

Web App Design

Advanced Web Technologies

SET09103

Andrew Taison

Matriculation: 40538519

Word | Page count: 2510 | 7.5

(excl. diagrams, references and appendices)

Submission date: 18th October 2023

Table of Contents

Introduction and description.....	1
Background Research.....	2
Core Features.....	4
Additional Features	6
Site organisation.....	7
User Interface	8
Database design.....	11
Implementation plan.....	12
References	13

List of Figures

Figure 1. HTML date input, before and after clicking	2
Figure 2. Bootstrap modal test	3
Figure 3. Flask flash message used as Bootstrap alert message	3
Figure 4. Organisation tree	7
Figure 5. Base - public home page	8
Figure 6. Manage subscriptions - modal	9
Figure 7. Find a game page	9
Figure 8. My game chats page	10
Figure 9. Initial colour pallet	10
Figure 10. Database ER diagram	11

Appendix

UI wireframes	A
---------------	---

Introduction and Description

"Golf is a game in which you yell 'fore', shoot six and write down five." [1] - Paul Harvey. Although not everyone's cup of tee (pun intended), and I'm sure most have heard the famous Mark Twain quote - "Golf is a good walk spoiled" [1] , I hope everyone can appreciate the social side of the game (as in a lot of sports), it brings people together.

This project is centred around creating a server-side webapp using Python Flask micro-framework. This report sets out to outline the initial steps in design and development of this project along with an initial plan for implementation.

The idea for the project began with a love for golf. Initially with the intention of creating a way to store and review stats (such as best scores for each hole / for a round etc.). However as the idea developed, the focus has become more on the social aspect of the game. As a recently new club member, one of the challenges I found is finding so called 'bounce games' – games organized (usually last minute) with strangers. This is with the intent to meet members, enjoy the social side of a game as well as potentially find a guide around a new or unfamiliar course.

The core of the design is based on allowing users to arrange events and notifying interested individuals if an event is scheduled at their subscribed or chosen clubs, on a given date. Once people have been connected, a mechanism will hopefully be setup to allow event participants to communicate via real-time chat to arrange the finer details. The intent is for the app to bring people together, making it easier for golfers to find and organise their games, meet new players, and navigate unfamiliar courses.

In the following pages, background research covers some of the reading and test code which has been carried out to find out what is possible in the limited amount of time for this project. The core and additional planned features are then covered briefly, followed by a discussion about the intended site organisation and UI design.

The data intended to be stored and approach to security and privacy are considered, and finally the implementation plan is presented.

Background Research

During the initial design of this project, a lot of reading was carried out alongside coding some examples to see what was possible and how it could be implemented.

The inspiration on how the website should look and feel (albeit very much cutdown due to time limitations) came from looking at similar websites. Meetup is an event organisation website [2], it has a lot more features than this project is looking to incorporate. However it has some notable features such as a focused and simple colour scheme. They use a calendar to list events and the user must be authenticated via a log in or sign up popup window. WhatsApp [3] provide a nice user interface for users to message in real time and display groups / chats to the side and messages in the main window with message delivery times and history.

For this design, a calendar or date input will be required. This could be done using HTML's input type 'date' [4]. Although not the ideal (a permanently displayed calendar), it should suffice. If time allows, other methods could be researched or one could be build from scratch using HTML, CSS and JS. Figure 1 (below) shows how this would be displayed before and after clicking the input.

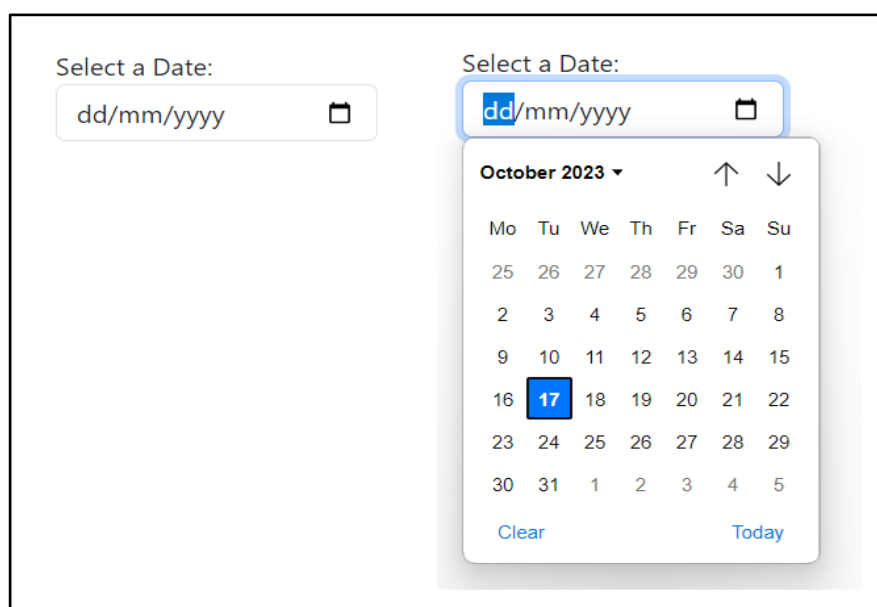


Figure 1. HTML date input, before and after clicking (Taison, 2023)

It is the intention that some of the information (such as log in, sign up, update handicap, and manage subscriptions) are to be displayed to the user by means of a pop up window. It was found that this would be possible using bootstrap modals [5] (which is more convenient to use than if built from scratch with HTML, CSS and JS). Figure 2 (below) shows what has been created as a test. Each modals content is dynamically created via a JS function. Although dynamically creating the content for each modal has been shown to work, dynamically passing in the URL routes required (for the buttons) within the modal still need to be worked out.

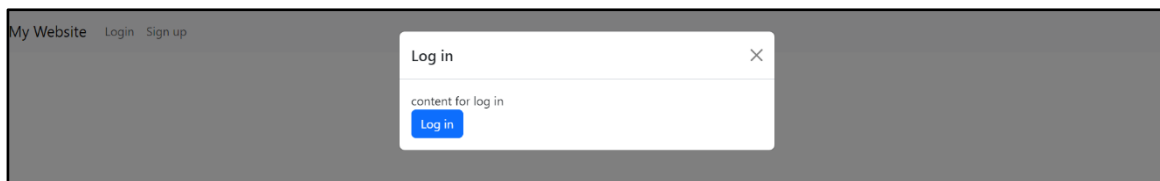


Figure 2. Bootstrap modal test (Taison, 2023)

Flask provides a way to provide feedback to users via message flashing [6]. These can then be combined with Bootstrap alert [7] to further enhance the user experience. Figure 3 (below) demonstrates Flask flash messages being used as a Bootstrap alert message - created as part of a demonstration test.

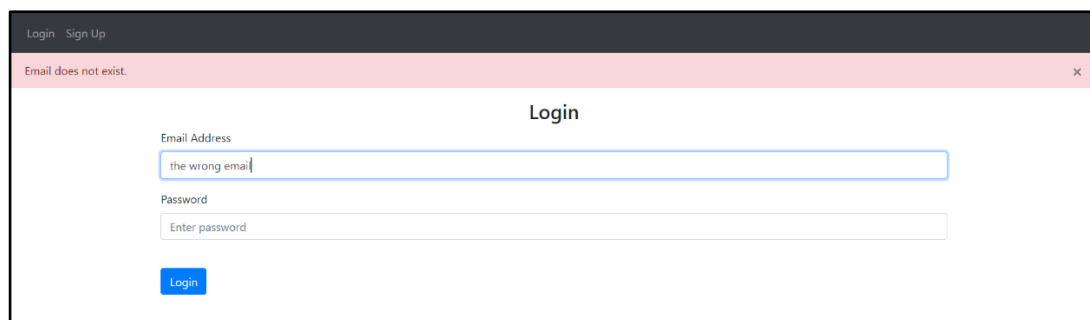


Figure 3. Flask flash message used as Bootstrap alert message (Taison, 2023)

Some pages should only allow access to authenticated users. This can be achieved using flask_login loginManager which uses sessions for authentication [8]. This allows views to be restricted using the decorator '@login_required'. Login manager can be initialised to tell flask where the user should be redirected to should they not be logged in. The logged in user can then be accessed on every template using the 'current_user' proxy.

Data can be stored in databases using SQLite3 (a database storage engine). SQLite3 performs CRUD operations (Create, Read, Update, and Delete), although this is a viable option to use, a better alternative might be to use SQLAlchemy which internally uses the DB-API driver module for SQLite3. Whilst SQLAlchemy may introduce some performance overhead, it significantly makes life easier for development, especially with shorter development timeframes. SQLAlchemy provides ORMs (Object-relational-mappers) which enables work with the database using Python objects and classes [9]. Flask also provides an extension which makes SQLAlchemy easier to use [10].

Data can be sent and received using flask request. The request object remembers the matched endpoint and view arguments [11], [12].

SocketIO is a cross-browser Javascript library that abstracts the client application from the actual transport protocol [13]. Flask-SocketIO gives Flask applications access to low latency bi-directional communications between the client and server [14]. This should allow real time messaging, however a lot more research is required in this area to understand how to set this up. It is an area of interest for the project.

Although push notifications would be ideal for this project, setting them up appears to be overly complex and involves service workers [15]. For the purposes of this project, Bootstrap notification badges will most likely serve the requirement [16].

Core Features

Below is a list of the intended features aimed to be implemented in the projects website. Each is followed by a short reason why each is included.

- **Readable text** – Text should be in a readable font and easy to scan.
- **Logo** – For brand recognition and for visitors to be able to click from any page and return to the home page for easy navigation.
- **Strong calls-to-action** – To help guide visitors what to do next.

- **Simple, but contrasting colour scheme** – To assist readability of website content and make elements stand out, ideally in a 60-30-10 split (dominant-secondary-accent (CTA)) [17].
- **Clearly labelled links** – To aid navigation and help direct users.
- **User accounts** – Users should only be able to access the public pages (base, log in, and sign up) unless authenticated.
- **Use of templates** – For reusability and ease of development and maintenance.
- **Navigation bar** – To provide easy access to common pages. The navigation bar should change depending on the users authenticated status.
- **Footer** – To contain copyright information (if time to implement, should also include links to a sitemap and about us page (these features will be additional)).
- **Pop up windows** – Ideally using Bootstrap modals. These will be called from JS functions and used for common and smaller size forms (log in, sign up, manage subscriptions, update handicap).
- **Option to change subscriptions** – The subscriptions relate only to the club events shown to the user when they select the ‘Subscribed clubs’ option on the Find a game page. Though future expansion of the site might make use of this option to email or otherwise notify users of new event listings relating to their subscriptions.
- **Option to change handicap** – A users handicap will change regularly, they should be able to update it easily. This can be used as an optional parameter when creating events (in so allowing a minimum or maximum handicap for admittance to the event).
- **Use of a database** – To store information regarding each user along with golf club information, events listed, and chat messages.
- **Hashed passwords** – Passwords should never be stored as a raw string.
- **ISO formatting** – Event dates and message timestamps should be stored in ISO format to avoid confusion and errors processing dates.
- **Appropriate events listed** – Events listed on the ‘Find a game’ page should be appropriate to the users selections (date, subscribed clubs or selected club(s)) and only show those which the user can participate in if there is a

handicap restriction. Events should only be listed which are 'open' – i.e. the event time and date have not passed and they are not at full capacity.

- **The ability to create an event / game** – Users should have the option to create an event / game if they so choose. Planned date, time and a specific club are mandatory fields.
- **The ability to add a new club** – The users should be able to add a new club which is not listed. This should require admin approval before it is to be listed.
- **Means to approve new clubs** – Admin should be able to approve new clubs. Users without admin status should not be able to access this.
- **Real time messaging** – Users should be able to message other participants of a chat in real time.
- **Option to leave chat** – Users should have the option to leave a chat / event.
- **Chat history** – Past messages should still be able to be viewed (so long as the user is still a member of the event / chat).
- **Delete chat/event** – Once all participants of an event have left (deemed inactive), the event and chat should be deleted (possibly via admin).
- **Link to club website** – There should be a link to the relevant club in a chat so users have easy access to book a tee time.
- **Ability to sign out** – Users should be able to sign out from their account and in doing so be redirected to the base home page.

Additional Features

In addition to the core features set out, the following will hopefully be added as time allows. Note these will be implemented only after the core features have been incorporated.

- **404 page** – There should be a 404 page to assist users to find the page they were looking for if it cannot be found.
- **Event creator can close access** – The creator of an event should be able to close the event to new participants.
- **Notification badges** – Badges should be used to show how many unread messages a user has for a particular chat.

- **Site map** – Guides visitors to where they want to go for easy navigation. Accessible from a link in the footer section.
- **About us page** – To give users additional information about the website. Accessible from a link in the footer section.
- **HTTPS** – Secure transfer protocol should be used if possible for added security.
- **End-To-End** encryption – To further improve security.

Site Organisation

Figure 4 (below) shows the site organisation tree. Following the connections in the diagram shows how a typical user may navigate the site. A templating system will be used to implement the site. 'Base' will be the default home page, 'Home' extends base for the authenticated user and has some additional functionality (notably an updated navigation bar). The other pages all extend 'Home'.

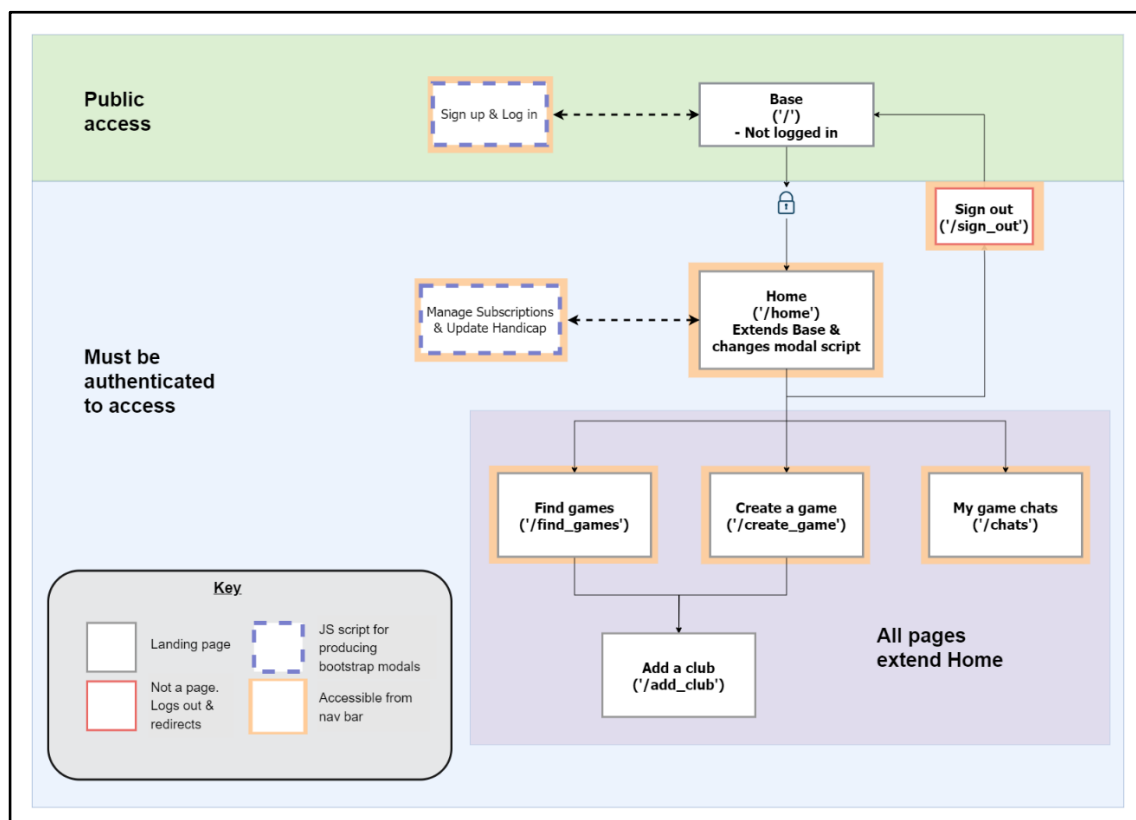


Figure 4. Organisation tree (Taison, 2023)

User Interface

For clarity, only a selection of UI diagrams are shown here. The complete set of UI diagrams can be seen in Appendix A. They have been designed to hopefully be self explanatory and need little commentary. Below some of the major UI pages have been discussed.

Figure 5 (below), shows the (base) 'home' screen for public access. This is where unauthenticated users will taken to when visiting the site. Bootstrap modals will be used to display the log in and sign up forms via a JS function.

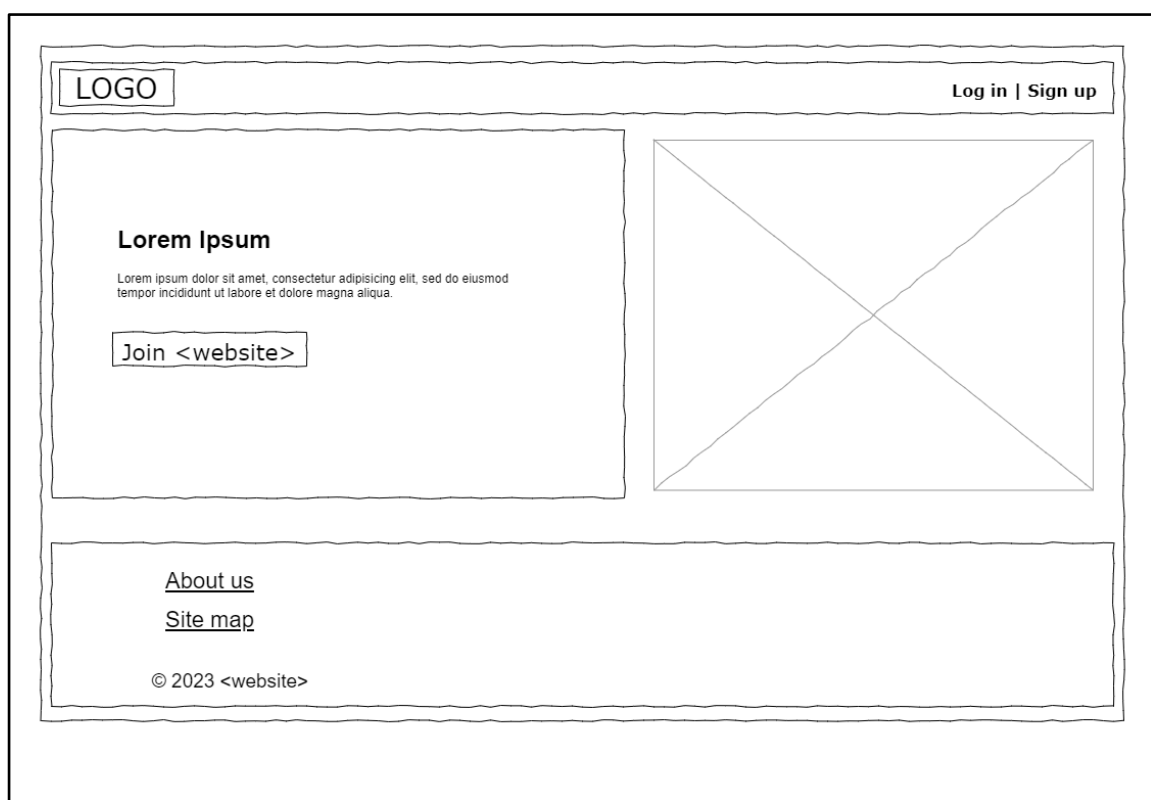


Figure 5. Base - public home page (Taison, 2023)

Once the user has logged in, they will be taken to the authenticated 'home' page. If the user at any time selects either 'manage subscriptions' or 'update handicap' from the navigation bar, the JS function (updated in the template when the user logged in) is once again called. Figure 6 (below), shows how the 'manage subscriptions' might be displayed.

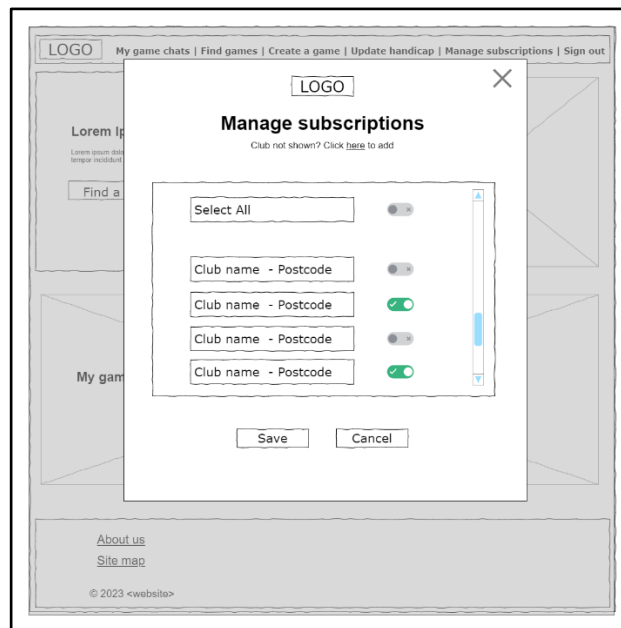


Figure 6. Manage subscriptions - modal (Taison, 2023)

The 'Find a game' page lists open events based on the users selections. If the event has a name (optional), then this will be displayed in the listings. Ideally (and if time allows), the date input will be a permanently displayed calendar. However it is assumed to most likely be the HTML input type 'date' which is an input box similar to text until the user clicks on the calendar icon. Figure 7 (below), shows the 'Find a game' page.

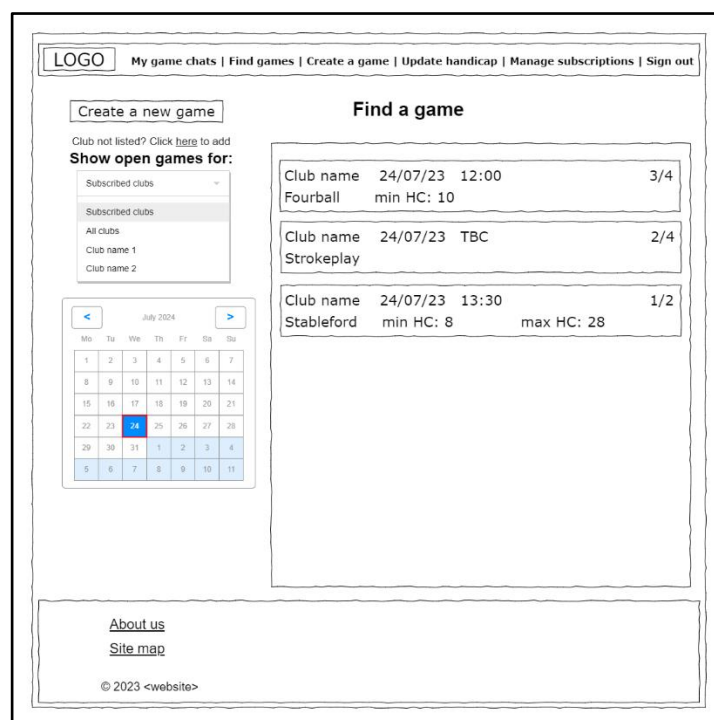


Figure 7. Find a game page (Taison, 2023)

Finally, Figure 8 (below), shows the 'My game chats' page. Here, the event chats which the user is a participant of are displayed to the side. When a user clicks on one of the event chats, the messages are displayed. The club name at the top of the messaging window should link directly to the clubs website so tee times can be easily booked. A checkbox is also provided so all participants can see if the booking is in place.



Figure 8. My game chats page (Taison, 2023)

Figure 9 (below), shows the initial colour pallet idea. The intention is to apply it in a 60-30-10 ratio [17].

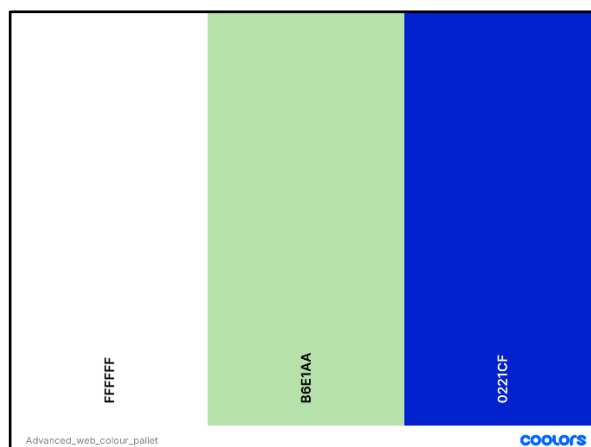


Figure 9. Initial colour pallet (Taison, 2023)

Database Design

Figure 10 (below), shows the ER diagram for the intended database implementation. Data types are shown based on SQLite data types. Not null constraints have not been shown as these will be likely to change as the design progresses. Note a composite unique constraint is also shown within the user_subscriptions table, this is not an entity attribute. The is_admin attribute of the user should be protected from modification by normal users. Ideally chat messages should be encrypted, though that is unlikely for this particular project due to time restrictions.

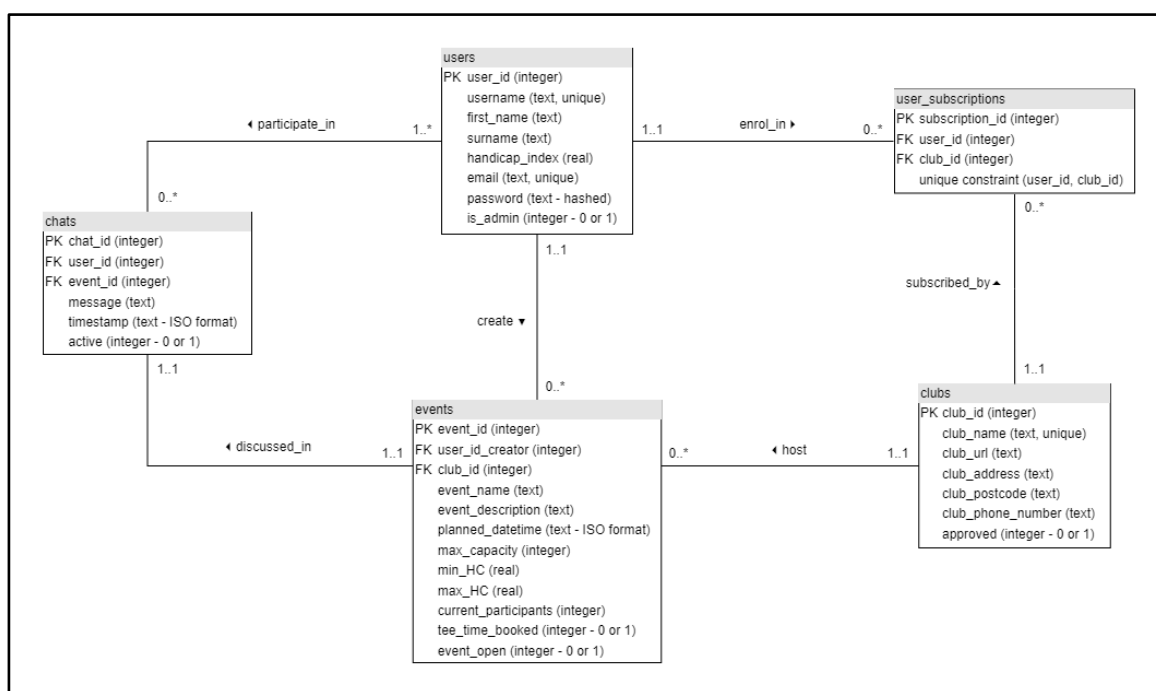


Figure 10. Database ER diagram (Taison, 2023)

Implementation Plan

The project goal is ideally to implement all of the core features and most of the additional features. Although there is a lot to get done and still some research required for some aspects (such as SocketIO).

At the time of writing, there is basically 6 weeks until the final deadline. Below is a rough outline of the intended completion timeline:

By end of:

- 29/10/23 – The base page template should be mostly defined, ideally at this point the log in and sign up should be mostly functional (although a database may not have been implemented at this stage).
- 05/11/23 – Database should be fully implemented, manage subscriptions and update handicap should now also be functional.
- 12/11/23 – Web sockets should be setup and chat should be functional
- 19/11/23 – Complete outstanding pages (Find a game, Create a game, Add a club)
- 26/11/23 – Should be pretty much complete by this stage - testing.
- 30/11/23 – Final testing and checking details

References

- [1] R. Furlong. “50 Greatest Golf Quotes of All Time.” Bleacher Report. <https://bleacherreport.com/articles/540759-50-greatest-golf-quotes-of-all-time> (accessed Oct. 14, 2023).
- [2] “Join Meetup.” Meetup. <https://www.meetup.com/> (accessed Oct. 10, 2023).
- [3] “Message Privately.” WhatsApp. <https://www.whatsapp.com/> (accessed Oct. 10, 2023).
- [4] MDN Contributors. “<input type=“date”>.” Mozilla. <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/date> (accessed Oct. 12, 2023).
- [5] “Modal.” Bootstrap. <https://getbootstrap.com/docs/4.0/components/modal/#:~:text=Use%20Bootstrap's%20JavaScript%20modal%20plugin,notifications%2C%20or%20completely%20custom%20content.> (accessed Oct. 15, 2023).
- [6] “Message Flashing.” Flask. <https://flask.palletsprojects.com/en/3.0.x/patterns/flashing/> (accessed Oct. 02, 2023).
- [7] V. Kristof. “Flask Flash Messages as Bootstrap Alert.” Coderwall. <https://coderwall.com/p/ijrrpa/flask-flash-messages-as-bootstrap-alert> (accessed Oct. 02, 2023).
- [8] M. Countryman. “Flask-Login.” Read the Docs. <https://flask-login.readthedocs.io/en/latest/> (accessed Oct. 04, 2023).
- [9] M. Lathkar, “High-Performance Web Apps with FastAPI: The Asynchronous Web Framework Based on Modern Python.” Apress, ISBN-13(pbk): 978-1-4842-9177-1, ISBN-13 (electronic): 978-1-4842-9178-8, 2023, p.156. [Online]. Available: <https://doi.org/10.1007/978-1-4842-9178-8>.
- [10] “Flask SQLAlchemy.” Pallets Projects. <https://flask-sqlalchemy.palletsprojects.com/en/3.1.x/> (accessed Oct. 07, 2023).
- [11] “flask.Request.” Read the Docs. <https://tedboy.github.io/flask/generated/generated/flask.Request.html> (accessed Oct. 07, 2023).
- [12] “The Request Context.” Pallets Projects.

- <https://flask.palletsprojects.com/en/3.0.x/reqcontext/>
(accessed Oct. 07, 2023).
- [13] S. Alla. “*Building Your First Chat Application Using Flask in 7 Minutes.*” Codeburst.
<https://codeburst.io/building-your-first-chat-application-using-flask-in-7-minutes-f98de4adfa5d>
(accessed Oct. 10, 2023).
- [14] M. Grinberg. “*Flask-SocketIO.*” Read the Docs.
<https://flask-socketio.readthedocs.io/en/latest/>
(accessed Oct. 10, 2023).
- [15] K. Basques, M. Gaunt. “*Push Notifications Overview.*” Web Dev.
<https://web.dev/articles/push-notifications-overview>
(accessed Oct. 10, 2023).
- [16] “*Badges.*” Get Bootstrap.
<https://getbootstrap.com/docs/4.0/components/badge/>
(accessed Oct. 11, 2023).
- [17] J. Lopez. “*The 60-30-10 Rule: A Foolproof Way to Choose Colours for Your UI Design.*” UX Planet.
<https://uxplanet.org/the-60-30-10-rule-a-foolproof-way-to-choose-colors-for-your-ui-design-d15625e56d25>
(accessed Oct. 17, 2023).

[Appendix](#)

Appendix A – UI Wireframes

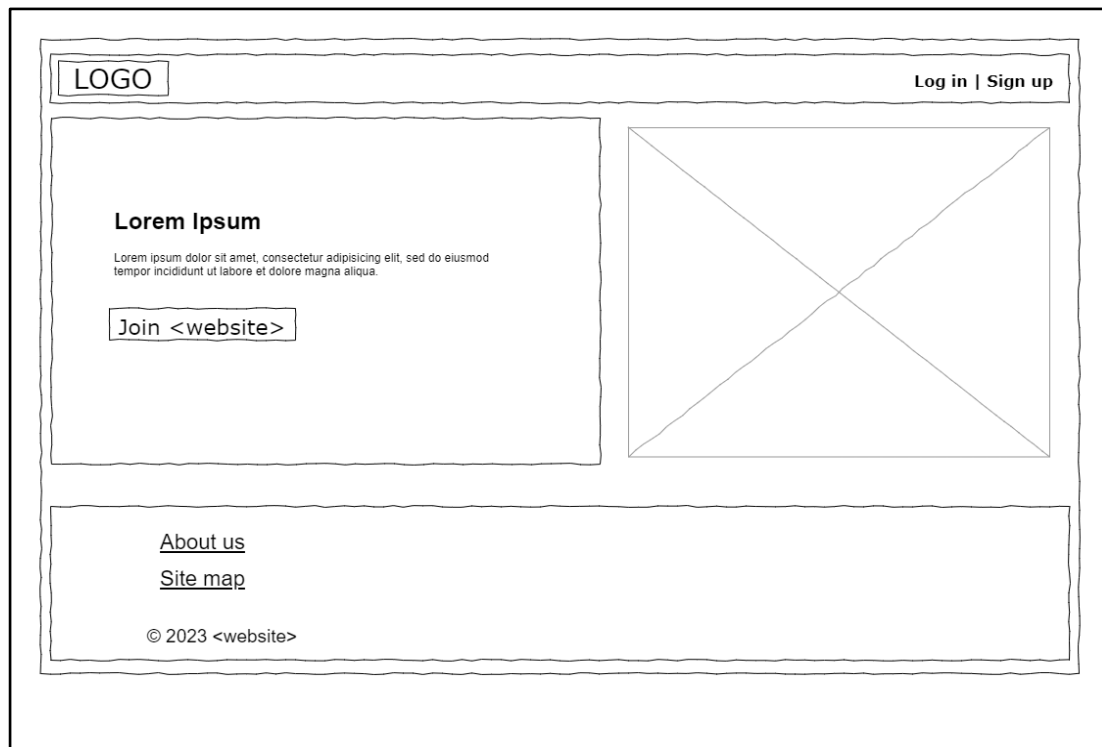


Figure A-1. Base - public home page (Taison, 2023)

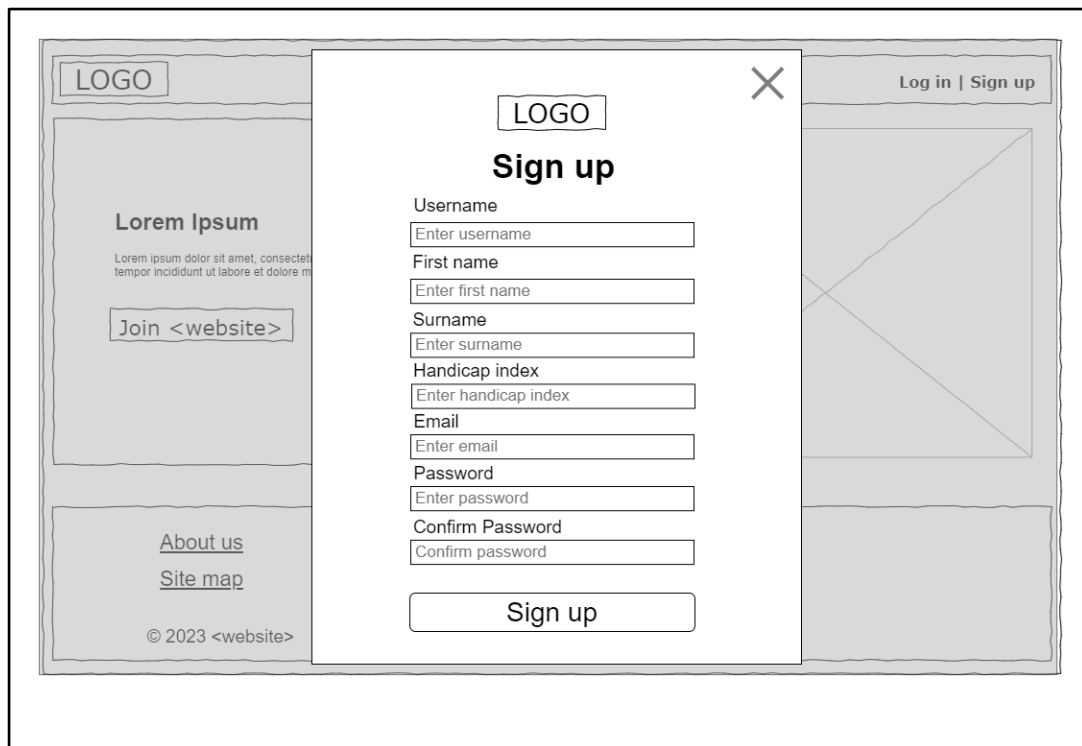


Figure A-2. Sign up – Bootstrap modal (Taison, 2023)

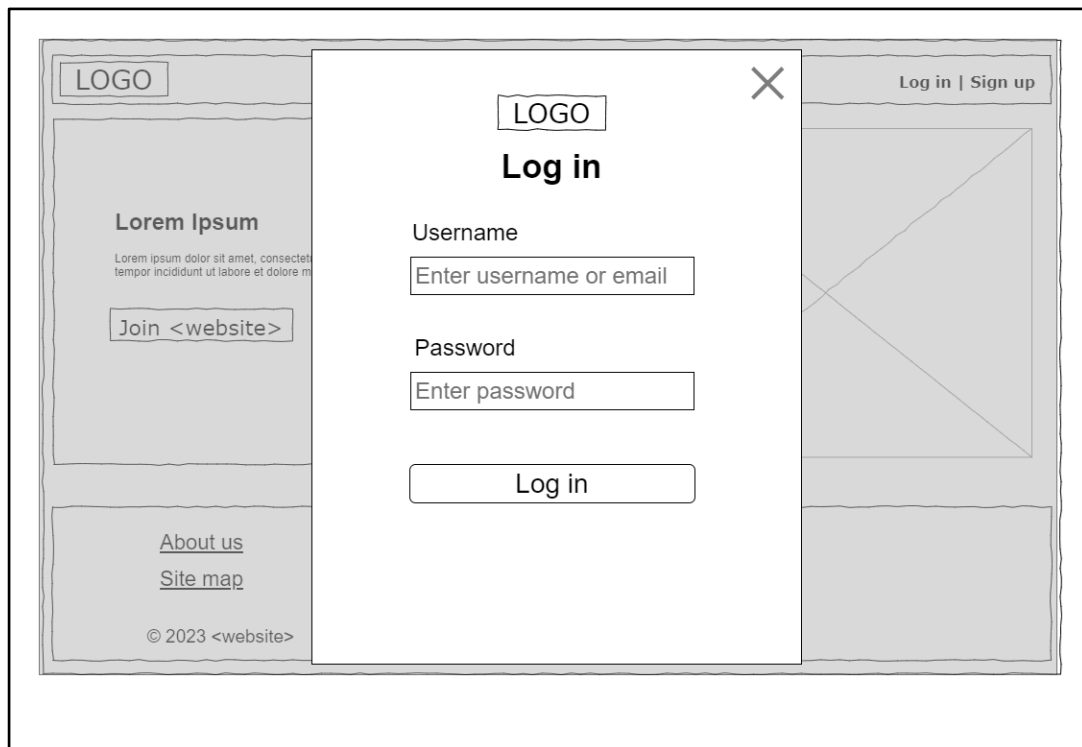


Figure A-3. Log in – Bootstrap modal (Taison, 2023)

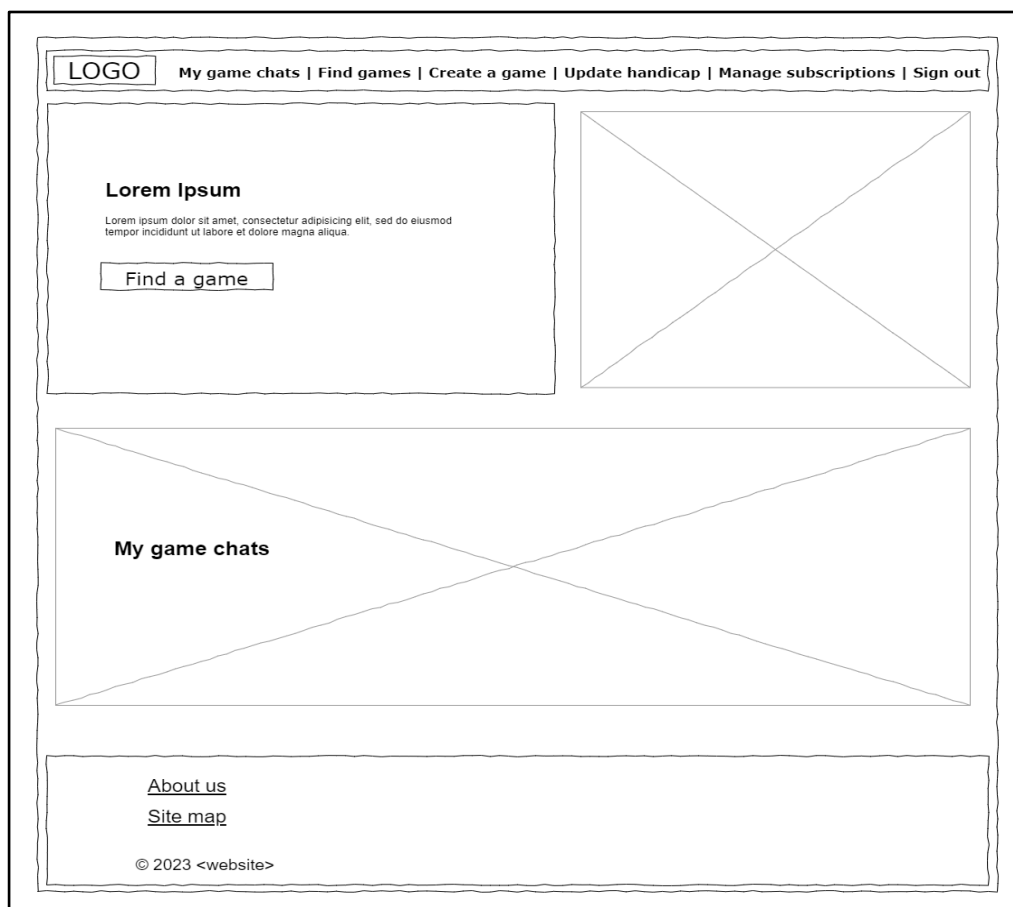


Figure A-4. Home - authenticated (Taison, 2023)

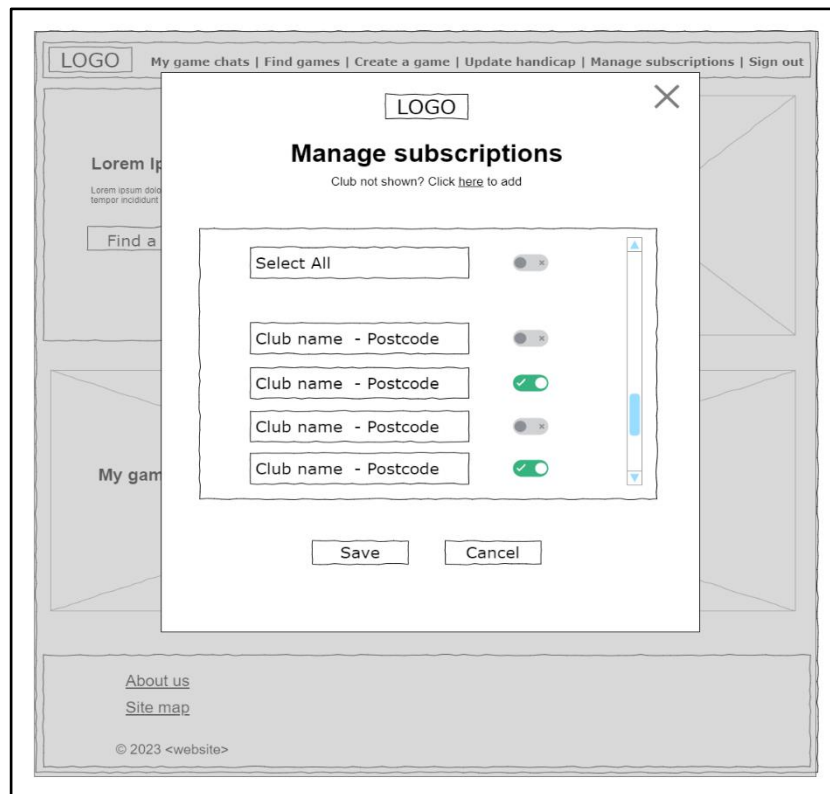


Figure A-5. Manage subscriptions – Bootstrap modal (Taison, 2023)

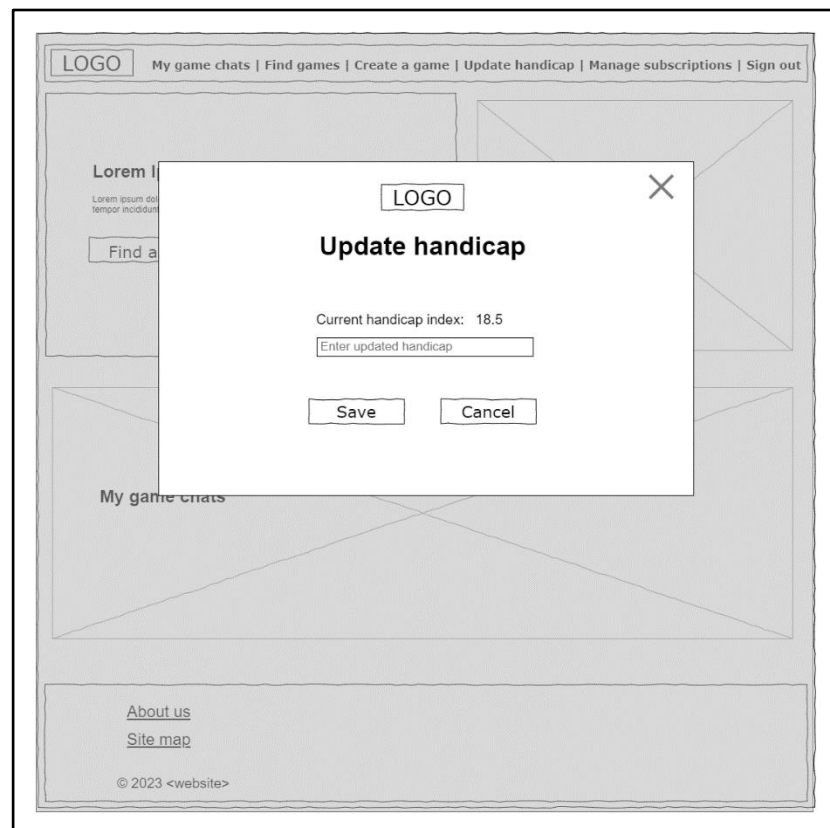


Figure A-6. Update handicap – Bootstrap modal (Taison, 2023)

LOGO

[My game chats](#) |
 [Find games](#) |
 [Create a game](#) |
 [Update handicap](#) |
 [Manage subscriptions](#) |
 [Sign out](#)

Create a new game

Club not listed? [Click here](#) to add
Show open games for:

Subscribed clubs

Subscribed clubs

All clubs

Club name 1

Club name 2

<

July 2024

>

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

Find a game

Club name

24/07/23

12:00

3/4

Fourball

min HC: 10

Club name

24/07/23

TBC

2/4

Strokeplay

Club name

24/07/23

13:30

1/2

Stableford

min HC: 8

max HC: 28

[About us](#)
[Site map](#)
 © 2023 <website>

Figure A-7. Find games - authenticated (Taison, 2023)

LOGO

[My game chats](#) |
 [Find games](#) |
 [Create a game](#) |
 [Update handicap](#) |
 [Manage subscriptions](#) |
 [Sign out](#)

Create a game

Club not listed? [Click here](#) to add
 Select club:

Club 1

 Planned time:

:

 Max group:

4

 Min handicap:

(optional)

 Max handicap:

(optional)

 Event name:

(optional)

 Event description:

(optional)

 Tee time booked? ☐

Create

[About us](#)
[Site map](#)
 © 2023 <website>

Figure A-8. Create a game - authenticated (Taison, 2023)

LOGO

My game chats | Find games | Create a game | Update handicap | Manage subscriptions | Sign out

Add a club

Note this will require admin approval

Club name:

Enter club name

Club website:

Enter clubs home page URL in full

Club address:

Enter clubs address

Club postcode:

Enter clubs postcode

Club phone:

Enter clubs phone number

Send for approval

[About us](#)

[Site map](#)

© 2023 <website>

Figure A-9. Add a club - authenticated (Taison, 2023)

My game chats Find games Create a game Update handicap Manage subscriptions Sign out	
<h2 style="margin: 0;">My game chats</h2>	
<p>Show by:</p> <p>Club: All ▾ Date: All ▾</p> <hr/> <div style="background-color: #f2f2f2; padding: 5px; margin-bottom: 5px;"> Event name, Club name 26/07/23 12:00 3/4 Fourball min HC: 10 </div> <div style="padding: 5px; margin-bottom: 5px;"> Event name, Club name 28/07/23 TBC 2/4 Strokeplay 2 </div> <div style="padding: 5px;"> Event name, Club name 07/08/23 13:30 1/2 Stableford min HC: 8 max HC: 28 </div>	<div style="text-align: right; padding-right: 20px;"> Event name, Club name Date: 26/07/23 Time: 12:00 </div> <p>Tee time booked? <input type="checkbox"/></p> <hr/> <div style="background-color: #f2f2f2; padding: 10px; margin-bottom: 10px;"> <p>Event 1</p> <p>Lorem</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <i>Suspendisse varius et ligula ut sollicitudin.</i> 09:31 </div> </div> <div style="background-color: #f2f2f2; padding: 10px; margin-bottom: 10px;"> <p>Event 2</p> <p>Romana</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <i>Aenean viverra aliquet eros</i> 09:22 </div> <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <i>Aenean venenatis hendrerit felis, sit amet ornare odio vehicula id.</i> 10:01 </div> </div> <div style="background-color: #f2f2f2; padding: 10px; margin-bottom: 10px;"> <p>Event 3</p> <p>Romana</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <i>Nulla faucibus a nisi ac convallis.</i> 10:15 </div> </div> <div style="background-color: #f2f2f2; padding: 10px; margin-bottom: 10px;"> <p>Event 4</p> <p>Lorem</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <i>Dolor sit amet</i> 10:30 </div> <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <i>Neque porro quisquam est qui</i> 10:32 </div> </div> <div style="background-color: #f2f2f2; padding: 10px;"> <p>Type a message</p> </div>
<div style="display: flex; justify-content: space-between; align-items: center;"> <div> About us Site map </div> <div>© 2023 <website></div> </div>	

Figure A-10. My game chats - authenticated (Taison, 2023)