with one interface as this reduces complexity, especially for elders,

in a system and markets a wider audience.

2.2.2 Family Wall

FamilyWall is an application, available on both the App store and Google Play Store, that serves the purpose of organizing a family through the use of a Shared Calendar, To-do an Grocery lists, Timetables, Shared Location, Gallery and much more (FamilyWall - Happy Family Organization, n.d.). FamilyWall is one of the most known applications out there in its category. FamilyWall is very similar to Apple's Family Sharing, however it is it's own system and a whole application in itself. FamilyWall supports as many members in a group and so targets anywhere from a small family to a large family. It has many different features that other applications do not consist off. For example, it consists of a Shopping List where different members of the group can add items they need to buy whereas Family Sharing, by Apple, does not consist of this. FamilyWall consists of some unique features which make the application stand out which are, Meal Planner, connecting Google Outlook External Calendar, additional storage for Gallery and a few more.

However, despite these being unique features which make the application standout, they are apart of the FamilyWall Premium bundle. To access these features, you need to purchase the FamilyWall Premium bundle. Additionally, many users have left a 3 star or below rating due to the fact they were looking forward to using the Meal Planner but it turned out to be an in-app purchase. One user wrote a review on Google Playstore stating the following: "After browsing through and trying some of the features I learned that I would need to pay for the meal planning option. I currently have another app that does not require me to pay for the recurring but does not have the meal planner. I guess I will continue looking for an app to meety needs that is free" (Adkison, 2021). Additionally, despite it having a GPS locator, many users have complained that it does not work well and is extremely slow and lags (Chapman, 2021). Furthermore, the interface is not simple to use which prevents the target market, especially they parents, to use the system.

2.2.3 Cozi Family Organizer

Cozi is mobile based application available on both the App Store and Google Play Store. Cozi is free and easy to download. It provides members of the family, who have a different day-to-day schedule, to bring their schedule to one place. Cozi provides families with a Family Calendar, Shopping To-do Lists and a Recipe Box. These are the only three features that Cozi provides its users. Cozi focuses on bringing everyone's schedules to one place rather than focusing on increasing interactivity by features that FamilyWall or Family Sharing implements. The benefit Cozi has over other applications like FamilyWall is that it provides a very simple interface, especially for the Calendar. This is very important as parents, who are not as advanced with technology compared to their second generation, will need to use the application and organize all different things for their children.

However, Cozi is very limited in the features it provides. Families want one application to suit all their needs. Members using Cozi can not message each other. This is an important feature that differentiates all other applications with Cozi. Families want to increase interaction within their home, and so they would like an application that provides the ability to communicate with each other. Additionally, Cozi does not offer the ability to have multiple members in your Cozi Family Group. In fact, to use Cozi, all users must sign in with one shared main account. This further reduces the chances of Cozi implementing ideas that are user specific. For example, users will not be able to share their location as there's only one main account. Another issue found with Cozi is that users are facing issues syncing all their content across different devices. Katherine Schowalter, a user of Cozi, has said in her review that she is finding difficulty in viewing her appointments across different devices (Schowalter, 2019). Furthermore, a 'chores/task' features has been implemented to allow parents to keep track of the tasks their children have completed. However, the downside to this is that, users are unhappy as there is no incentive or a reward system in place when a child completes a task.

2.2.4 Home Organizer

Home Organizer is an application which is only available on the Google Play Store. Home Organizer is a very simple-to-use app. It allows family members to:

- Create tasks and allocate points for it
- Claim rewards from the points they have gained
- Plan meals for the day
- Add groceries in a shopping list
- Message other members (premium feature)
- Have a shared calendar (premium feature)

Home Organizer provides many features that families are looking for. It also focuses on creating a positive interaction between members by introducing a reward system for children which also is an incentive for them. It allows users to have a shared meal planner for the day.

However, all these features come with limitations. Users are able to create tasks, however, there is no set deadline for a child to complete their tasks. This demotivates a child from doing their task and thus possible leading to family division. Additionally, there is no messaging system that allows all members to communicate with each other. There is no GPS locator which can allow parents to track their child's movements and make sure they are safe. Despite this system being simple to use, there is no interactivity with the interface. Children are attracted to applications that offer interfaces with interactivity.

2.3 Comparison Of Similar Systems

In this section, I will be summarizing the similar features in the applications mentioned above and will be creating multiple tables to show a comparison of similar systems. I will then use this to compare it to my system and demonstrate how it overcomes the limitations of existing systems.

2.3.1 Similar Features Across All Applications

In this section, the table below shows all the features that are common in all the applications I have spoken about above.

Table 1: Comparison of common features in similar applications

Similar Features Across All Applications				
	Family Sharing	FamilyWall	Cozi	Home Organizer
Shared To-do list	✓	✓	✓	✓
Shared Calendar	✓	✓	✓	✓
Notifications	✓	✓	✓	✓
in-app purchases	✓	✓	✓	✓
Requires downloading an app	✓	✓	✓	✓

2.3.2 Comparison Of Different Features In Similar Applications

In this section, I will be looking at the features that not all application share between themselves. For example, Table 2 below, shows us that FamilyWall, Family Sharing and Home Organizer all share the feature which allows multiple users to create an account and use their system, respectively. However, it can also be seen that Family Sharing and Cozi do not implement the chat system feature whereas other applications do.

Overall, from Table 2 we can see that there is not a single application on the market that implements all the features that are listed. This therefore means that there is a potential gap in the market for an application that implements all those features in one system. So therefore, I have decided to implement all the features listed on Table 1 and all the features on Table 2 that are not common across all applications. Thus, there will be one system with all the features included.

Table 2 is shown below on the next page.

Table 2: Comparison of the different features in similar systems

Different Features Across All Applications					
	Family Sharing	FamilyWall	Cozi	Home Organizer	YourFamily (my App)
1+ users in a group	✓	✓		✓	✓
Meal planner		✓ (in-app purchase)		✓	✓
Chat system		✓		✓ (in-app purchase)	✓
Reward system				✓	✓
Personal todo list					✓
Shared location	✓	✓ (in-app purchase)			✓
Shared Gallery	✓	✓			✓
Allocating Tasks					✓

2.4 Proposed Solution

From the research taken above in Section 2.2 and with the tables produced in Section 2.3, it is evident that similar systems fall short when it comes to having all useful and demanded features in one system. For example, Cozi, a family based application does not have a chat system implemented. The aim of this project is to produce a web based application that focuses on reducing family division and a gap in a family as much as possible. I believe, having a chat system is a very important feature for any application that aims to get a family interacting together. Additionally, I will be implementing more features to overcome the limitations of existing systems. The additional features, that are not on any similar application, I will be implementing are as follow:

- A Meal Planner but with a voting system to let users vote for their meals - this increases interaction and participation.
- An advanced Shared Gallery where it allows users to list the location of where the image was taken - this will then allow other users to navigate to the location, through Google Maps API.
- A Reward System - tasks will be allocated points, and for any task a user completes, they will be given 'x' amount of points. These points can then be used to purchase rewards.
- A Leaderboard - shows users who have collected the most points in descending order. This is a strategy to increase participation and interaction.
- A Personal To-do list - even though this application aims to bring families together by providing transparency, it is important that users have some sort of privacy. Having a Personal To-do list will allow users to keep track of their personal daily tasks.

I will be developing a system that implements all features, as shown in Table 1 and Table 2, as well as with the unique additional features listed above.

3 Project Requirements and Project Plan

This section will go through the different functional and non-functional project requirements. It will then show the different requirements that have been fulfilled as of today, 28/11/2021, and the requirements that are still in progress. The project plan table will provide a project plan that will clearly display the features that are directly related to the functional requirements that need to be completed by a certain time frame.

3.1 Functional Requirements

Table 3: Functional Requirements

	Requirement	Completed
1	Users should be able to register, choose a group, and login to the application.	✓
2	Once logged in, parents should be able to create/delete tasks for a specific child	✓
3	Children should then be able to update their tasks: completed/pending	✓
4	Children should receive 'x' amount of points for completing their tasks	✓
5	Parents should be able to set rewards that children can redeem for 'x' amount of points.	✓
6	Children should then be able to redeem those rewards and points should be deducted accordingly.	✓
7	The application should display a leaderboard of all users in the group with the amount of points they have earned, in descending order.	✓
8	Parents should be able to create a meal planner for Breakfast, Lunch and Dinner. Three options should be created for each part of the day and voting should be allowed.	✓
9	Children should be able to vote for their choice of meal for the specific part of the day.	✓

10	Voting should commence at the end of the day and those meals that have 'won' should be used for the following day.	✓
11	The application should allow all users to post pictures in their Shared Gallery	✓
12	Pictures in the Shared Gallery should have a location to which users can then use the Google Maps API to locate the way from their current location.	✓
13	The application should allow members in the same group to locate each other using the Geolocation and Google Maps API	✓
14	The application should allow members from the same group to message each other through the application	✓
15	The application should allow users to create events on a calendar for their weekly schedule.	

3.2 Non-Functional Requirements

Table 4: Non-Functional Requirements

	Requirement	Completed
1	The system should be available to all users on all different devices through the web	✓
2	The application should be responsive using bootstrap	✓
3	The application should provide error prevention	✓
4	The application notifies a user when they are messaged or a task is added for them	✓
5	The application should use AJAX through Fetch API for asynchronous requests	✓
6	The application should use VueJS for interactivity and responsiveness	
7	The application should make sure that a user from group 'x' can not access data from group 'y'	✓

8	The map API's should provide a quick result without any unnecessary delays	✓
---	--	---

3.3 Project Plan

As most of the functional requirements have been completed during Semester 1, the plan table will provide a schedule for Semester 2.

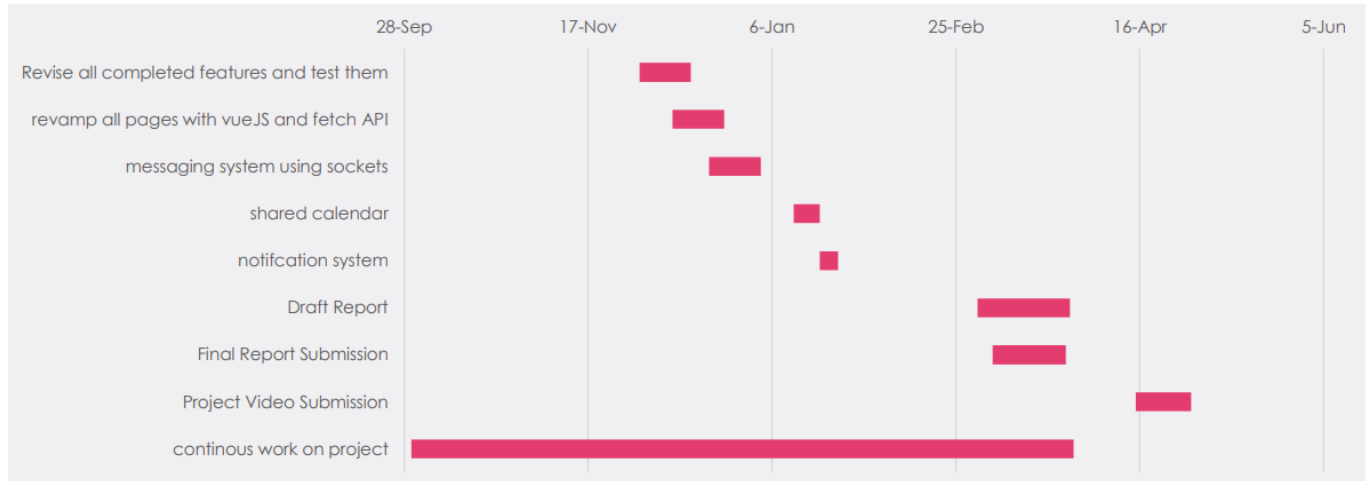


Figure 3.1: Gantt chart for semester 2 activities

Table 5: Project Plan, Semester 2

	Task	Time frame
1	Go through all completed functional requirements and test each page to see if it meets the purpose.	Dec: week 1 and 2
2	Implement AJAX through fetch API using VueJS on each completed page	Dec: week 3 and 4
3	Implement a messaging system using Django websockets	sem 2: week 1 and week 2
4	Implement a shared calendar and allow users to create events and be notified	sem 2: week 3 and week 4
5	Implement a notification system	sem 2: week 5 and week 6

6	Go through each component of the web application and make sure layout is correct and is responsive	sem2: week 7 and week 8
7	Make sure that each page considers who is logged in eg. Parent or Child	sem 2: week 9
8	Run unit tests and conclude report	sem 2: week 10 to week 11 and 12

4 Design and Model Methodology

This section will aim at explaining the user interface as well as the different web pages and how a user may go from one page to another. Additionally, there will be flow diagrams which will convey the different steps a user may take to complete a task depending on what page they are on.

During the final part of this section, there will be a walk through of the system's architecture and the design methodology used to build the system.

Note to reader: The next part of this section is on the next page as placing the diagram on this page will cause the content to be lost during the shrinking process.

4.1 High-Level Design Overview

This section display 2 different flow-diagrams. The first one shows the flow of the application if you were to be a 'Parent' in the system, whereas the second diagram shows the flow of the application if you were to be a 'Child' in the system. It is important to note that a parent can perform every task a child can perform, however, not vice-versa.

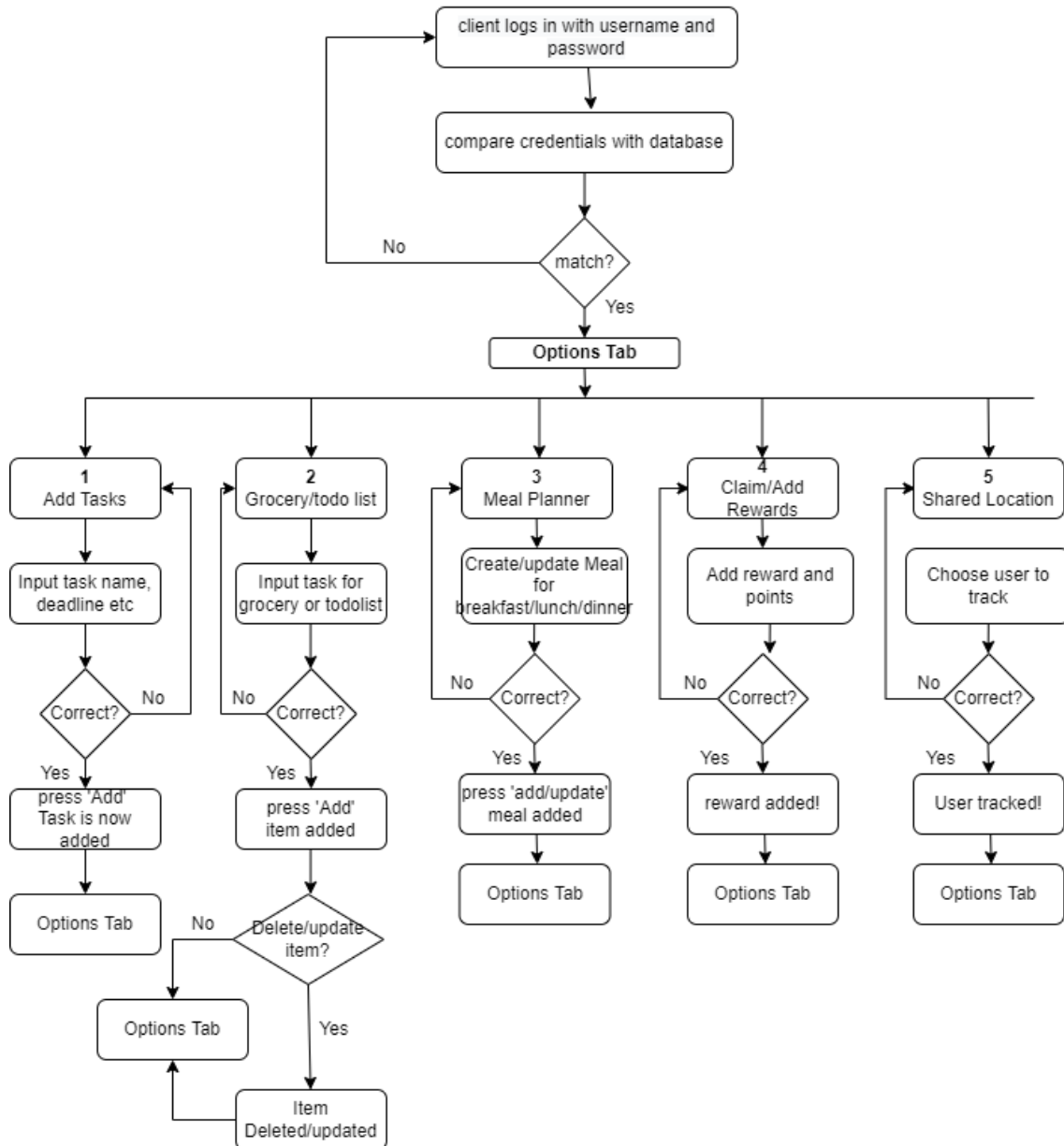


Figure 4.1: Flow diagram of the system by a Parents point of view

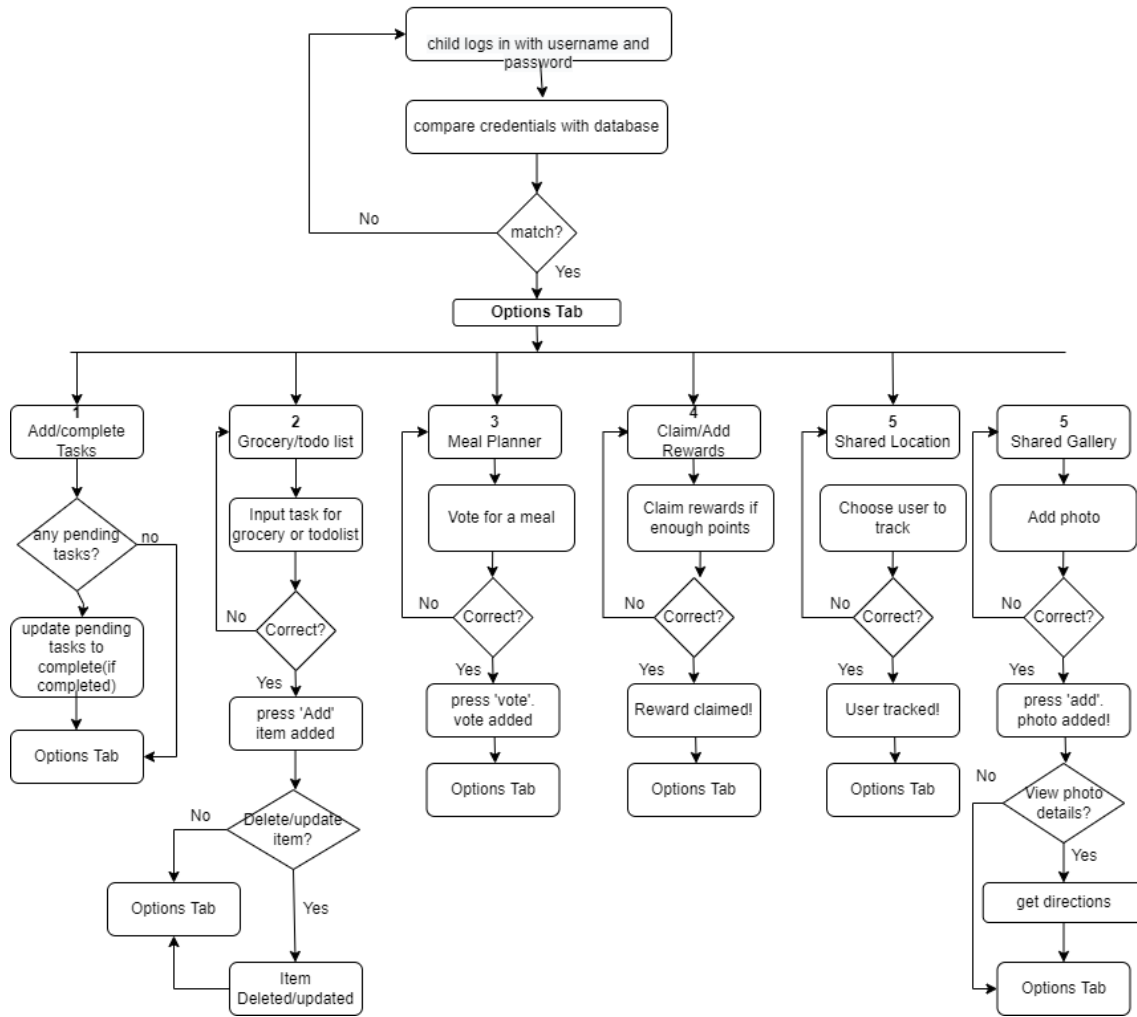


Figure 4.2: Flow diagram of the system by a Child's point of view

4.2 Application Walkthrough

This section aims to provide the functionality of each web page and how it works and transitions to another web page in the system. The section will go through the two different 'user levels'; Parent and Child and how each user level changes how a given web page reacts.

As there are many different states in this web application, it is only appropriate to convey them with a state transition diagram which will elegantly explain how a given state may transit to another.

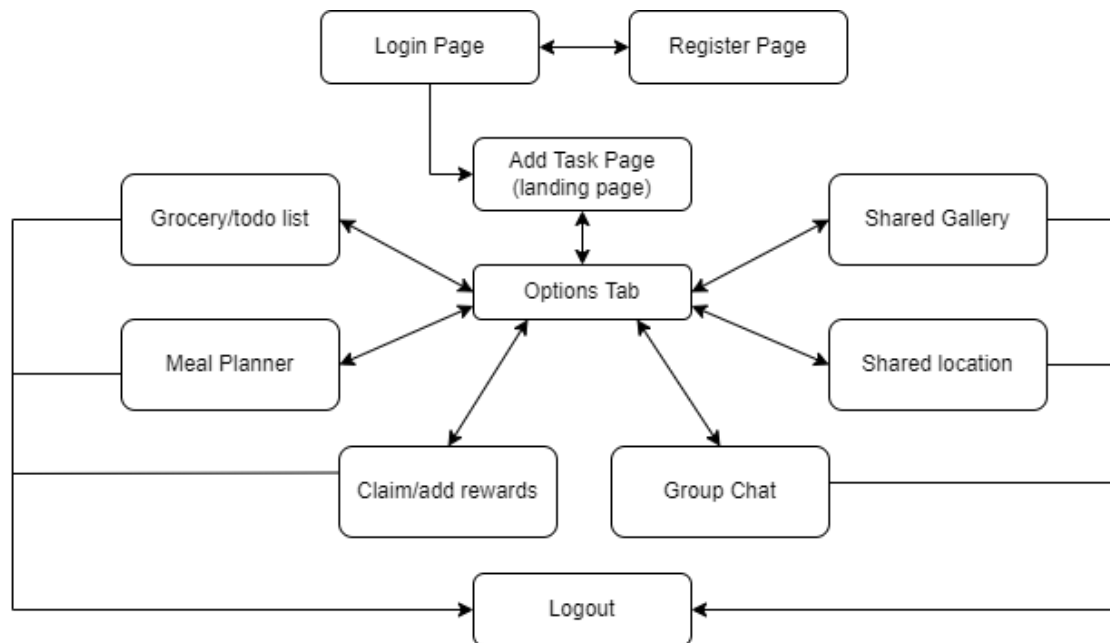


Figure 4.3: State Diagram of system

Login Page

The login page is the main landing page when a user visits the website. This page is simple and displays two simple text boxes which ask the user for their username and password.

If a user is new, they can simply make an account by clicking on the sign-up page.

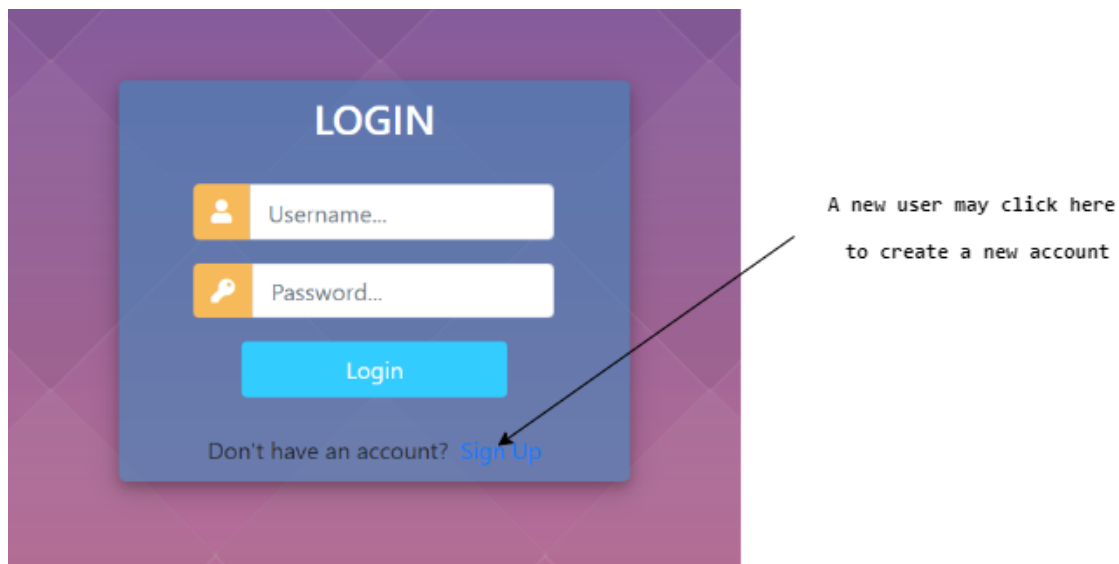
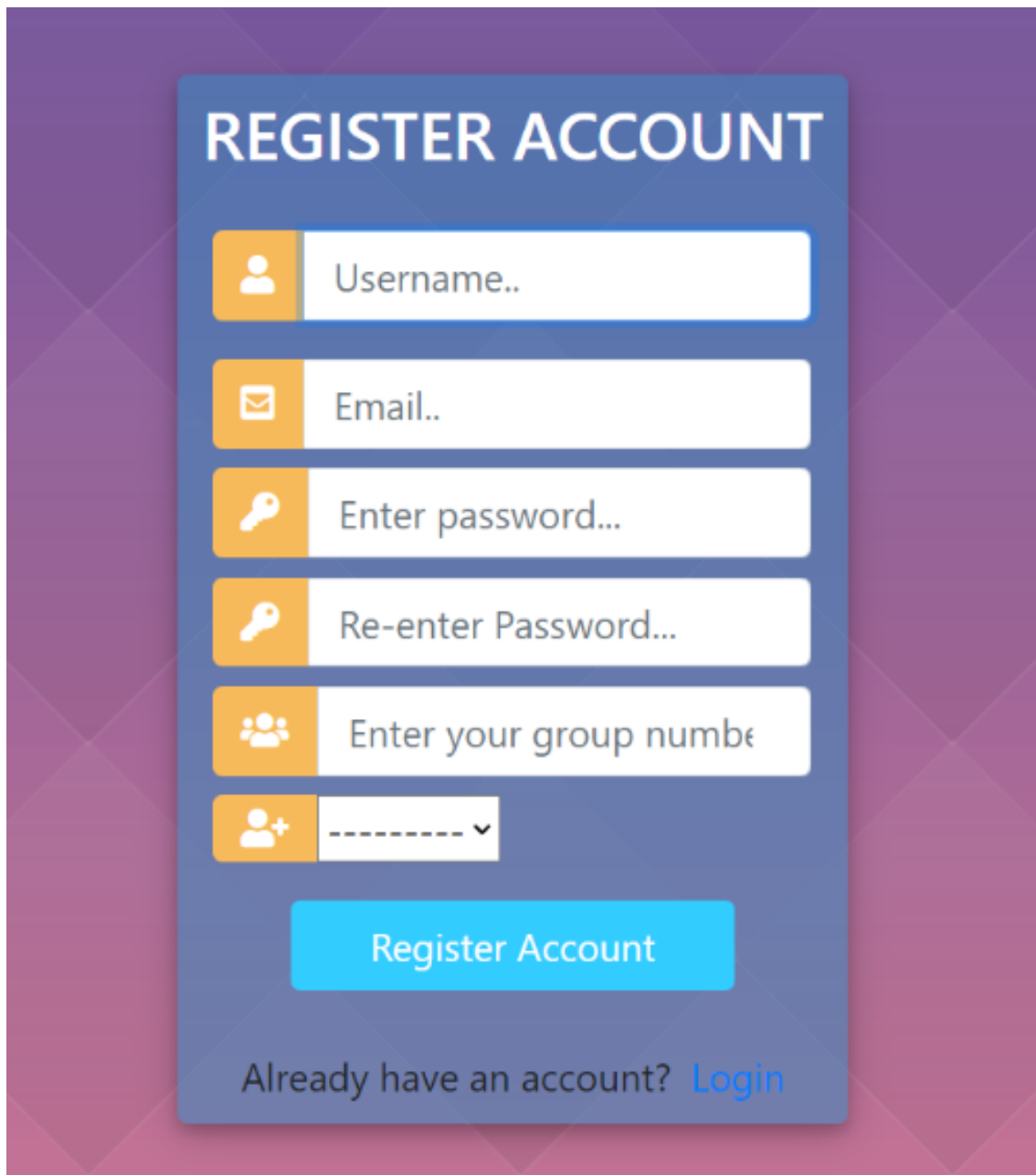


Figure 4.4: Login Page

Register Page

From the login page, the user can transition to the register page as shown below. The

register page is very simple. It asks the user to enter their username, email, password, the user group they want to create/join, and their user level.



The image shows a registration form titled "REGISTER ACCOUNT" on a purple gradient background. The form is a light blue rectangle with rounded corners. It contains six input fields, each with an orange icon on the left: a person icon for "Username..", an envelope icon for "Email..", a key icon for "Enter password...", another key icon for "Re-enter Password...", a group of people icon for "Enter your group number", and a person with a plus sign icon for a dropdown menu showing "-----" with a downward arrow. Below the fields is a bright blue "Register Account" button. At the bottom, it says "Already have an account? [Login](#)".

Figure 4.5: Register Page

Add Task Page / options tab

After a users login has been successful, they are then redirected to the 'Add task page' (if a parent), or view/complete tasks page (if child). These pages are the exact same, however, a parent has the ability to add tasks whereas their children are not shown this feature. A user is able to perform different tasks such as, adding a task (if parent),

deleting a task (if parent), viewing their tasks and updating them (if child) and viewing your accounts metric.

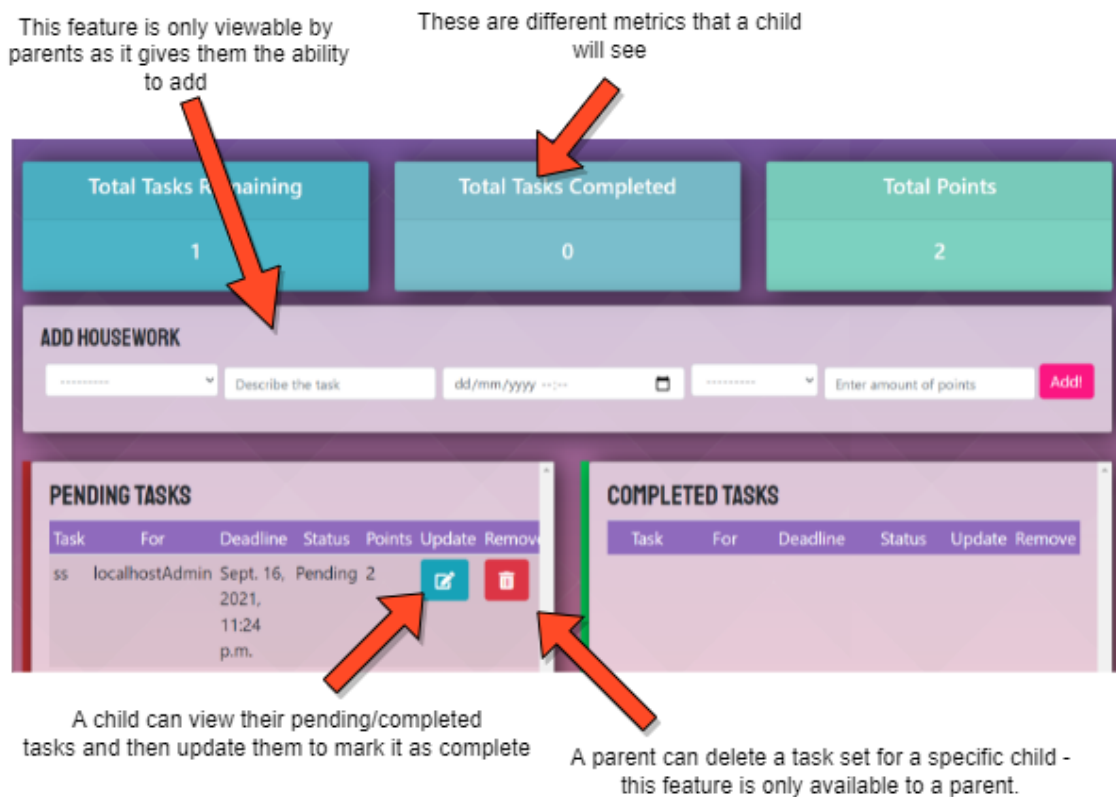


Figure 4.6: Add Task Page/ View Your Task Page

There is an options tab/slider which is available on every page which allows users to transition from one page to another.

Note to reader: The next part of this section is on the next page as placing the screenshot on this page will cause the content to be lost during the shrinking process.

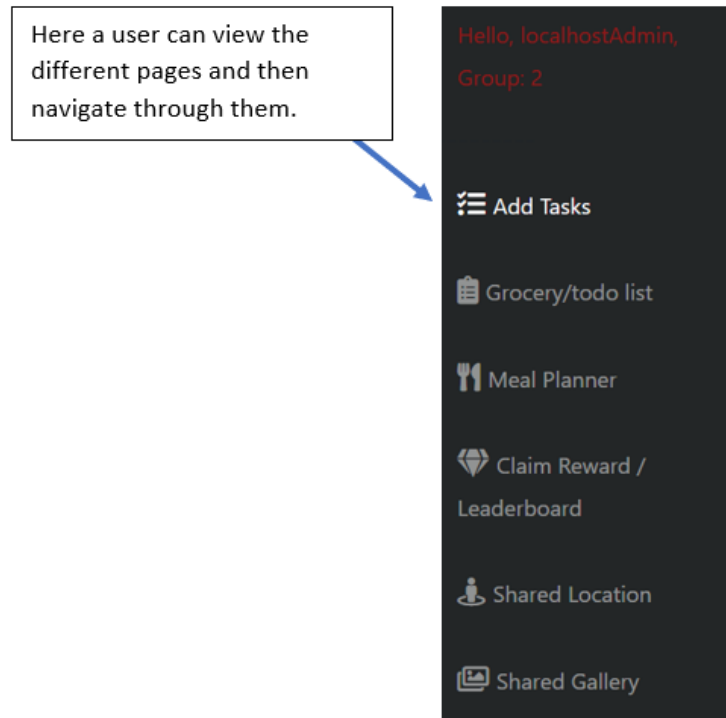


Figure 4.7: Slider/options tab

Shared Grocery - personal todo list

This page allows a family group to add items on their grocery list. This list is shared across the whole group. Users can add items to the grocery list and update them to mark them as completed or uncompleted. However, the personal to do list is a user-only list where the current user can only see their own list hence 'personal' to do list.

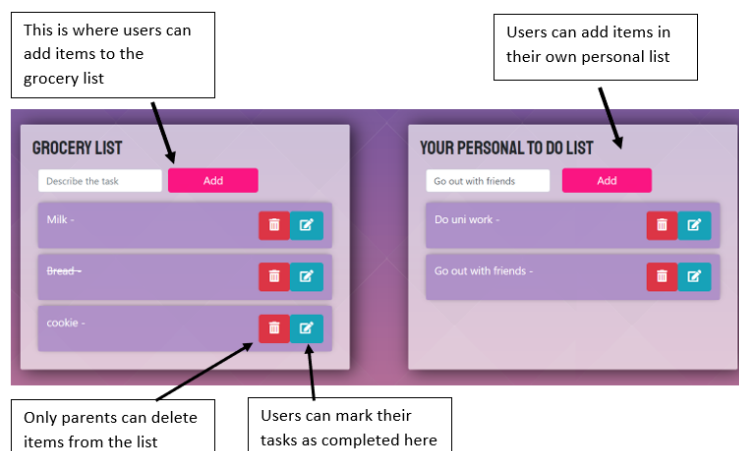


Figure 4.8: Shared Grocery List - Personal todo list

Meal Planner with Voting system

This page allows parents to create different meals for a specific time of the day. Users besides parents are not able to see the feature which enables parents to create meals; this feature is parent-only. After meals have been created with three different options, the family can then vote on one of the options out of the three for a given meal of the day. After voting, users can see the results and the meals that won by majority vote.

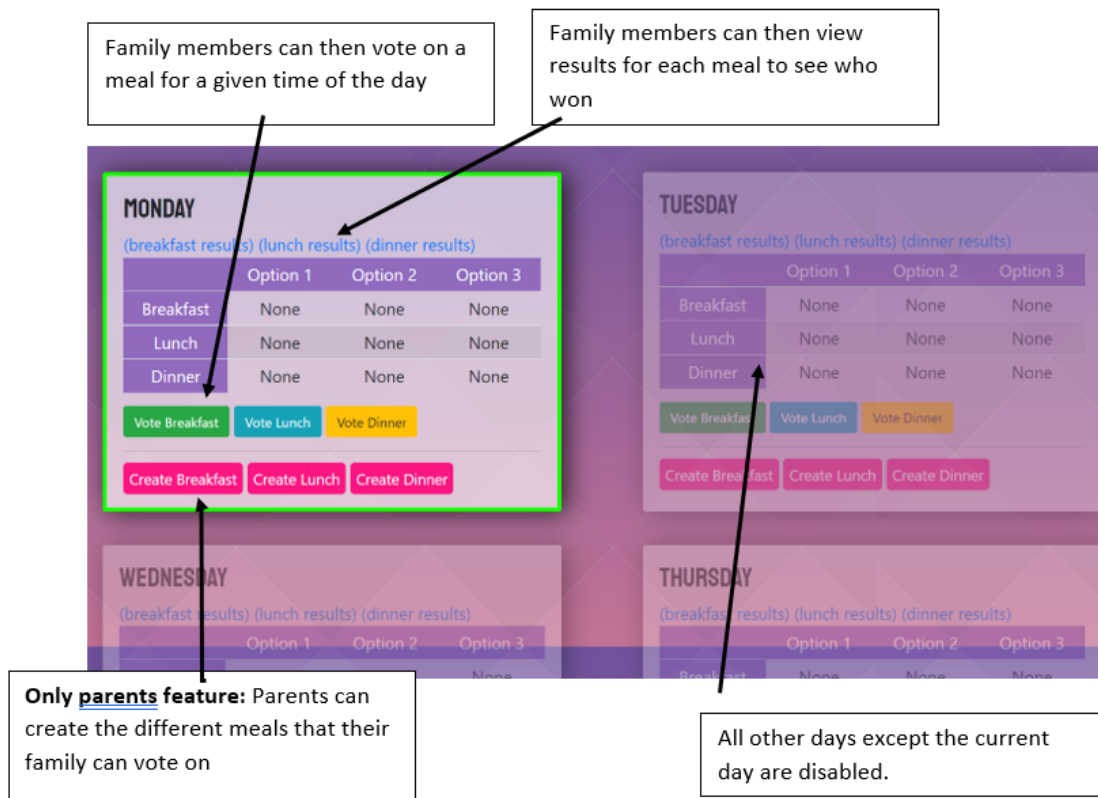


Figure 4.9: Meal Planner Home Page

UPDATE/CREATE

Sunday

Breakfast

Option 1 Option 2 Option 3

eggs rice subway

Info! Only update count if creating a NEW poll!

Update all counts to 0 if creating a new poll

Count 1 Count 2 Count 3

0 0 0

Update/Create

Parents can add the three different options that members can vote for

Parents can choose either 'breakfast', 'lunch' or 'dinner'

They can reset the votes to 0 when creating a new poll

Figure 4.10: Parent view only: Creating/updating meals

RESULTS

eggs — 1

None — 0

None — 0

Total — 1 // Winner — eggs

Figure 4.11: Results page

Add/Claim Rewards

This page has a parent-only feature which allows parents to add rewards that their children can claim for 'x' amount of points. There is also a leaderboard which will show the top 5 users who have earned the most points. The page will show children the rewards

they have claimed and show parents a list of users that have claimed rewards.

The screenshot shows a web interface with several sections. Annotations with arrows point to specific features:

- ADD REWARDS:** A form with input fields for 'Reward' (containing 'Ps5') and 'Points' (containing '100'), and an 'Add' button. Annotation: "Parent only feature: this feature allows parents to add rewards for a certain amount of points".
- CLAIM REWARDS:** A table with columns 'Reward', 'Points', and 'Claim'. It lists 'Ps5' with '100' points and a 'Claim' button. Annotation: "Here members can see the rewards and try claim them If they have enough points".
- LEADERBOARD:** A table showing users and their total points earned. Annotation: "The leaderboard will show the users with the most amount of points earned over a life-time period".
- YOUR CLAIMED REWARDS:** A section for the user's claimed rewards.
- ALL CLAIMED REWARDS:** A section for all claimed rewards, with a 'Parent view only' link. Annotation: "Parent only feature: here parents can see a list of people who have claimed rewards."

#	Name	Total Points Earned
1	localhostAdmin	4
2	testParent	0
3	group2parent	0
4	haneefchild	0

Reward	Points	Claim
Ps5	100	Claim

Figure 4.12: Add/Claim Rewards page

The screenshot shows a modal titled 'CLAIM YOUR REWARD'. It contains a question 'Are you sure you want to claim your reward : Ps5 ?' and a 'Submit' button. Below this, a red error message is displayed: 'You dont have enough points!'. An annotation points to the error message: 'A user trying to claim rewards, however, they do not have enough points so an error message appears.'

CLAIM YOUR REWARD

Are you sure you want to claim your reward : Ps5 ? [Submit](#)

You dont have enough points!

Figure 4.13: Error message when claiming reward

Shared Location

This page allows each member in a specific group to track each others location by pin-pointing their last 'checked-in' location through the help of Google Maps API and the Geolocation API. For security reasons, chrome and many other browsers do not let someone to be tracked unless the user themselves first agree to turn their location on. Even after that, a given member's location is only updated whilst they are on this current page. This means, if they turn off the app or are on another page, their location stops being tracked - this is all due to the security features browsers have placed in order to prevent any potential security breaches.

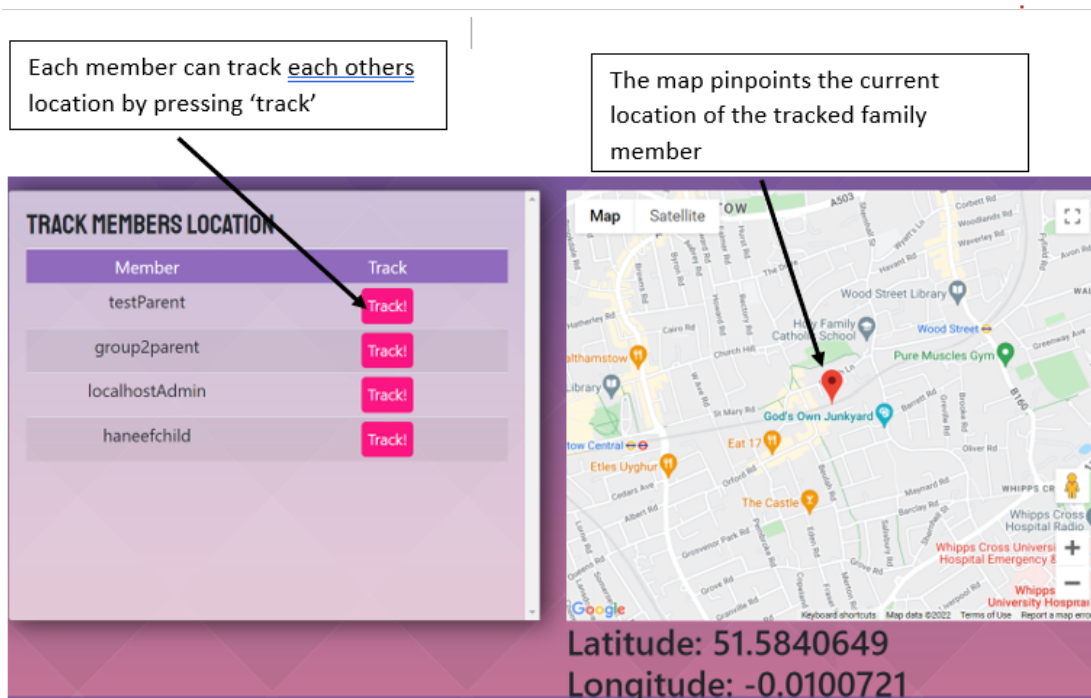


Figure 4.14: Shared Location Web Page

Shared Gallery

This page allows members in the family to add photos to their 'shared gallery' where everyone is able to see the pictures. With the pictures you may add their details like describing the picture and adding its address. Another user can then press 'View Details' on a given picture and view it at a larger size as well as find out how to get to the location of the picture from their current location given the fact that there is an address for the picture that was added.

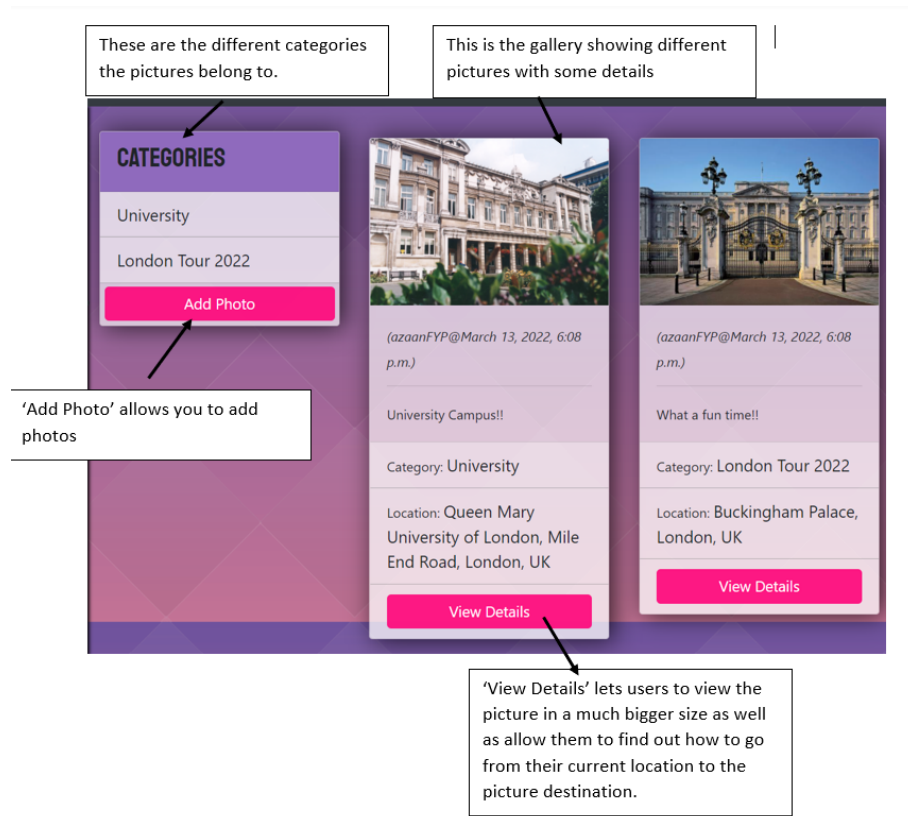


Figure 4.15: Shared Gallery Main Page

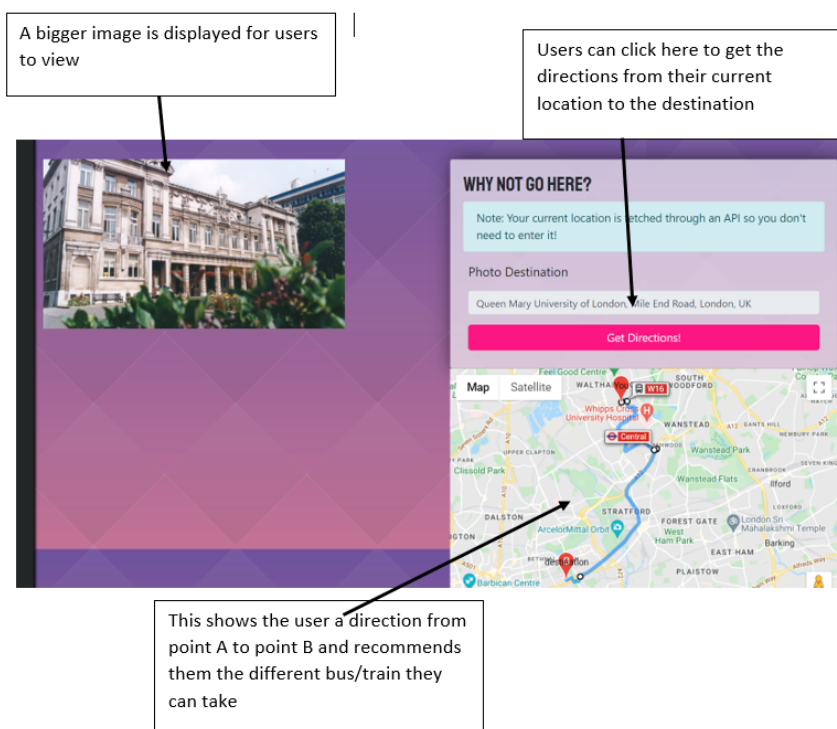


Figure 4.16: View Details Page showing the picture in a bigger format as well as the map from current location to picture location

Messaging System - Asynchronous

This page allows members in the family to communicate with each other through messaging asynchronously. A user will first join their chat room (one for each group) with their username and then can send messages to all other members in the chat. The chat is a groupchat as the aim of this project is to encourage families to interact with each other and having a groupchat rather than a single chat makes more sense to achieve this aim.

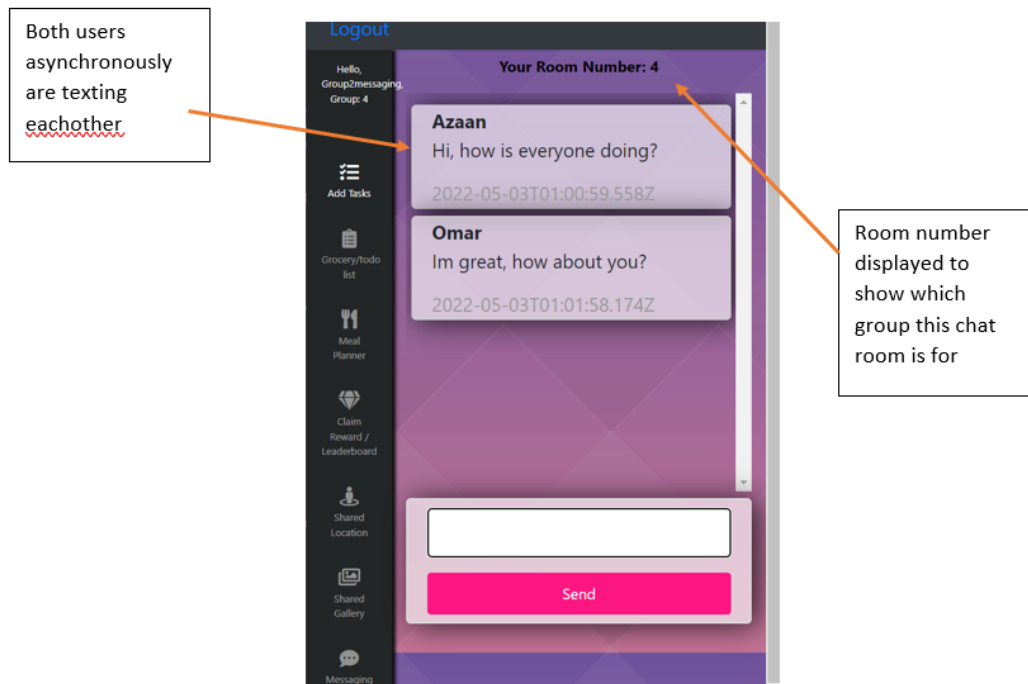


Figure 4.17: Shared Gallery Main Page

4.3 System Architecture, MVC Design Pattern

YourHome is a web application hence it was important to use a design pattern that would support the development of this project. The most suitable design pattern for this project is the Model View Controller (MVC) design pattern. This design pattern separates the three main concerns: the Model, the View and the Controller thus allowing to create a project in a cleaner, simpler and efficient way.

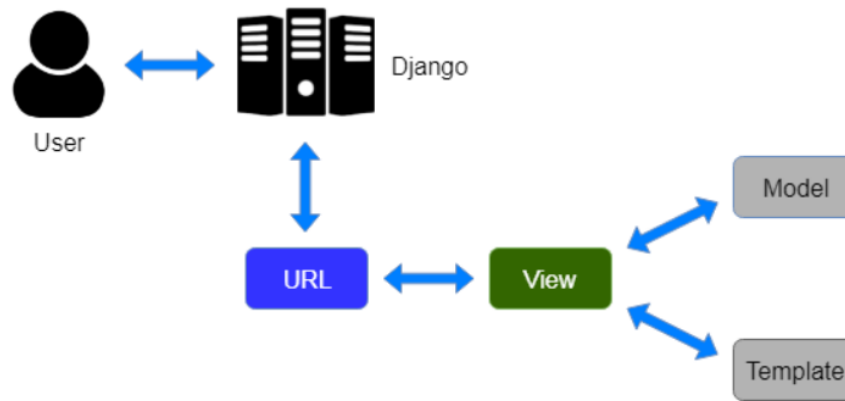


Figure 4.18: MVC Design Pattern (MVC Design Pattern - GeeksforGeeks, 2018)

The model is known for data handling. The model for YourHome is its database where all the data is stored. As this project is based in Django, the database used is SQLite. The model will update the view; the view represents and handles the different elements in the graphical user interface and decides how the data from the model will be displayed. The controller is used to handle and manipulate the model in this case the controller is Django. An example from YourHome would be a user adding a photo in the shared gallery. The controller will then manipulate the model and add it into the database which in turn will update the view and Django will map the template into a URL for the user to access.

4.4 Model Methodology

It was very important to choose a sensible software design method for this project as it was a year long project with many API's and many features which meant that thousands of lines of code was written and therefore, testing becomes harder. The design method chosen was an agile design method. This method allowed bigger tasks to be broken into smaller ones; once a feature was completed on a web page, it would then undergo testing to see if it meets the requirements. During the testing phase, the new feature implemented would be uploaded onto a Git branch which was called 'pre-approval', this was where all new features would be placed into for the testing phase. After a feature has passed the testing phase, it would then be merged onto the 'main branch' of the Git repository. This process would be repeated for every new feature allowing for small increments to take place thus being an agile method.

5 Deployment

The application was successfully deployed through Heroku - a cloud platform enabling developers to publish their website for public use. Publishing the website on Heroku was a tough process and therefore I followed the following tutorial on YouTube: [Click to see video](#).

To access the application please [CLICK HERE](#) or enter the following url <https://azaan-fyp-live-cli.herokuapp.com/login/> into your browser.

You may login with the following credentials:

username: testuser_1

password: testing786

or, you may register your own user. Please make sure you are aware that when registering you have the option of choosing a family group and the option to be a parent account or a child account.

6 Implementation

This chapter will go through the different software technologies used and how they were implemented to build this web application. Furthermore, this chapter will discuss how the main features of the application were built as well as discuss how the API's were implemented.

6.1 Rationale for Platform and Technologies

6.1.1 The Django Framework

This project is implemented using Python due to it being a language which promotes simplicity and efficiency. It is known for writing high-end complex code with it being simple to write due to the easy-to-learn syntax which allows the code to be executed much quicker than most programming languages. Additionally, Python provides many in-built libraries and functions which supported the build of this application. To make this a web application, the Django framework was used. The reason for choosing this framework was that it a high-level Python web framework which promotes the rapid agile development of secure and maintainable websites (Django introduction - Learn web development — MDN, 2022). Also, Django focuses on increasing productivity by reducing development time as it takes care of many different aspects of web development.

6.1.2 Bootstrap

Bootstrap was used due to it being a popular front-end framework which is used to create responsive light-weight websites through the help of Css, HTML and JavaScript. It also provides pre-built components for design patterns and has a responsive grid layout system to suit the need of the device that the application is being used on.

6.1.3 AJAX

AJAX was used due to it being a very popular and simple front-end script which allows an application to be a Single Page Application where upon submitting a form the page will not refresh and data will asynchronously update. AJAX was specifically used for the

messaging system so that when a user sends a message, the message will automatically update without forcing the page to refresh.

6.2 API's and Libraries used

This section will go through the different API's and libraries used to develop the features of this web application.

minted

6.2.1 GeoLocation Google Maps API

The GeoLocation API is a JavaScript based API which enables the users location to be tracked given that the user allows this. This functionality is used in the application for the 'Shared Location' page where each user in a group can find out the location of another user. When a user first visits this page, they will receive a prompt asking if they would like to share their location or not - this is due to security purposes. This is a high-level view of how everything works: Prompt user to share location through using the code `navigator.geolocation`, if user accepts then use `showLocation(position)` function to get the users latitude and longitude and then send those values to the backend using AJAX as shown below.

```
function showLocation(position) {
  console.log(position)
  var latitude = position.coords.latitude;
  var longitude = position.coords.longitude;
  document.getElementById("pos").innerHTML="Latitude: " + latitude + "<br>Longitude: " + longitude;
  //ajax is used to send the location of request.user and save it in the DB
  $.ajax({
    url: '/sharedLocation/',
    // post for security reason
    type: "POST",
    // data that you will like to return
    data: {
      xlat: position.coords.latitude,
      ylong: position.coords.longitude,
      csrfmiddlewaretoken: '{{ csrf_token }}'
    },
    error: function (xhr, textStatus, errorThrown){}
  });
}
```

Figure 6.1: code to get the users coordinates and send it to backend

Then run the `initMap()` function to retrieve the position coordinates from the database and then plot it on the map using Google Maps API. The snippet code below demonstrates how the Google Maps API is used to create a new map. `center:myLatLng` is used to reflect

the location on the map created, where myLatLng are the users coordinates retrieved from the database.

```
const map = new google.maps.Map(document.getElementById("map"), {  
  zoom: 15,  
  center: myLatLng,  
})
```

```
//the function below receives the lat and long from the backend and then plots it on the google maps API  
function initMap() {  
  const myLatLng = {lat: parseFloat("{{one}}"), lng: parseFloat("{{two}}")};  
  const map = new google.maps.Map(document.getElementById("map"), {  
    zoom: 15,  
    center: myLatLng,  
  });  
  
  // the function below allows a 'marker' to be placed on the map depending on the lat and long positions  
  new google.maps.Marker({  
    position: myLatLng,  
    map,  
    title: "Map",  
  });  
}
```

Figure 6.2: code to fetch coordinates from backend and display on Google Maps with marker

6.2.2 Places API

Places API is an autocomplete API used when entering a location. So for example, as seen in the figure below, when entering the first few letters of 'Queen Mary' such as; 'Queen Ma...' the Places API will autocomplete it with the closest accurate location for you to select.

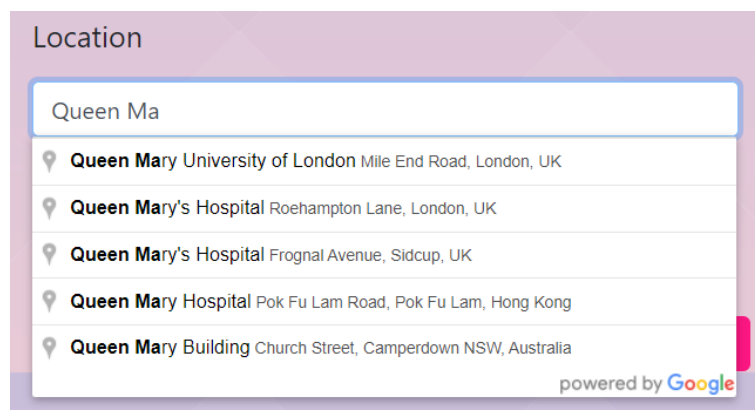


Figure 6.3: Auto completing location using Place API

```
function placeSearch(){
  //referencing the input box in the html form
  let input = document.querySelector('#searchLocationInput');
  //using the api to create a new object which takes in our input (search box)
  let autoComplete = new google.maps.places.Autocomplete(input);
  google.maps.event.addListener(autoComplete, 'place_changed', function () {
    var place = autoComplete.getPlace();
    //storing the lat and long coordinates which then are stored on the backend
    document.getElementById('cityLat').value = place.geometry.location.lat();
    document.getElementById('cityLng').value = place.geometry.location.lng();
  });
}
```

Figure 6.4: Code for Auto complete Place API

6.2.3 The Directions API

The Directions API, by Google, is then used when a user clicks an image in the Shared Gallery and they can then further click on 'Get Directions!' which will display them the direction (using The Directions API, by Google) from their current location (using GeoLocations API) to the picture destination.

Please Note: The figure below is a zoomed out version - if a user zooms in they can clearly see how to go from their location to the bus stop, which bus to catch and then where to get off. In short, zooming in allows a user to see specific directions to get to their destination.

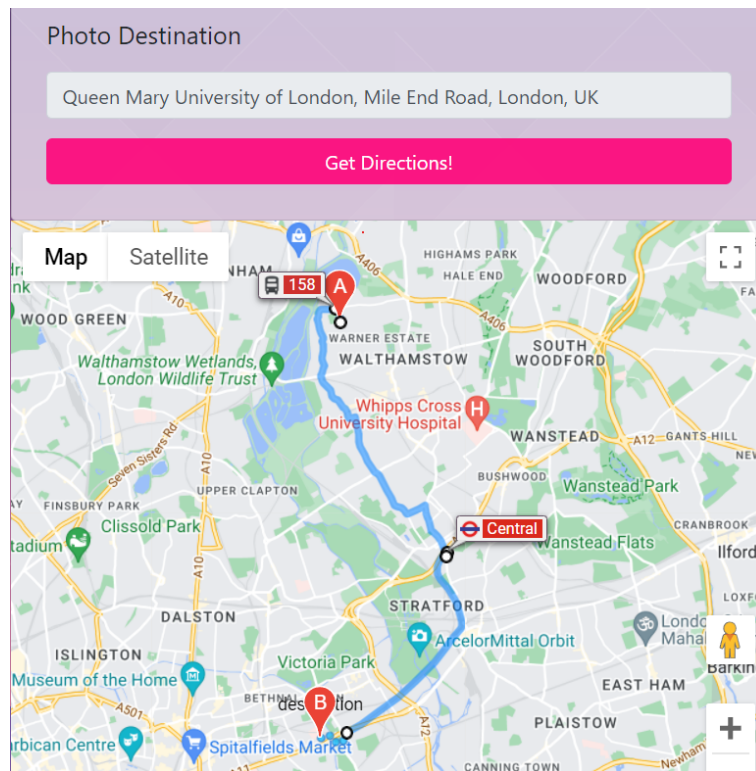


Figure 6.5: Showing the directions from Point A to point B

```
//function below is being used to create the map and output the results into a div precreated in the DOM
//we input the current location as a parameter
function initMap(location){
  console.log(location);
  //place the current location onto google maps API
  directionsMap = new google.maps.Map(document.getElementById('map'), {
    center:location,
    zoom: 16
  });
  directionsService = new google.maps.DirectionsService();
  directionsDisplay = new google.maps.DirectionsRenderer();
  directionsDisplay.setMap(directionsMap);
  //destination lat lng coords imported from photos model
  let destination = new google.maps.LatLng("{{lat}}","{{longi}}")
  //this function is then used to calculate the route from point A to point B
  calcRoute(location, destination);
}
```

Figure 6.6: Code for the, The Directions API

6.3 Core Functionalities and Models

This section will go through a few core functionalities and their models. Code snippets will be provided for support.

6.3.1 Login Page

Both users, parents and children, will need to have an account to access their portal. For this functionality, I have used the libraries provided by Django: `authenticate` and `login`. The user will enter their username and password and will press login. This will then activate the backend function `loginPage(request)` in `views.py` which will then use the `authenticate()` function provided by Django and will try compare to see if the credentials are correct as following `user = authenticate(request, username=username, password = password)` . If they are correct then users will be redirected to the 'addTask Page', else they will be shown a message stating that their credentials are incorrect.

```
def loginPage(request):  
  
    if request.method == 'POST':  
        username= request.POST.get('username')  
        password = request.POST.get('password')  
        #comparing to see if the credentials match  
        user = authenticate(request, username=username, password = password)  
        #if user exists and credentials match then log them in  
        if user is not None:  
            login(request, user)  
            return redirect('addTaskPage')  
        #if credentials are incorrect, then tell user!  
        else:  
            messages.info(request, 'Username or Password is incorrect!')  
  
    context={}  
    return render(request, 'management/login.html', context)
```

Figure 6.7: Code for authenticating the user

Django has different ways to allow developers to implement their User model. For ease of purpose Django has a prebuilt user model and allows developers to extend that user model by adding fields that they require. For this project, I created a `UserProfile` model, `class UserProfile(models.Model)` where I then used `OneToOneField` to refer to the User model provided by Django as follow: `user = models.OneToOneField(User, on_delete=models.CASCADE)`. The code snippet provided below shows the user model in depth and the fields it contains.

```

class UserProfile(models.Model):
    TYPE = (
        ('Parent', 'parent'),
        ('Child', 'child'),
    )
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    userGroup = models.IntegerField()
    userType = models.CharField(max_length=200, null=True, choices = TYPE)
    points = models.PositiveIntegerField(default=0)
    totalPointsGained = models.PositiveIntegerField(default=0)
    lat = models.CharField(max_length=200, null = True ,default=0)
    longi = models.CharField(max_length=200, null = True,default=0 )

```

Figure 6.8: code for the User model in models.py

6.3.2 Add Task Page

First and foremost, the Add Task page should only be accessed if a user has logged in and so to prevent non authenticated users to accessing pages that only logged in users can access, we have to decorators provided by Django. I have placed the `@login_required(login_url = 'login')` decorator on top of my `addTaskPage` function in `views.py` which essentially makes sure that non authenticated users can not access this page.

As this page allows Parents to add a task, I used Djangos forms library in `forms.py` which allows me to create a form with all the fields in the `Task` model. I then use the form in `views.py`, `form = addTaskForm()`. I then check if the request method is post by `if request.method == 'POST'` and if this is True then the form is saved by `form.save()`. The figure below shows the code snippet for this part.

```

form = addTaskForm()
if request.method=='POST' and 'addTaskModelFormButton' in request.POST:
    form = addTaskForm(request.POST,{ 'taskAuthor':request.user})
    if form.is_valid():
        form.save()
        taskAuthorx = form.save()
        taskAuthorx.taskAuthor = request.user
        taskAuthorx.save()

```

Figure 6.9: Code for adding a task

As you may see in the figure above, after saving the form I have then added the author of the task to the form separately and then saved it to the database by

```

taskAuthorx.taskAuthor = request.user
taskAuthorx.save()

```


6.3.3 Meal Planner Voting and Results Functionality

The meal planner allows parents to set up 3 different meals for each part of the day, breakfast, lunch, dinner, and then allow children to vote for 1 meal out of the 3 for each part of the day. This section will explain the functionality of how the voting takes place and how the results are decided.

6.3.3.1 Voting Functionality

A user is presented three different options from the poll. When a user selects an option, for example if `selected_option == 'option1'`, the counter will then increase by 1 as follow: `poll.option_one_count+=1`. The counter increasing will reflect the field for this model in `models.py` which is `option_one = models.CharField(max_length=30,default='None')`. The figure below shows the overview of the code explained.

```
poll = mealVotePlanner.objects.get(pk=poll_id)
if request.method=='POST':
    selected_option = request.POST.get('poll','')
    print('is:',selected_option)
    if selected_option == 'option1':
        poll.option_one_count+=1
    elif selected_option=='option2':
        poll.option_two_count+=1
    elif selected_option=='option3':
        poll.option_three_count+=1
    else:
        return HttpResponse(400,'Invalid form')

poll.save()
```

Figure 6.10: Code for voting poll

6.3.3.2 Results Functionality

After the voting has commenced, users will be able to see live results of the leading meal option or essentially the winner. `poll = mealVotePlanner.objects.get(pk=poll_id)` fetches the poll for the specific meal eg. Breakfast, lunch or dinner. The code below loops

three times through each option and compares the count for each of the three options. Each time if a given meal option is higher than the current highest meal option then the current option will become the highest for example,

```
high = 0
if poll.option_one_count >= high:
    high = poll.option_one_count
```

full example is shown below for all three options of a given meal for a given part of the day.

```
high = 0
for x in range(3):
    if poll.option_one_count >= high:
        high = poll.option_one_count

    if poll.option_two_count >= high:
        high = poll.option_two_count

    if poll.option_three_count >= high:
        high = poll.option_three_count
```

6.3.4 Chat Functionality

The chat system allows members in a given group to message each other. When a user first clicks the messaging link, it will ask them to enter a personal username. A user will then enter their chat room which is unique to every family group. For this chat system to be asynchronous, I have made use of AJAX which allows me to send and fetch data (messages) asynchronously without having to force the page to refresh and update data. In the figure below I have made use of a form which has three hidden inputs, one input to let the user enter their message and a submit button.

```

<!-- here we send messages data to backend to save in DB-->
<form id="post-form">
  {% csrf_token %}
  <input type="hidden" name="username" id="username" value="{{username}}"/>
  <input type="hidden" name="room_id" id="room_id" value="{{room_details.id}}"/>
  <input type="hidden" name="actualuser" id="actualuser" value="{{request.user}}"/>

  <input type="text" name="message" id="message" width="100px" />
  <input type="submit" class="btnCustomised btn-block " value="Send">
</form>
</div>

```

Figure 6.11: Code for voting poll

Upon the user pressing the submit button on the form, the AJAX script is then run which sends the three hidden inputs and the message to the backend view function 'send' which essentially stores the data to the database. This is shown in the figure below.

```

<!-- this ajax script is used to send the data to backend without refreshing the page-->
<script type="text/javascript">
  //this script is run upon submission of form
  $(document).on('submit','#post-form',function(e){
    e.preventDefault();

    $.ajax({
      type:'POST',
      url:'/send',
      data:{
        //The three inputs being sent to the view function 'send'
        //the view function will update the database with these variables
        username:$('#username').val(),
        room_id:$('#room_id').val(),
        message:$('#message').val(),
        csrfmiddlewaretoken:$('input[name=csrfmiddlewaretoken]').val(),
      },
      success: function(data){
        //alert(data)
      }
    });
    //after submission, make the input textbox empty
    document.getElementById('message').value = ''
  });
</script>

```

Figure 6.12: Code for voting poll

7 Testing

This chapter aims to go through the different testing strategies used on the different functionalities of the system. Testing the application aims to highlight and rectify any bugs that may come up. The more tests done, the higher quality the application will be. Code snippets of different test cases will be provided when testing the functionality and usability of the system.

Note to reader: The table is on the next page in landscape mode to avoid any content being loss.

Table 6: Authentication testing

Authentication Testing			
Test Case	Steps	Expected Results	Actual Results
User logging in with existing credentials.	1) Enter username: JohnSmith 2) Enter Password: Mypassword123@ 3) System backend will compare details entered with database. 4) If credentials are correct, system will redirect them to the addTask page.	System accepting the credentials and redirecting to addTask page.	Pass.
User logging in with incorrect login details	1) Enter username JohnSmith 2) Enter wrong password: 123hello 3) System backend will compare details entered with database. 4) if credentials are in correct, system will display message 'Username or password are wrong!'	System will display 'Username or password are wrong!'	Pass.

User logging out	<p>1) Logout button displayed on every page when a user is logged in.</p> <p>2) User presses 'log out'</p> <p>3) System will close the session and log the user out and take them back to the login page.</p>	Log the user out and redirect them to the login page	Pass.
Parent view vs Child view	<p>1) A child should not be able to see some specific features that parents can see. These features are on the following pages:</p> <ul style="list-style-type: none"> - add task page - rewards page - meal planner page <p>2) System differentiates between Child and parent and prevents a child account from seeing features that parents can see.</p>	Successfully differentiate between a child vs parent account.	Pass.

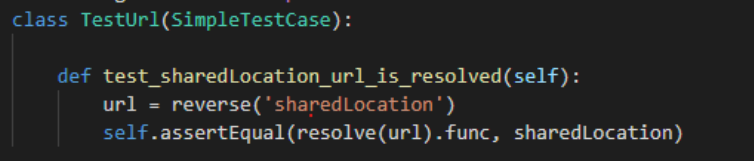
7.1 Unit testing by Django

This section will go through the different unit selenium tests the system has undergone. Unit testing is very similar to the agile method of development. In both cases we focus on smaller parts of the system. In unit testing, we use the divide and conquer method where we break a bigger task into smaller units of tasks and test them separately. Whereas the Selenium testing tool is used to automate tests across browsers for web applications. It's used to ensure high-quality web applications — whether they are responsive, progressive, or regular (Kinsbruner, n.d.).

unittest, a Python library, is used to perform unit tests on different functionalities of the system. It provides with many different tools that allow a developer to create and run tests as well as automate them.

7.1.1 URL Testing

Django provides developers with URL testing tool. URL testing allows Django to take in a URL and then display some sort of response to when that URL is called, this response is in a HTTP format. For this test case, I wrote a unit test for '/sharedLocation' URL which is shown below:



```
class TestUrl(SimpleTestCase):  
    def test_sharedLocation_url_is_resolved(self):  
        url = reverse('sharedLocation')  
        self.assertEqual(resolve(url).func, sharedLocation)
```

Figure 7.1: Unit test for a given URL

The test returned 'OK' as shown in the figure below. This means the test was successful and that it returned an expected HTTP response. I then ran the test for all the URL's and they all returned 'OK'.

```
Ran 1 test in 0.013s
OK
C:\Users\University\Desktop\homeApp>
```

Figure 7.2: Result of URL test

7.1.2 Login Test Case

This test case will automate a simulation of a user logging in. The test case will enter the users details, name and password, and then run it. The test case returned an 'OK' which means that the user successfully logs in and is allowed permission to enter their portal.

```
class LoginTest(TestCase):
    def login(self):
        UserProfile.user = User

        user = User.objects.all()

        username = 'JohnSmith'
        password = 'Mypassword123@'

        client = Client()
        UserProfile.objects.create(username=username, password = password)
        response = client.post('/login/', {'username':username, 'password':password})
        self.assertEqual(response.status_code, 302)
```

Figure 7.3: Unit test for a logging in

```
Ran 1 test in 0.019s
OK
```

Figure 7.4: Result of Login test case

7.1.3 Adding Rewards

This test case ensured that rewards and points were being added correctly. Before the actual test is run, it is ensured that the user is logged in. The reward and points were added directly through form input and printing the response ensured that the test ran successfully.


```

class LoginTest(TestCase):

    def test_check(self):
        #ensures that a user is logged in prior to the test
        username = 'testuser'
        password = 'password132'
        client = Client()
        self.client.post('/login/', {'username':username, 'password':password})
        #actual test is below
        rewardName = 'Elden Ring'
        rewardPoints='200'
        response = self.client.post('/rewards/', {'rewardName':rewardName, 'rewardPoints':rewardPoints})
        print(response.content)

```

Figure 7.5: Unit test for adding rewards

7.1.4 Summary of Unit Testing

Table 7: Unit Testing For All Functionalities

Test Case	Expected Result	Method	Actual Results
Add Task	Task is added.	POST	Pass.
Update/ delete task	Task is updated/ deleted.	PUT/DELETE	Pass.
Add items in grocery/ todo list	Items added.	POST	Pass.
Add meals in meal planner	Meals added.	POST	Pass
Vote for a given meal	Voting takes place.	POST	Pass
Update/delete meals	Meals are deleted or updated.	PUT/DELETE	Pass
Add a reward	Rewards are added.	POST	Pass
Claim reward	User should be able to claim a reward	POST	Pass
Post picture in gallery	Picture is added in gallery	POST	Pass
Messaging system	Message sent successfully with AJAX	POST	Pass

7.1.5 Shared Location Google Maps API Selenium Test Case

In this test case we look to see if the google maps API on the 'sharedLocation' page is working as expected. Selenium will check if the front-end + the API is running as

expected. If everything runs correctly, the test should return an output 'OK'.

As we use selenium, we need to ensure a user is logged in and accesses the shared location page step by step as a normal user would. In the code below we first find the input elements and the submit button from accessing the page's CSS for example, `driver.find_element_by_name('username')`. We then pass the username into the input through the following code `username.send_keys('testuser')`, the password will be passed in a similar way.

```
class apitest(LiveServerTestCase):

    def testhomepage(self):
        path = r"C:\Users\University\Desktop\homeApp\chromedriver.exe"
        driver = webdriver.Chrome(path)
        driver.get('http://127.0.0.1:8000/login/')
        wait = WebDriverWait(driver, 5)

        try:
            logi_button = wait.until(EC.presence_of_element_located((By.XPATH, "/html/body/div/div/div/div[2]/form/div[3]/input")))
            logi_button.click()
            print('first test passed')
        except:
            print('first test failed')

        #find the elements you need to submit form
        username = driver.find_element_by_name('username')
        password = driver.find_element_by_name('password')
        submit = driver.find_element_by_class_name('login_btn')

        #populate the form with data
        username.send_keys('testuser') #change this lol asap
        password.send_keys('password132')

        #submit form
        submit.send_keys(Keys.RETURN)
```

Figure 7.6: Code for logging in

On the shared location page the map div is empty when a user freshly comes to the page. We then find the 'Track' button and make our automated user click that button as shown in the code below by `track_button.click()`. After the 'Track' button is clicked, we then find the div element where the map is placed in and compare it with the empty div element. If the new div element's XPATH changes, we can then conclude that the map has successfully been displayed and the test is successful. The code is shown in the figure below.

```

try:
    track_button = wait.until(EC.presence_of_element_located((By.XPATH,
        "/html/body/div/div/div/div[1]/div/table/tbody/tr[2]/td[2]/form/button")))
    track_button.click()
    print('tracking button clicked!')
except:
    print('tracking button not clicked')

try:
    # compare x path of map
    print('testing map xpath')
    map_xpath = wait.until(EC.presence_of_element_located((By.XPATH,
        "//*[@html/body/div/div/div/div[2]/div[1]/div/div/div[2]/div[2]")))

    if map_xpath.is_displayed() == True:
        print('Map is showing')
    else:
        print('Map not showing')
except:
    print('logic did not work')

```

Figure 7.7: Code for map test

```

first test passed
GOING TO SHARED LOCATION WORKED
tracking button clicked!
testing map xpath
Map is showing
.
-----
Ran 1 test in 8.151s
OK

```

Figure 7.8: Output for map API test

7.1.6 Summary of Selenium Testing

Table 8: Selenium Testing

Test Case	Expected Result	Method	Actual Results
Shared Location map API Testing	The current location of a user is shown.	GET	Pass.
Shared Gallery Google maps API testing + directions API testing	Google maps displays how to get from current location to the destination where the picture was taken.	GET	Pass.
Autocompletion of input typed. (Place autocomplete API)	When a user starts typing the first few words of a location, the API should autocomplete and recommend them places based on their input.	GET	Pass.

7.2 Cross-platform Browser Compatibility Testing

This test ensures that no matter what device or browser platform a user may use, they should still be able to access the web-application in all circumstances. In the tests below, it can be said, with confidence, that the application runs on all the devices and browsers stated without losing any core functionality.

The test was inducted on all main stream browsers and on Apple and Android devices. The output of the tests were all successful.

7.3 Summary of Testing

Testing is an important aspect of this report. All different functionalities in the application are tested in-depth to ensure that there are no bugs and errors when the system is made live. Table 7 and Table 8 include functionalities that were mostly carried out through

script automation but some of the functionalities were carried out manually. As a result, all the functionalities were running as expected. The only issue that I came across was that users were able to claim a reward even though they did not have enough points. This was quickly rectified by adding further checks to see if a user has enough points.

8 Evaluation

This section undergoes the different evaluation techniques used to demonstrate the strengths and weaknesses of the project. Additionally, receiving user feedback is always appreciated as you are receiving first-hand primary feedback from people who are potential stakeholders of the application.

8.1 Usability/Normal Evaluation Metrics

As seen below, I have carried out a questionnaire to evaluate my application based on many different factors including, Usability, Effectiveness, Efficient to use, Easy to Learn etc. Additionally, there are also questions which ask the user about their opinion on the application and its ability to meet its goal of reducing family division. This allows me to ask specific questions to my stakeholders in order to receive a high-quality response which would ensure if the application meets their needs or not. 8 different families were asked to participate in this questionnaire and their results are shown in the next section.

The questions asked are all related to the different Usability Metrics and are the following:

1. Intuitive design - How easy do you find navigating from one page to the next in the application?
2. Ease of Learning - Which features do your family members use the most and is it an easy process to achieve your task?
3. Error Tolerant - Can you use the API's efficiently without coming across any errors?
4. Meaningful Functionalities - What features do you think help reduce family division?
5. Subjective Opinion - Is this system useful to you and your family - does it reduce family division? If so, will you recommend it to other families?

8.2 Results and Analysis

8 Families were asked to take part in this opportunity of answering the questions. They were randomly picked without any prior knowledge of any family division in their house-

hold. This further shows that despite being randomly picked, many families are facing family division.

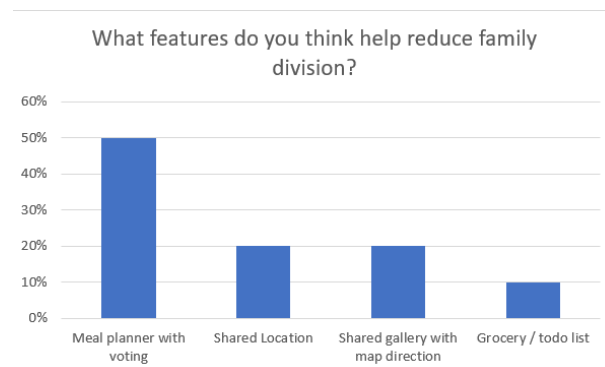


Figure 8.1: Barchart displaying the features users use the most to reduce family division

As seen in the figure above, when the 8 families were asked about the feature that they think has helped reduce family division, they said it was the meal planning with the voting functionality. When asked parents why they used this feature a lot, most of them said that their children gave a positive outcome from using this feature as allowing children to vote for their meals get them excited for the day and really allows them to interact between each other. Additionally, it was mentioned that the voting functionality is unique to YourHome and no other application possesses this unique feature thus the popularity for this feature.

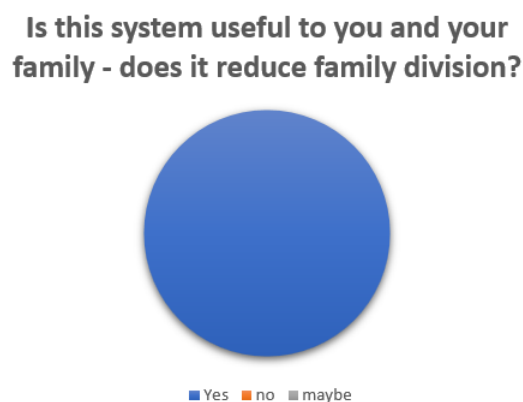


Figure 8.2: Is this system useful to you and your family - does it reduce family division? If so, will you recommend it to other families?

The figure above shows that when asked users about if this application is useful and reduces family division, all 8 families agreed and said 'yes'. This further shows that this

application has achieved its aim of reducing family division by introducing a way where family members can interact with each other freely.

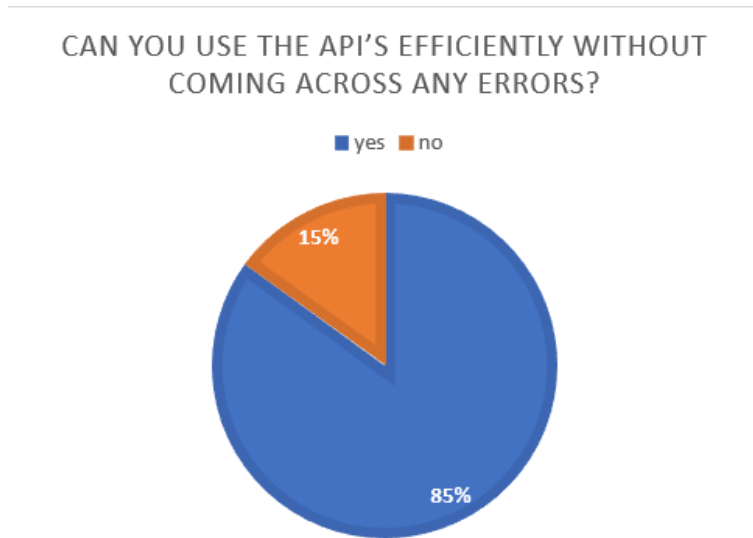


Figure 8.3: Can you use the API's efficiently without coming across any errors?

As seen in the figure above, 85% of the families said that they did not face any issues using the API's whereas 15% of the families said that they did face problems. The 15% that said they faced issues were asked what issue in specific they faced. They specifically mentioned that the tracking functionality did not work as whenever they tracked a family member it would not always be the correct location and or if they were on a different page of the application, their location would not update. Both issues are common and normal and have been addressed by Google and Mozilla. Users sometimes may not see their actual location as the network you may be on may connect with a local network tower which would be at a 10-15mins distance from your actual location. Also, your location does not update automatically if you close the app or browse a different page for security reasons and this change was brought in by Mozilla recently.

8.3 Summary of Evaluation

Overall, looking at the analysis it can be seen that the sample space who were questioned had a positive view of the system. However, there were a small number of people who rightfully objected the functionality of the Shared Location page where users could track each other through an API. This is due to the fact that the location would not be 100%

accurate all the time and that it would not update 'on-the-go'. These are common issues and have already been addressed by Google and Mozilla. Users may see this as an issue but the developers of the API have addressed this problem by stating that they are focusing on users' security by preventing some features to be 100% 'real-time'. To overcome this issue, more sophisticated API's would need to be considered whilst keeping users' security a high priority.

To test the accuracy of the functionality, further surveys were done, on 20 users, as seen below. It is clear that only 5% of the users faced an issue where their location would not be 100% accurate. In other words, 1 in 20 people would come across this problem.

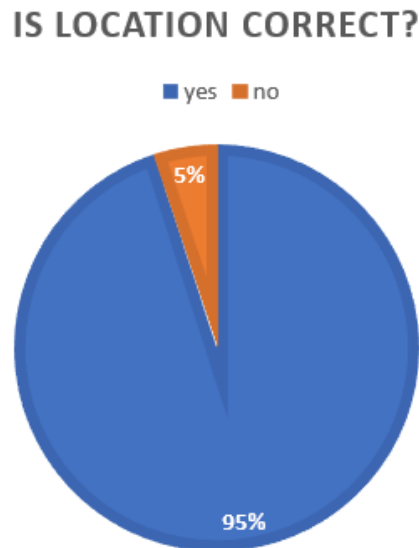


Figure 8.4: Results for survey about location being accurate

Additionally, as seen in figure 7.2, 100% of the sample space agreed that this application has helped their family and has allowed them to reduce family division. This is a positive outcome for this project as the aim was to ensure that families develop a good relation through reducing the gap they have between themselves. The outcome of the analysis proves that this project has met its aims.

9 Conclusion

This chapter intends to conclude the project and report. It also aims to go through the achievements and the different challenges I came across as well as any future work should I continue this project.

9.1 Achievements and Challenges

I can confidently say that the project itself has proven to be a huge success. It has allowed me to apply my knowledge which I have gained in the past 3 years from a Computer Science background, and has enabled me to explore new areas in different domains. It is evident that the aims and objectives listed in chapter 1 of this report indicated the goals that this project is looking to achieve and it can be said that the aims and objectives have successfully been met. The literature review in chapter 2 allowed me to distinguish my project from others by carrying out an in-depth research about similar systems which then further allowed me to create useful and unique functional and non-functional requirements in chapter 3. All the functional and non-functional requirements were met, but 2, which will be discussed in the section for future improvements. Evidently, as seen in the testing chapter, all the functional and non-functional requirements ran as expected. Finally, the evaluation chapter highlights the different opinion of other users who were asked about the application, and it can be inferred from the results that the aims and objectives for this project were met.

Furthermore, before this project I had no experience in working with different API's. This was a new challenge for me which allowed me to delve into new areas of Computer Science. With hardship, comes ease and I believe I have put in a lot of work into stepping out of my comfort zone which allowed me to understand and implement API's to achieve the main aims and objectives of this project. The aim of this project was to reduce family division by introducing features which would play an important role. Two of those features were the shared location feature and the directions feature which both made a use of different Google API's. I can say this specific aim of the project was achieved through the help of these features.

Finally, embarking this project allowed me to develop many new skills as well as learn many new lessons. My supervisor, Paulo Oliva, has taught me the value of time and how important it is when working on a very important project. Also, his experience in web development taught me how important it is to continuously test my application to ensure that it is running as expected. This project also allowed me to learn the different languages and frameworks for example, HTML, Bootstrap, Django framework, CSS, vueJS and many more. I believe that embarking on this journey has allowed me to develop many important programming skills which would assist in my future career.

Throughout the entirety of this project I faced many challenges. One of the challenges that I came across during the development of this project was with the Geolocation API and the Google maps API. Initially, I had no experience in working with API's and so implementing it in my application became a tough job. This was a core feature for the application and would allow me to meet the aims and objectives stated in chapter 1. When first implementing the API's, it was very difficult to understand how things work. As I had to use two different API's for one feature it created many issues for example, I would first have to fetch the users current location, using GeoLocation, and then would have to map their location onto Google Maps using Google Maps API. After initially implementing the API's and getting them to function, they would output the wrong results. The reason for this was that there was no data of the longitude and latitude being passed from the front-end to the back-end and so they could not be retrieved when placing the pin-point on Google Maps. Therefore, for me to overcome this, I had to use AJAX to pass data to the backend and then further retrieve it to pin-point it on Google Maps.

Furthermore, another challenge I faced was that the location fetched of a user was not always accurate also did not update in real-time. This is a very common and general issue faced from using API's which gather a users current location as they do not want to intrude a users personal data due to security reasons. As of yet, there is no concrete solution to this as this is a new requirement that all browsers have to comply with.

Lastly, managing time and balancing other modules in my final year with the project

was extremely difficult. This may be due to the fact I had to put in extra work at university due to it being online which made learning difficult for me. I spoke to my supervisor about this specific issue, and was told to produce a gantt chart and timetable. This allowed me to focus on smaller tasks of the project step-by-step which is similar to the agile development method. This ensured that I implement all the core features of the application besides one; the shared calendar which is spoken about in the next section.

9.2 Future Improvements

I believe that YourHome can be improved further by adding extra functionalities that and making it more user-friendly and responsive which will in return help achieving the aim of reducing family division. In the future I would look to add the two functionalities which I was unable to add during this year due to time management.

Firstly, I would add a shared calendar using Google Calendar API - this functionality would be extremely useful and will help families manage their timetable better and in sync. All members in the family will be notified of the different events taking place on a given day and it would give them a chance to interact with each other.

Despite having AJAX for the messaging system, I would also revamp other pages to a Single Page Application where I would use Vue.js. This would allow the application to be more responsive and interactive and as the application is targeted at families who may have members who are not good with technology or members who are young (eg. children) who prefer light colours, interactivity and responsiveness from a website.

Furthermore, I would add functionalities which would make the website more secure. For example, instead of children signing up and choosing their family group, I would allow parents to sign up and then create accounts for their children or send their children group invites where when they click the link they automatically join the group.

9.3 Final Remarks

Overall, it is evident from the outcome of the project and the results from the Analysis chapter that this project was a success. It has met all its aims and objectives. There is always room for improvement and this has been mentioned in the section above.

10 Appendix

10.1 Appendix A: Risk Analysis

Please note: The Risk Analysis table, Table 3, is on the next page in landscape mode. This prevents the table from shrinking and thus losing content.

Table 9: Risk Analysis

Description of Risk	Impact Description	Likelihood Rating	Impact Rating	Preventative Actions
Poor time management	This will cause me to fall behind and will therefore will cause the system/project to be incomplete.	Low	High	Make sure to keep an eye on the Gantt chart. Follow each task according to the Gantt chart as this will allow me to understand how difficult a task is and how much time it may require me to complete it.
Learning new skills which may be consuming time	An unnecessary amount of time spent on learning new skills will cause the project to be delayed and incomplete. The quality of the project will be low.	Medium	High	As I am doing the module: Web Programming, and my project is also a web application, I should use the skills acquired in the module to implement my project rather than completely devoting more time to learn more skills where I can use the skills learnt in Web Programming to develop my project.
Work loss	Project being delayed or an incomplete project.	Low	Very High	Having multiple backups on different locations and keep a backup on the cloud as this is the safest method.

Google API's stop working	Many different features of my web application will stop working due to this issue and so the application will fail as the core features that require API's will no longer work.	Low	High	<p>1) Use API's that are reliable, for example Google API's. These are not known to go down.</p> <p>2) Use multiple API's from different sources as this will prevent a single point of failure.</p>
Feature Creep	Implementing more and more features day by day will cause me to forget the aim of the project which will lead to core features not being implemented properly. Therefore, it will lead to an incomplete project with excessive unnecessary features.	Medium	High	Follow the Gantt Chart and refer back to it daily when implementing the different features. Complete the core features first then think of implementing new features.
Unrealistic project	An unrealistic and large project will essentially lead to a project being incomplete. Therefore, the dissertation will also be incomplete.	Low	High	Have regular meetings with the supervisor every 2 weeks so that they can see my progress and put me back on track if need be. Write down SMART aims and objectives which will help in seeing the 'bigger picture' of the project and get it approved by the supervisor.

Unable to implement a given feature	Failure to implement specific core features of the application will lead the project to be delayed and incomplete. The overall project will be incomplete and the aims and objectives will not be met.	Low	High	<p>If a feature can not be implemented, time should not be wasted on it. Rather, other implementable features should be implemented to utilize time. However, for other tasks that are hard to implement, a supervisor meeting should be conducted as they may lead you to the right way for a solution. Other resources such as YouTube or Stack overflow for solutions.</p>
Change in requirements	The project requirements may change during the creation of the project. This could be due to the fact that previous project requirements are not fulfilling what the project is meant to do. So therefore, changing the requirements will ensure the project meets its aims and objectives and creates a solution to the problem statement.	Low	Medium	<p>Constantly reevaluate the aims and objectives of the project and ensure my supervisor is kept in the loop so that they can advise me if they think I'm going off track. Additionally, the project should be referred back to the research gathered and see if the completion of the project will lead to the problem statement being solved.</p>
Unforeseen circumstances	Becoming ill or any other circumstances can cause me to fall behind in my project.	Medium	Very High	<p>Get in touch with my supervisor and head of school right away and inform them of the situation.</p>

References

Taylor, J., 2013. Is Technology Creating a Family Divide?. [online] Psychology Today. Available at: <<https://www.psychologytoday.com/us/blog/the-power-prime/201303/is-technology-creating-family-divide>>[Accessed 20 November 2021].

Bulao, J., 2021. How Fast is Technology Growing? [Statistics and Trends for 2021]. [online] TechJury. Available at: <<https://techjury.net/blog/how-fast-is-technology-growing/#gref>>[Accessed 20 November 2021].

Penner, F., Hernandez, J. and Sharp, C., 2020. Change in Youth Mental Health During the COVID-19 Pandemic in a Majority Hispanic/Latinx US Sample. [online] jaacap. Available at: <<http://www.jaacap.org>>[Accessed 20 November 2021].

Haig, M. (2002) Mobile Marketing: The Message Revolution. London:Kogan Page Ltd.

Hardy, M., 2021. Importance of Family Communication. [online] Family.lovetoknow.com. Available at: <<https://family.lovetoknow.com/about-family-values/importance-family-communication>>[Accessed 20 November 2021].

Ngunan, A. and Regina, O., 2016. Digital divide: A Gap in interpersonal communication amongst nigerian family relationship. [online] GSTF Journal on Media & Communications (JMC). Available at: <<https://link.springer.com/article/10.7603/s40874-016-0003-y>>[Accessed 20 November 2021].

Apple Support. 2021. What is Family Sharing?. [online] Available at:<<https://support.apple.com/en-us/HT201060>>[Accessed 20 November 2021].

Apple Support. 2020. How to share albums in Photos on your iPhone, iPad, and Mac. [online] Available at: <<https://support.apple.com/en-us/HT202786>>[Accessed 20 November 2021].

Play.google.com. n.d. FamilyWall - Happy Family Organization. [online] Available at: <<https://play.google.com/store/apps/details?id=com.familywall&hl=en&gl=US>>[Accessed 20 November 2021].

Adkison, A., 2021. REVIEW FROM ANGELA ADKISON. [online] Play.google.com. Available at: <<https://play.google.com/store/apps/details?id=com.familywall&hl=en&gl=US&review>>[Accessed 20 November 2021].

Chapman, M., 2021. REVIEW FROM MOLLY CHAPMAN. [online] Play.google.com. Available at: <<https://play.google.com/store/apps/details?id=com.familywall&hl=en&gl=US&review>>[Accessed 20 November 2021].

Schowalter, K., 2019. REVIEW FROM KATHERINE SCHOWALTER. [online] Play.google.com. Available at: <<https://play.google.com/store/apps/details?id=com.cozi.androidfree&hl=en&gl=US&review>>[Accessed 21 November 2021].

GeeksforGeeks. 2018. MVC Design Pattern - GeeksforGeeks. [online] Available at: <<https://www.geeksforgeeks.org/mvc-design-pattern>>[Accessed 15 March 2022].

Developer.mozilla.org. 2022. Django introduction - Learn web development — MDN. [online] Available at: <<https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction>>[Accessed 20 March 2022].

Kinsbruner, E. (n.d.). How to Use Selenium Testing Tools — by Perforce. [online] Perfecto by Perforce. Available at: <https://www.perfecto.io/blog/how-use-selenium-testing-tools>.