Iyengar Yoga Class Generator

Ventsislav Antov

School of Computing Science

Sir Alwyn Williams Building

University of Glasgow

G12 8RZ

A dissertation presented in part fulfillment of the requirements of the Degree of Master of Science at the University of Glasgow

14 September 2020

Abstract

The Iyengar Yoga Class Generator is a webapp designed to auto-generate yoga classes consisting of a number of poses, all based on a combination of user preferences and yoga class structure rules. A free webapp with this exact functionality for Iyengar yoga specifically is a unique supplementary product for people interested in Iyengar yoga. The aim of this project was to develop a software that is simple, intuitive and could contribute towards increasing the non-practitioners' interest in yoga. The final result was a working webapp developed using the ReactJS framework adhering to the aims mentioned above.

This paper is focused on the analysis, design, development, deployment and testing of the software behind the webapp. It also explores the practicality of it in the world of yoga especially when it comes to the Iyengar type of yoga. The code itself is actually consisting of two separate apps and will be made available for the public audience at https://github.com/VentsislavAntov/YogaClassGenerator_PGAPI and https://github.com/VentsislavAntov/IyengarYogaClassGenerator. As the software was deployed through Heroku, some of the code was adapted which means that running it locally does not work anymore.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic form.

Name: Ventsislav Antov Signature: 

Acknowledgements

I would like to thank my supervisor, Dr. Ornela Dardha, for her guidance and overall support throughout the entirety of this project as well as all the individuals who participated in the survey for this project.

Contents

Chapter 1 Introduction 1

1.1 Aim and Objectives 1

1.2 Report Structure 1

Chapter 2 Analysis 2

2.1 Background 2

2.2 Requirements 3

2.3 Analysis 4

2.4 Features and Justifications 5

Chapter 3 Design and Implementation 7

3.1 System Architecture 7

3.2 Implementation and Technologies 8

3.2.1 Program Logic 10

3.2.2 Visuals 10

3.3 Deployment 11

Chapter 4 Testing and Evaluation 13

4.1 Software Testing 13

4.2 Evaluation Strategy 14

4.3 Results 14

4.3.1 Manual Test Results 14

4.3.2 Task-Based Results 15

4.3.3 System Usability Scale Results 16

4.3.4 Summary 18

Chapter 5 Conclusion 19

5.1 Achievements 19

5.2 What to improve 19

5.3 Future work 19

Chapter 6 References 21

Appendix A User Stories 1

Appendix B Pivot Animator 3

Appendix C Additional Sources of Guidance for Animations and Pose Descriptions 3

Appendix D **pgAdmin CREATE TABLE** **3**

Appendix E **pgAdmin Database** **4**

Appendix F **Google Form** **4**

Appendix G **Ethics Checklist** **5**

Chapter 1 Introduction

1.1 Aim and Objectives

The Iyengar Yoga Class generator project aims to develop a webapp with which users can create their own Iyengar Yoga classes and use them for their workouts. If the user is a yoga practitioner, the webapp could serve as a supplementary product for the user's yoga practicing in order to generate new combinations and to filter exercises in a custom way. If the user is not a yoga practitioner, the yoga webapp could also serve as a tool to promote this form of exercise as research suggests non-practitioners experience an exclusivity entry barrier to yoga (Marlynn Wei, 2016). The webapp is aimed to be simple to use and to not require prior knowledge or experience with yoga. An accompanying music player was incorporated as a secondary objective. It allows the users to play music during the classes. Given the lack of this exact product in the marketplace and the exclusivity problem for beginners, the online and anonymous Iyengar Yoga Class generator could provide real value to the world of yoga.

1.2 Report Structure

The report is split into chapters, each explaining different parts of the project, its components, decisions and background where applicable.

Chapter 2 will focus on the background analysis, the discussion of the requirements gathering, the subsequent analysis and the presentation of features chosen for the project.

Chapter 3 will explore the design and implementation for the webapp. The chapter will look into the system architecture design, the implementation of the webapp including the use of a secondary API application and the deployment of the two apps.

Chapter 4 will cover testing and evaluation, starting with the testing methodology. Afterwards, the evaluation strategy and the results will be discussed.

Chapter 5 will conclude the report and will discuss the achievements, the points for improvement and the potential future work related to the project.

Chapter 2 Analysis

2.1 Background

Yoga as a practice has been around for thousands of years with its earliest known origins dated to India about 5,000 years ago (Feuerstein, 2006). Recently, its popularity has been skyrocketing with practitioners in the US alone increasing from 20 million to 36 million in the period between 2012 to 2016, whilst 90% of the overall population is aware of what yoga is all about (Yoga Alliance, 2016). The number of practitioners on a worldwide scale could be up to 300 million according to Montigny (2020) showing that it already has a significant presence.

Yoga is a great practice to improve overall well-being. In general, it is an excellent substitute of exercise for almost all health-related aspects regarding both the healthy population and those with health issues (Ross and Thomas, 2010). There is evidence that yoga can contribute towards stress, anxiety and even depression reduction (Pascoe and Bauer, 2015). Furthermore, yoga spills over into other benefits as 75% of yoga practitioners also do other sports and most live sustainably (Marlynn Wei, 2016). The hidden emotional, spiritual and intellectual benefits of yoga have not been recognized properly as explained by the founder of Iyengar yoga himself (Iyengar, 2019) which is arguably even a greater benefit than health wellbeing.

Iyengar yoga specifically focuses on precision, alignment and the use of props such as blocks or blankets while performing some of the postures (also called asanas). The props help people who find some of the asanas difficult to perform them and experience the full benefits of yoga (Goldberg, 2014) which makes it suitable for those with injuries or certain medical conditions as well as for beginners (Beirne, 2014). Iyengar yoga as a specific type of yoga is very commonly applied as a lower back pain remedy and evidence strongly suggests that it works successfully for that purpose (Williams et al., 2005). A different study confirms Iyengar yoga has significant benefits to the cardiac system due to its relaxing nature (Khattab et al., 2007). In addition, it helps with knee osteoarthritis treatments and is especially welcoming for patients who are obese and/or over the age of 50 (Kolasinski et al., 2005). These are just some of the numerous proven benefits of this style of yoga.

A major reason behind why people do not practice yoga, however, is because of the idea of exclusivity. Some people believe yoga is only for those who are already spiritual, flexible and athletic (Marlynn Wei, 2016) preventing those who are not yet yoga practitioners from joining in. In a study for fitness mobile apps it was revealed that when users can customize their own personal workouts, they felt in control of their fitness style, had more autonomy and determination. It was suggested that users felt part of a community of fitness culture (Molina and Sundar, 2018). This appears to be a solution to the problem of exclusivity suggested earlier, but applied to the sphere of fitness, rather than yoga.

The current market for Iyengar yoga webapp is relatively small. One of the best apps that were found during the research for this paper is called Tummee. The webapp allows users to filter poses and/or use already prepared sequences given certain filtering. The problem is that the webapp is paid, requires certain pose

knowledge from the way the sequences are presented and is not fully randomized, but rather offers a collection of prepared sequences. There are a lot of features present, but arguably this can make it seem overwhelming and discourage those who are just starting with Iyengar yoga (Tummee, n.d.). There are other webapps that seem to be on the market such as Yoga Selection (Yoga Selection, 2019), but since they don't even have a free trial, this research was unable to establish what exactly they offer. They do, however, hint that they are more focused on training yoga rather than generating classes for users. There are other webapps for yoga in general, but not specific for Iyengar yoga.

2.2 Requirements

The requirements for the project were gathered by a combination of discussions with the project supervisor, Dr. Ornela Dardha, and an analysis of what the user should expect from a webapp of this nature that is not already present on the market.

Dr. Ornela Dardha provided a basic template of what an Iyengar class should consist of in terms of structuring the pose type sequencing, what poses to include in order to have Iyengar style yoga and what poses should be fixed for all classes. She suggested filtering options, so that users can build classes based on preferences for difficulty, type and length. An additional filter for props was later decided to be included. Classes were also agreed to be unique each time a form is submitted, so some element of randomization of the poses is always required. The necessary poses needed to be stored in some type of database in order to allow the extraction and facilitate the randomization functionality of the webapp. The webapp needed to be simple enough, so that the average user can use the app without any prior knowledge. The simplicity and random nature of the webapp is what will really make it unique given what is already in the market for Iyengar yoga.

The rest of the requirements were formed based on what the user should expect. The conclusions were that besides having basic information for each pose, the user should have some form of visual representation of the pose for assistance when performing it. This was decided to be in a form of animation, rather than a picture in order to illustrate the movements. An additional feature for a music player was included as a requirement to supplement each class which is to be chosen as a separate preference when submitting the rest of the form.

As the user will not be storing or adding any information or will have unique information displayed on the webapp which other users can't reproduce by having the same preferences, no login functionality was required.

Based on the above, the finalized webapp requirements have been prioritized using the MoSCoW technique as shown in Figure 1.

Must Have:

- Request a random class which will query the database to show the exercises to the user including Sanskrit name, English name, difficulty, type, length and props
- Be able to filter the class based on type, difficulty, length, props and a combination of these
- Be able to select None for each preference
- Be able to see the description of each pose
- Be able to resubmit the form to create a new random class

Should Have:

- Have a music player preference option with different music types and an option for having no music
- Have a music player displayed whenever a music type is chosen upon submission

Could Have:

- Show animations of each exercise
- Be able to control the music player (volume level and pause/continue)
- Add or remove specific exercises from the database
- Create/delete personal account
- Log in/out of account

Would Have:

- Facebook/Google log in
- Receive email notifications as alerts to exercise

Figure 1 - MoSCoW

This MoSCoW is slightly different compared to the original MoSCoW draft submitted on proposal as the concept of the project was not fully known yet. Nevertheless, this version was completely valid upon the start of the project work.

2.3 Analysis

The pool of poses itself was created by using the list of essential poses provided by Dr. Dardha and some other poses were included to create a sufficient representation of different pose types such as with props/without props or beginner /intermediate/advanced difficulty reaching the final total of 54 poses to be included.

The filtering options based on the requirements were finalized to be Props (None/No props/Props), type (None/Meditate/Balance/Stretch/Mix), difficulty (None/Beginner/Intermediate/Advanced), length (None/15 minutes/30 minutes/1 hour/1 hour 30 minutes) and music (Disabled/Classical Yoga Style/Lofi Hip Hop/Chill/Nature). It was also decided that the user can always have no preference selected if no choice was wanted. In this case, the program will not filter for that specific preference and the default length will be 30 minutes. For music, it was chosen to include styles that seem to align with the relaxing nature of yoga.

The randomization requirement will have to be achieved in the business logic of the program. The poses will be stored in an external database for the data extraction requirement and the reasoning behind that will follow in section 3.2. As the sample size of the poses will not be extremely large, the class length was designed to be up to 1 hour and 30 minutes which is a practical maximum anyways. As there are 3 different pose filtering options (props, type and difficulty), not all preferences are possible to be applied for each individual pose for each class generated as there are too many unique clusters of filtering combinations. There will be some priority given to the preferences. If possible, all preferences will be applied and a random exercise will be chosen from that pool. If the pool is empty, props will be removed from the filtering list and an exercise will be chosen from this wider pool of poses. If that doesn't work, type will be removed. If that doesn't work, difficulty will be removed which will leave no filtering at all and a random pose will be chosen from the overall pool of poses. The reasoning for this order is that there will not be that many exercises with props anyway, so props has to be removed first. Type will be next given the 4 different categories versus the 3 different categories for difficulty (less poses per category). Finally, the difficulty will be removed so that, if needed, a pose can still be extracted even if the user's preferences cannot be even partially applied. The last pool will be very unlikely to be used at all as the poses are relatively well distributed between the different categories of each type, but it's still possible and is good practice if more categories and/or preferences are added in the future, expanding the filtering possibilities and narrowing individual unique cluster sizes.

To achieve optimal simplicity, the user interface will have to be simplified as much as possible, so no yoga-heavy jargon will be used on form submission and no unnecessary complexities will be added to avoid confusion potential.

2.4 Features and Justifications

As already touched upon, the overall idea of the Iyengar Yoga Class Generator Webapp is to provide simple Iyengar style yoga classes consisting of poses filtered by user preferences. A webapp of this nature could overcome the exclusivity issue mentioned earlier observed with people reluctant to try out yoga. The second primary purpose for developing the webapp is to provide a support tool for yoga practitioners to simply have a way of generating different classes with ease. With this in mind, to aid the development of this project, user stories were written, identifying individual features that were planned to be incorporated into the project. Due to the project being carried out by an individual rather than in a team, no sprint methodology was applicable and user stories do not have points assigned to them. The user stories are illustrated in Appendix A.

Given that the main idea of the project was to be able to create classes from yoga poses, the user story "As a user I can select my preferences for the poses and submit the form" reflects that main functionality, thus justifying the highest priority.

The user story "As a user, upon submitting the form I will receive a class which will be for my desired length preference" is linked to the ability to filter based on the desired class length which is necessary given the filter options mentioned earlier. Because this is not a critical feature, the priority has been set to 2.

The user story “As a user, upon resubmitting the form, I will have a new randomized class generated for me using the new preferences” will provide the user the option to resubmit and receive new classes every time even when submitting the same preferences. This is crucial, so the priority is 1.

The user story “As a user, I can see the English and Sanskrit names, the exercise group, the type, the difficulty, the length and the props of each pose after the class is generated” is focusing on the details of each pose which is useful information to the user, necessary to differentiate among poses for inexperienced users. Therefore, this user story has the highest priority.

The user story “As a user, I can see a visual representation of the pose in the form of an animation” builds on the necessity to have an illustration of each pose to aid the user in completing the pose. This is also crucial for new yoga practitioners, so the priority is 1.

The user story “As a user, I can see the description of each pose when the form is submitted” aims to provide a functionality where descriptions are linked with each pose. Due to the fact that there are already animations aiding the users, this user story has a priority of 2.

The user story “As a user, I can request music to be played as part of the form submission and can select from multiple types” is an additional feature which is secondary to the purpose of the app. Having background music could be a nice additional feature for those users who like to listen to music while exercising. Given the secondary nature, this priority is of 2.

Finally, the user story “As a user, I can control the volume, pause and unpause the music if I had selected a music type before submitting the form” is building on top of the previous story by providing convenience to the user in terms of controlling the music player. As the overall music player is secondary for the purpose of this webapp, the priority here is again 2.

Chapter 3 Design and Implementation

3.1 System Architecture

Based on the requirements, multiple approaches were possible for developing the web application. Middleware such as Django would have been relatively easier to implement. I personally chose to use the React JavaScript framework as I was eager to learn how to use it.

React, as described on the official React website, is a convenient and efficient way to create user interfaces that are rendered in real time as the program is updated. It has a unique reusable component-based system which aligns with the concept of separation of concern. All the logic of components is integrated into JavaScript, so the state is out of the DOM. Server rendering is enabled through Node.js. Components can have their own "state" with which the rendering can change the output (React 2020). The virtual DOM enables much faster manipulations which is why React is better than normal JavaScript in that aspect (Codecademy, 2020). Finally, React is actually maintained by the tech giant Facebook (Malhotra, 2018). The drawback to choosing React is that the database has to be developed separately and can't be included as part of the same program similarly to what Django does. For the Iyengar Yoga Class Generator this meant that an API is necessary in order to extract information from the database.

Figure 2 represents the system architecture for the webapp.

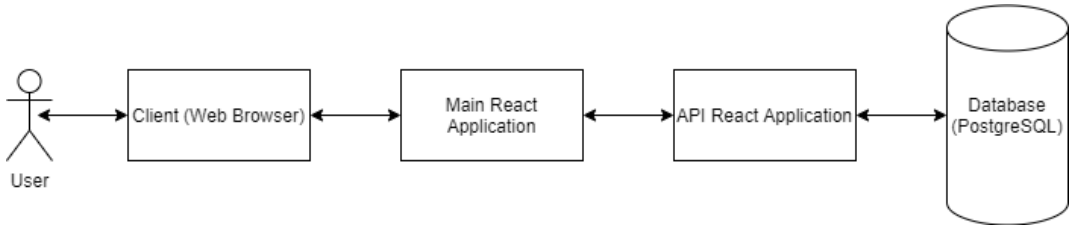


Figure 2 - System Architecture

The user goes to a web browser (client) and accesses the Main React Application. Upon submitting the form, the Main React Application fetches data from the API React Application. The API React Application uses GET and POST requests to query the database which is in PostgreSQL.

The decision for the split of Main React Application and a separate API React Application was for two reasons. During the coding process, having a main process for the application and a separate one for the API within the same overall application caused the need of two separate localhosts which hinted issues upon deployment. A solution was to split the webapp into two separate applications. Secondly, the idea of microservices is becoming an increasingly popular common practice. Having a single app for multiple services is referred as a "monolith" which can have issues especially long term when the web application becomes large-scale. Such issues could be related to dependencies, memory, computational requirements, maintenance and scalability limitation. Microservices provide an alternative with which processes are independent and communicate between each other (Dragoni et al., 2017). Furthermore, this aligns with the separation-of-

concerns principle which is one of the most important concepts of software engineering where different concerns of a problem are solved by different components of the software (Win, Piessens, Joosen and Verhanneman, 2002). Arguably, this is more suitable for larger applications, but the development of the Iyengar Yoga Class Generator was done with the intention of potential future developments and complexities.

3.2 Implementation and Technologies

Because of the significant time consumption for animating the poses, the number of poses had to be limited to a certain maximum. The choice of the specific poses to be added to the database was based on 45 mandatory class poses (IYENGAR YOGA (UK), 2018) suggested by my supervisor as well as additional poses extracted from Tummee (Tummee, n.d.). The latter group was selected with the intention of having sufficient poses for each filtering combination. Nevertheless, as the user can filter poses from three different options and more importantly, filter by all 3 options simultaneously, an enormous number of poses will be needed for complete filtering for all scenarios especially for the longer class length options. Having that in mind, a balance of 54 total poses was chosen as sufficient for the purposes of this project. In some long class length cases, the program will aim to select as many exercises from as many filters provided by the user as possible, but where that will not be possible, some of the filters will be ignored, which was the most reasonable solution given the stated limitations.

The pose animations were first developed using Pivot Animator which is a very simple stick-figure animation tool. The large time consumption was due to the fact that some animations were up to 150 frames and each pose required an animation. An example can be found in Appendix B. These were then saved as .piv files and subsequently uploaded to Giphly in order to have the animations as a gif link. Some of the animations were created with the help of YouTube videos. The YouTube channels used can be found in Appendix C. A database of Yoga poses with video examples was also used for some specific poses (Yoga Journal, 2020).

Once the pose animations were completed, the database itself was created. For this purpose, pgAdmin 4's PostgreSQL platform was used. The create table command can be found in Appendix D. Each pose has a unique ID.

Each pose's details were acquired from a combination of Tummee data (Tummee, n.d.), the YouTube videos from Appendix C, information provided by my supervisor who is a yoga practitioner and several other sources for specific pose descriptions (Yogic Way of Life, 2019) (Studio Po, 2015). Some of the poses were amended later on after a review from my supervisor for improvements. This included straightening the legs more on pose animations such as for Utthita Trikonasana and fixing several poses entirely using the book by Iyengar himself (Iyengar, 1997). Appendix E illustrates the final database with the populated information.

The structure of each class has the following rules which were advised by my supervisor. Each class starts with a seated pose which can either be Siddhasana or Svastikasana where the latter is more likely. Note that this was implemented in the program as 30:70 probabilities. After that, standing poses follow which always start from the pose Tadasana. Then come backward extension poses, seated twists, seated forward extensions, the pose Sarvangasana and the pose Savasana.

Based on sample classes provided by Dr. Dardha, the visual class representations from Appendix C and the details of the 54 poses already included, the final structure for the classes is illustrated in Figure 3.

1	Siddhasana/Swatiksana	Random Generated with higher Swatiksana odds
2	Tadasana	Constant
3	Standing Backwards Extension	Random Generated
4	Standing Forward Bend	Random Generated
5	Seated Twist	Random Generated
6	Seated Forward Bend	Random Generated
7	Seated other	Random Generated
8	Prone/Supine	Random Generated
9	Sarvangasana	Constant
10	Sarvasana	Constant

Figure 3 - Class Structure

Some sub-categories which were originally planned to be included have been combined in order to reduce the number of smaller categories and better accommodate user preferences. An example is Standing Side Bends incorporated into Standing Forward Bends. Prone and Supine are also combined because Prone only had 3 poses in the database. Class parts 3 to 8 inclusive, visible in Figure 3, will all have assigned time equal to (User Time Preference - Sum of parts 1,2,9,10)/6. This will ensure that each section has the same amount of time attributed to it.

The two apps that comprise this project were developed using the React JavaScript framework and the IDE used was IntelliJ. As touched on before, React JS was completely new for me, so about 2 weeks of steep learning were necessary to research how to work with it. A class diagram can be found in Figure 4 for the finished Main App.

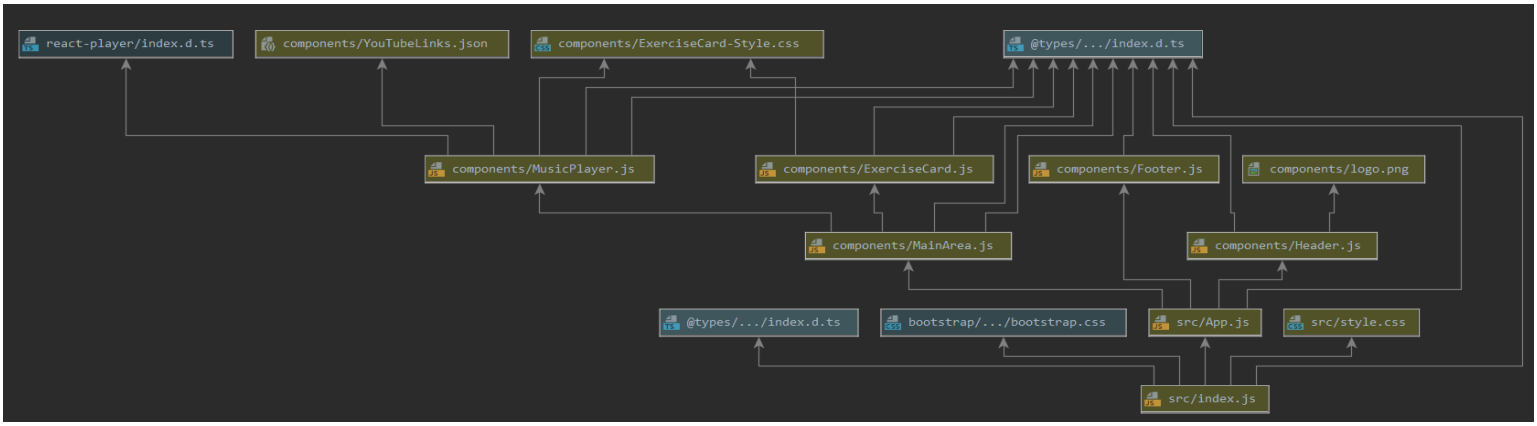


Figure 4 - Class Diagram

Having an API also means that two localhosts will always be active. For this reason, a second application is dealing separately and in isolation with the database, facilitating data extraction. One consequence of this development was that the deployment had to be done in a more complicated way. Django, for example, incorporates the deployment within itself, but in this case, Heroku will be utilized to deploy the React app instead as detailed in section 3.3.

3.2.1 Program Logic

The program uses the special handleChange function for changing the state attributes upon the user selection from the dropdown menus for each preference. The program then uses the special handleSubmit function to do the logic when the form is submitted by clicking on the "Generate Now" button. Note that the state for that submission logic is already chosen from the dropdown menu selections done by the handleChange function. All poses from the database are then fetched using the GET API from the API webapp and stored in an attribute in MainArea's state. This ensures that if the database is changed, the most updated database version is extracted when a form submission is made. State in ReactJS can take different data type equivalents compared to Java. In this case the fetched data is stored in a state attribute similar to a hash-table. The program stores the final poses for the class again in a similar attribute in state and adds every pose to the that attribute during the filtering function. This is the incredible benefit of the state's flexible functionality of ReactJS. The state data structure is automatically dealt with throughout the entirety of the project.

Before the logic for the different pose category section comes, one array is made for all exercises filtered only by the user's difficulty preference. Then another one by difficulty and type. Then another one by difficulty, type and props. The program uses these arrays which are all further filtered by the current category (the categories are parts 3-8 from Figure 3) to select a random pose. This loops until the combined time for the poses in the category reaches the specified time which, as mentioned, equals to (User Time Preference - Sum of parts 1,2,9,10)/6.

Upon selection, each pose is removed from all the arrays to avoid duplications in the future. First, an attempt is made to select an exercise from the most filtered option and if the array is empty, the program tries to select from the next, less filtered array. The program accounts for a scenario where there are no more poses for a given category without any further filters in which case a final attempt is made to select a random pose from the entire remaining pose pool. Given the maximum class length, this will never be reached, but it is good practice, allowing for future changes.

3.2.2 Visuals

The visual part of this webapp was developed within React, but using the Bootstrap framework. This covers the overall alignment of the items on the screen and because of the responsive grid system of bootstrap, this provides good responsiveness to different screen sizes for the webapp. Bootstrap was also used for the pose cards which allowed to incorporate the card flip functionality. Note that the card is a separate component within the program which utilizes the React component efficiency. Cascading Style Sheets were used as well. There is a second css file dedicated for the card component.

The webapp includes the Music Player which was incorporated using the ReactPlayer import and is contained in a separate React component. The Music Player works as if streaming directly from YouTube bringing along the buttons and navigations from YouTube including volume setter, pause/resume and the timeline navigator. The webapp stores URLs for several music categories in a JSON file and upon preference selection, a random URL is chosen from that selection category. All URLs are from Live videos at the time of creation which

means that different individual soundtracks are played on the channel at any selected time. Note that sometimes the YouTube video owners remove their videos or disable third-party playing which is essentially what this webapp does. In such case, the video will say it is unavailable. This is something that is outside of the webapp's power to control, but it is why several URLs were included in the JSON file. If it doesn't work and the form is resubmitted with the same music type, a new random URL will be selected which will likely work.

In terms of copyright, YouTube has a fair use policy which allows usage without permission from the original video owner in cases where it is for teaching/educational purpose (YouTube, n.d.) especially when it comes to situations with non-profit intentions rather than commercial (YouTube Help, 2020) which is the case with this webapp. Moreover, these guidelines are for users uploading new videos on YouTube using other YouTube content creators' material which is not happening here as this is just a third-party playing directly the same video. It is essentially streaming from YouTube rather than uploading a new video meaning there is no copyright infringement in any case, but even if, fair use is applicable. While working on the webapp, one of the content creators turned off his third-party playing which stopped playing the music on the webapp's end. This hints that for the rest of the URLs, the channel owners have intentionally turned it on for third parties.

The webapp includes a header and footer and the header has a logo which is a free-to-use image from an internet source (PNG MART, 2018).

3.3 Deployment

For the deployment of the product, an initial attempt was made with GitHub Pages. The deployment of the Main React Application was successful and was displayed properly, but there was no way to link the API React Application and the Main React Application. As a result, an alternative was necessary.

In the end, Heroku was used to deploy both apps through a pipeline which enabled both to run at the same time, to communicate correctly and removed the requirement for the end user to run the API separately. Prior to that, running the project locally required both apps to be launched using different terminal commands. Another benefit of Heroku was that each GitHub push to the remote master (origin) branch automatically redeployed the updated app. Note that Heroku causes the webapp to go to sleep if no web-traffic is registered for more than 30 minutes, so a small functionality delay upon opening the app for the first time in a while is normal (Heroku Dev Center, 2020).

In the end, the final product was enabled to be accessed via a single link (https://iyengaryogamain.herokuapp.com) which is leading directly to the Main App. Note that the drawback here is that the MainArea fetch link, GET Request link and pgAdmin database had to be amended to adapt for Heroku to work. Consequently, the webapp can no longer run locally.

The following figures illustrate the deployed webapp including the changes made after the user testing mentioned in section 5.1. The arrows are added to aid understanding. Figure 5 shows what the webapp looks like upon opening the webapp and selecting "Chill" on the music preference dropdown.

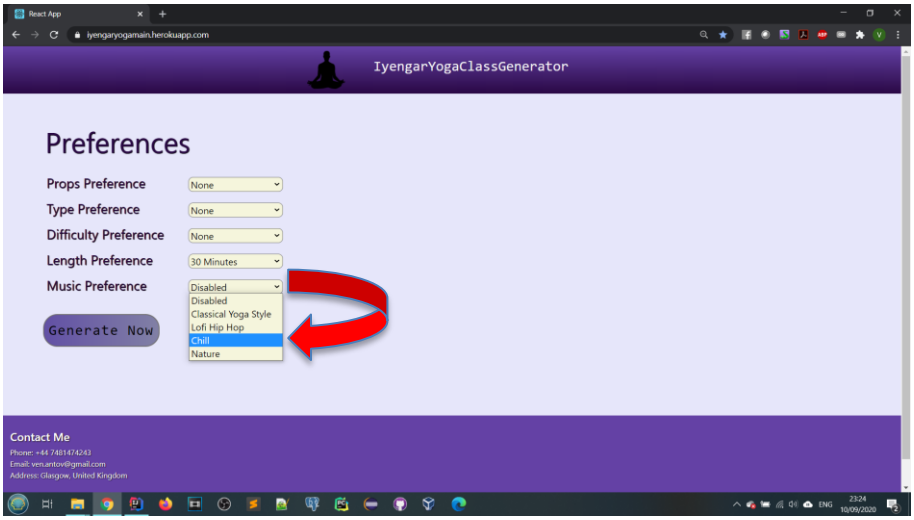


Figure 5 – The webapp when opened and a dropdown is selected

Figures 6 and 7 visualize the generated class; the latter also shows a card-flip.

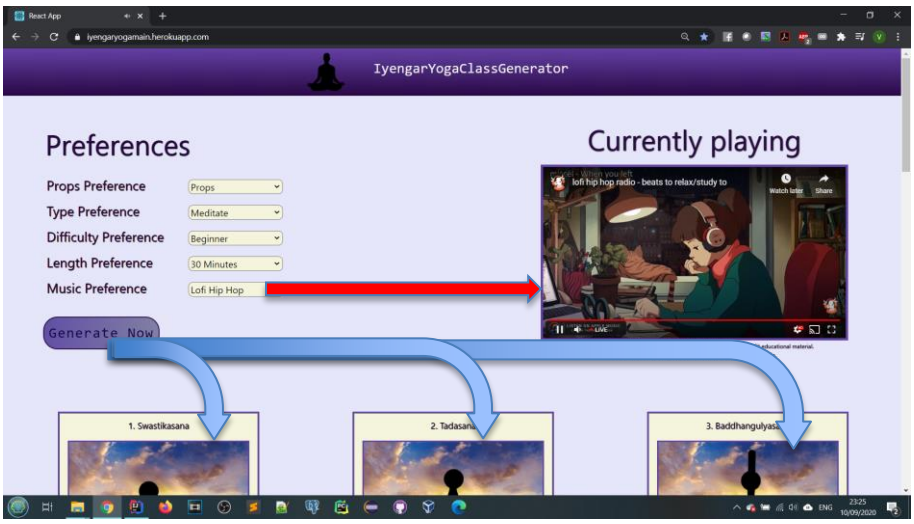


Figure 6 - The webapp when the Generate Now button is clicked.

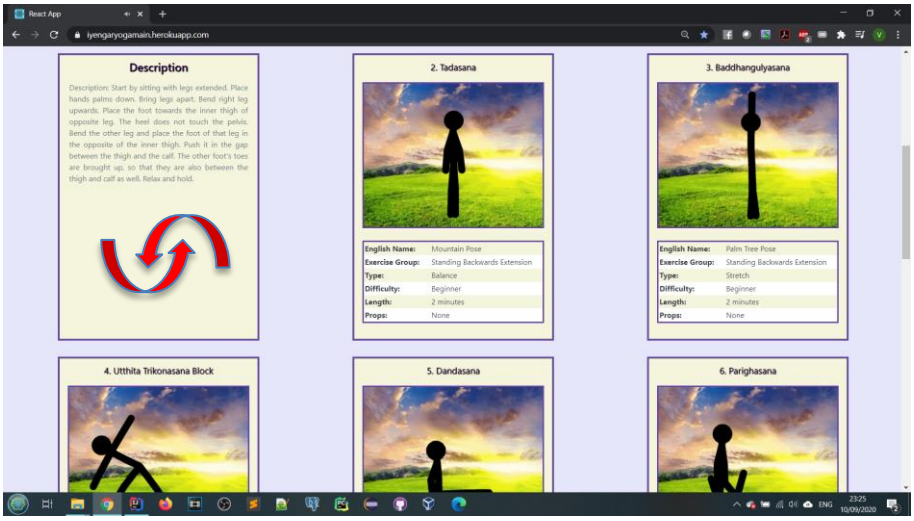


Figure 7 - The webapp scrolled down after form submission and after pointing to the first pose to flip the card to reveal the pose description

Chapter 4 Testing and Evaluation

4.1 Software Testing

Unit Testing was chosen not to be done due to the time cost, but manual tests were chosen to be conducted to examine user stories acceptance criteria from Appendix A. This will be revealed in detail in section 4.3.

User tests were also decided to be included and the platform via which the user testing was done was Google Forms. The entire questionnaire can be found in Appendix F.

Laboratory user testing for apps especially in mobile has been proven to be sufficient when reviewing user interfaces. This involves giving users tasks, completing these tasks and observing the outcomes (Kaikkonen et al., 2005). The first part of the Google Form is essentially focused on remote laboratory user testing where users are given tasks to do on the Iyengar Yoga Class Generator and asked questions regarding those tasks. All tasks and questions have been simplified as much as possible in order to avoid confusion and extract maximum insight. The tasks are focused around the webapp features and cover the entire functionality of the webapp. This is very specific to the webapp and in order to review the webapp in its entirety, some form of an overall satisfaction questionnaire will be necessary as well.

Empirical research has shown that a tool called the System Usability Scale is a very reliable way to test the overall usability of a product. The original version consists of 10 statements and they require the user to rate them from 1 to 5 (Bangor, Kortum and Miller, 2008). This is a very appropriate tool to test the system usability of the Iyengar Yoga Class Generator. Therefore, the second part of the Google Form consists of a modified SUS questionnaire where additional questions, focused on overall webapp satisfaction, have been included for increased insight. The last 5 questions are yoga specific as well and are only aimed at yoga practitioners. They have been made optional in the Google Form whereas the rest are mandatory to ensure complete data collection.

The form was sent out to an audience consisting of both yoga practitioners and non-yoga practitioners. Participants had to first consent to participating and using their data before continuing as a first and mandatory step. Note that the survey is anonymous and points out to participants that they can withdraw their participation at any time. The final sample consists of 10 completed forms which is sufficient for the purposes of this project. The sample includes enough yoga practitioners as well which makes use of the yoga-specific questions from the second part of the Google Form.

This research complies with the School Ethics Procedure. The Assessment Ethics Checklist form is applicable for this case and has been signed as shown in Appendix G where the introduction and debriefing scripts have also been included. The entire Google Form can be accessed at:

https://docs.google.com/forms/d/e/1FAIpQLSfYAl5vwMMngKAHrEGAit5gzWFkwlVYBVnPahq1p3yvNwv2uw/viewform?fbclid=IwAR2niRGxitpBb0VHpBAbGIgEQ3Kl_aDk97NN-fvFxCQeRSlp2dEjBMx8veM.

4.2 Evaluation Strategy

The evaluation of the manual tests will be done on a simple passed or not passed basis.

This will then be followed by an analysis of the survey feedback which included both averages and outliers where appropriate. Google Forms has a built-in functionality where the creator of the form can see some basic statistics of the results which was used for the analysis.

The task-based questions have been designed to either be quantitative or to reaffirm expected results from completing a given task. The evaluation here will confirm if the webapp is performing as intended.

The usage of the System Usability Scale (SUS) also allows for quantitative insight into the way users think about the product overall which is a different way of evaluating it.

Note that the evaluation is based on the collected sample of users only which is 10. For better results, future research could expand the survey to a wider audience for better statistical accuracy. There was no specific profiling of the users when given the questionnaire which could also be amended for future research based on what direction the product takes.

4.3 Results

4.3.1 Manual Test Results

Figure 8 summarizes the tests and the results. All manual tests passed indicating that user story acceptance criteria have been satisfied.

User Story	Test Made	Result
1	Attempted opening each dropdown, selecting a preference and submitting the form.	Class was generated underneath. Test passed.
2	Several different class submission attempts were made to test if combined pose length is smaller than length preference.	For 30 min preference, combined pose length was 26 min. For 60 min preference, combined pose length was 56 min. Test passed.
3	A resubmission was made with the same pose preferences multiple times.	Every time poses generally changed with the exception of the fixed ones. Test passed.
4	After a submission, all the poses were checked for details.	Each pose included all the details successfully. Test passed.
5	After a submission, all the poses were checked for having animations.	Each pose had a working animation. Test passed.
6	After a submission, all the poses were checked for having a description by moving the cursor over the pose.	Each pose revealed the description through the card flip. Test passed.

7	As part of a submission, "chill" music was selected.	Upon submission, a video player was activated playing "The Good Life Radio" live stream from YouTube. Test passed.
8	After the submission of a form with a music preference, the video player controls were tested.	The video could be successfully paused, unpaused and volume controlled. Test passed.

Figure 8 - Manual Tests

4.3.2 Task-Based Results

For this section, each question from the task-based survey part will be analyzed separately and chronologically.

For “Does the webapp generate a class consisting of multiple poses for you?” 100% of users reported class generations with multiple poses when clicking on the submit button. This indicates that the main functionality of pose generation is working including the communication with the API webapp, the deployment, the data extraction from the database and the display of those poses.

For “What is the first pose name?” the first pose was reported as overall 20% Siddhasana and 80% Swastikasana of all cases. The first pose was programmed to be either Siddhasana or Swastikasana with 30:70 probability ratio. Given that the survey size is of only 10, this is close enough to reveal that this task has worked correctly for all users and that the backend logic for the pose rules is working.

For “What is the last pose name?” all practitioners with the exception of one have reported Savasana which is what is expected. The last user has picked Utthita Hasta Padangusthasana Belt which is probably an error as the other 9 have it reported correctly.

For “What is the second pose name?” in all cases Tadasana was reported which is what is expected and again reaffirms that the pose logic on the backend is performing as intended.

The next two questions are connected with each other and for the benefit of understanding the different cases, Google Forms’ spreadsheet functionality was used to show each participant’s length preference and corresponding poses displayed for that specific length preference. Participants have picked primarily 15 min and 30 minutes and 1 user has picked 90 minutes. In all cases the ratio minutes to poses is about 3:1 which is what should show up given that the average pose length is about 3 minutes. This also indicates that the length preference part of the logic is working.

For “Can you see the stick-figure animation of the first pose?” the response was always positive indicating animations are always working using the gif URLs.

For the “What is the third pose's Difficulty and Length?” question there was a wide variety of responses which is what the aim of the question was. The third pose is always random in contrast to the first two which are constant. This proves that the randomization works.

For “What music is playing at the moment?” participants have reported all the different types which shows that the music functionality is working well for their chosen type.

For “At what % is the volume roughly set at the moment?” 70% have reported 50% volume which is what they should have been seeing if they have not changed the default. The rest have probably tweaked it before reaching this question which was not instructed, but is understandable if it was too quiet or too loud at start. The fact that they can see the volume mixer means the music player functionality is working as intended.

For “Can you read the description of the 4th pose by positioning the cursor over the animation?” all participants could read the 4th pose description indicating that the pose descriptions are shown as intended successfully.

For “Is there any music played/displayed now?” after submitting a form with music preference as disabled, all participants confirmed that they do not have music playing. This confirms that the functionality for selecting music to be enabled or disabled works correctly.

For “How many poses are displayed for this class now?” practitioners reported various numbers which were often different when compared to their corresponding first class’ pose numbers even though they have the same class length preferences. This proves that the class generation is often times creating a different number of poses for the same class length due to the different pose length durations and the randomization factor.

90% of the practitioners have reported that the poses have generally changed. The remaining 10% could have focused their attention on the fixed poses when the selected length is the 15-minute minimum. In that case, proportionately, the fixed poses will have taken a larger part of the entire class, explaining the outlier.

4.3.3 System Usability Scale Results

In this section, each question will be analyzed separately and chronologically.

Figure 9 shows that the practitioners would overall like to use the system as it is averaging out at 4. This indicates overall user-friendliness and usability of the webapp.

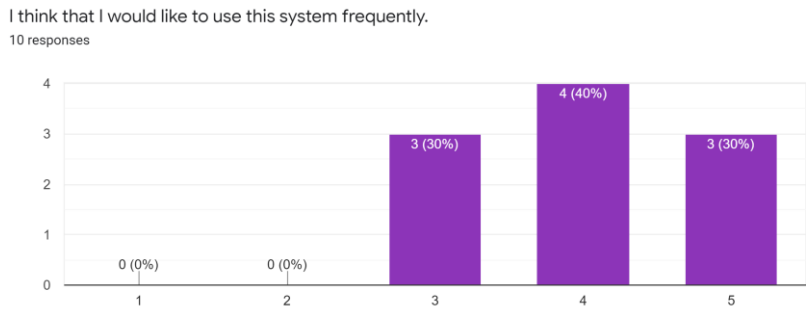


Figure 9 - System usability and friendliness question survey result

90% of the participants selected 1 as rating the system as unnecessarily complex, which evidences that simplicity was indeed achieved. The last rating was of 2. This is further supported by the fact that 90% of the users also thought the system was easy to use with the maximum rating of 5. The last rating was of 4. Lastly, 90% of the practitioners selected a score of 1 for the need of technical support to use the system whereas only 10% selected 2. These questions make sense as they are all correlated with the simplicity of the webapp which was what one of the primary goals was in the first place. The ratings for functional integration were of 4 and 5 only and the presence of system inconsistencies was almost entirely rated as 1.

80% of participants felt that a score of 5 and 20% felt that a score of 4 is appropriate for the statement that people can learn to use the system very quickly. This is of particular importance as, again, the webapp was aimed to be as simple as possible to encourage integration into yoga practicing. In addition, 80% completely disagreed that the system was cumbersome to use and 20% thought that it deserved a rating of 2 which is overall a very good result.

The system confidence had on average a rating of 4.8. This indicates that users found the system logical and predictable. On top of that an average rating of 1.1 was achieved for the need of learning things before using the system. These results support the notion that users are not expected to know or learn anything prior to using the webapp.

The system and design elegance are more spread out even though the average rating is still a very good 4.4 as seen in Figure 10. This indicates that further work can be done to make the design more appealing.

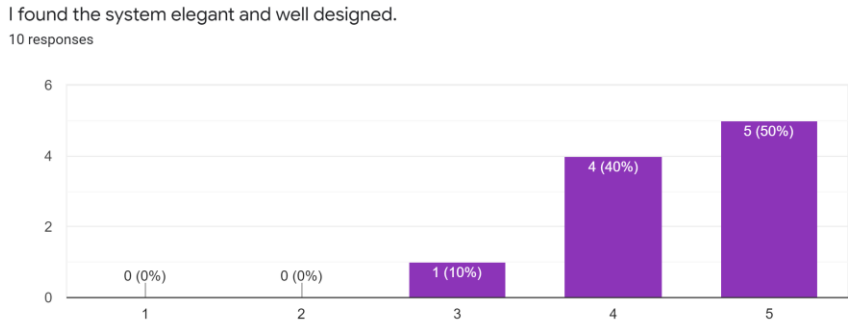


Figure 10 - System elegance and design question survey results

Description and animation straightforwardness scored a rating of 4.7 which exceeded expectations considering that the animations were made using a 2-D animator.

Users rated an average 4.4 for thinking that the webapp will encourage users to try yoga, but this question has to be considered in light of the fact it only reveals the users' thoughts about other users whereas empirical research suggested that a simple yoga app could solve the issue of new yoga practitioner exclusiveness.

User preferences were found to be applied to the poses at an average rating of 4.6 signifying that the preference filters worked well.

The optional questions related to the sequencing, pose specifications, pose descriptions, pose animations and beginning/ending poses were answered by 4 participants which is a good enough sample to get an idea for the pose accuracy. For almost all of the questions, the score was 5 signifying that the information displayed to the users is a good representation of what a class should be.

4.3.4 Summary

All the manual tests passed the user story acceptance criteria. The overall outcome was positive for the task-based part indicating that the entire webapp functionality works as planned. The SUS also showed positive results especially around the simplicity of the webapp which was one of the main goals of this project. Moreover, the system was reported to be easy to learn, accurate for pose/class details and elegant.

The comments in the end complimented additionally the user friendliness, color choice, interface simplicity and ease, music feature, intuitiveness and design appeal. The improvement suggestions included a potential switch to a single card that navigates to the next exercises instead of a list of cards. Another suggestion was to increase the size of the dropdown menus. Most importantly, several participants revealed both through comments and by personally reaching out to me that there are some interface issues. The webapp is not working properly on full screen and, depending on desktop screen size, some of the items, especially the cards, overlapped which made information partially hidden. The dropdowns in some cases did not work at all. Sometimes a refresh was needed to enable the usage of the form.

Chapter 5 Conclusion

5.1 Achievements

The status of the project is that the product is complete and fully functional as intended with minor improvements required for the future. The webapp has successfully achieved the initial requirements and, more specifically, has accomplished all the Must Have and Should Have objectives as well as the first two of the Could Have objectives. This seems fair given the MoSCoW way of prioritizing features and justifies the choices of features to be implemented. The actual days taken to complete the individual stories was very close to the original estimates given in Appendix A.

Separately, I was personally able to learn the basics of React which is a completely new framework for me. This was challenging given the time constraint, but will be very useful for future practice in the industry as it is a very popular framework.

The user testing results were overall positive and the comments revealed some flaws with the interface. Upon further investigation after the user testing evaluation was complete, bootstrap column sizes were identified as the issue for the card overlap issue as well as for the dropdowns not working. The reason for the latter was that the music player column was overlapping with the dropdown menus even though there was no video played yet. Regardless, this made the dropdowns non-clickable, but still visible. Both issues were fixed by increasing column sizes on smaller screens in order to occupy the entire length of the screen. This also fixed the problem of requiring full screen only for the app to work. The webapp is now fully responsive to desktop resizing and is optimized for no overlapping content.

5.2 What to improve

One thing that could be improved is to use a more complex animation software that enables 3-D animation for better visual representation especially with the more complicated poses which are difficult to see from the front or side on 2-D. With these specific poses, the Pivot Animator did represent the poses well enough from custom angles which were neither frontal nor side angles, but a 3-D animation would definitely be better as was noted by some of the participants. This would also improve the appeal of the webapp.

A second point for improvement was to adapt the webapp to mobile. Currently, some mobile adaptiveness is present via the bootstrap grid system usage of column sizes, but from testing on different devices, it was discovered that this does not work for all mobile screen sizes. This was out of scope for this project and could be improved in the future.

5.3 Future work

Linking back to the improvements, research can be made into 3-D animating products with which new, more comprehensive and easier to understand

animations can be made for each pose. This will be challenging and will require a significant amount of effort considering the time spent on the current relatively simple 2-D animations for only 54 poses.

The other improvement idea for future work is complete adaptiveness to mobile. This could be very valuable especially as users might want to use their mobile devices to follow the classes while performing the poses. Rather than improving the webapp, perhaps a mobile app could be developed which would be more convenient for users with mobile devices.

Something that could be an ongoing future work is to add a wider range of poses which would definitely benefit the webapp as it will generate more variations of classes and will apply all preferences in all cases rather than partial preferences where all preferences are not possible. More rules can also be added to the classes to be able to better represent Iyengar yoga as well as general yoga best practices. The extent of all Iyengar yoga practices was not applied for this project as priority was given to finishing a functional product using ReactJS.

More features could be added such as class reveal animations, login functionality, facebook login, favorites, custom class creation, email notifications, calendar review of classes done in the past and a system for progressing to more advanced poses. The product as it stands is the foundation of what a very complicated yoga webapp could become. Microservices, separation of concern and React Component usage will make the process of incorporating more features in the future easier.

Chapter 6 References

Bangor, A., Kortum, P. and Miller, J., 2008. An Empirical Evaluation of the System Usability Scale. *International Journal of Human-Computer Interaction*, 24(6), pp.574-594.

Beirne, G., 2014. *Yoga: A Beginner's Guide To The Different Styles*. [online] the Guardian. Available at: <https://www.theguardian.com/lifeandstyle/2014/jan/10/yoga-beginners-guide-different-styles> [Accessed 13 August 2020].

Codecademy, 2020. *React: The Virtual DOM*. [online] Available at: <https://www.codecademy.com/articles/react-virtual-dom#:~:text=In%20React%2C%20for%20every%20DOM,change%20what's%20on%20the%20screen.> [Accessed 25 August 2020].

Dragoni, N., Giallorenzo, S., Lafuente, A., Mazzara, M., Montesi, F., Mustafin, R. and Safina, L., 2017. Microservices: Yesterday, Today, and Tomorrow. *Present and Ulterior Software Engineering*, pp.195-216.

Feuerstein, G., 2006. A Short History of Yoga.

Goldberg, M., 2014. *Iyengar And The Invention Of Yoga*. [online] The New Yorker. Available at: <https://www.newyorker.com/business/currency/iyengar-invention-yoga> [Accessed 13 August 2020].

Heroku Dev Center, 2020. *Free Dyno Hours*. [online] Available at: <https://devcenter.heroku.com/articles/free-dyno-hours> [Accessed 14 September 2020].

Iyengar, B., 1997. *The Illustrated Light On Yoga*. New Delhi: HarperCollins Publishers India.

Iyengar, B., 2019. *Light On Life: The Yoga Journey To Wholeness, Inner Peace And Ultimate Freedom*. London: Hodder & Stoughton General Division.

IYENGAR YOGA (UK), 2018. *Iyengar Yoga Teaching Syllabus (Introductory Level)*.

Kaikkonen, A., Kallio, T., Kekäläinen, A., Kankainen, A. and Cankar, M., 2005. Usability Testing of Mobile Applications: A Comparison between Laboratory and Field Testing. *Journal of Usability Studies*, 1(1), pp.4-16.

Khattab, K., Khattab, A., Ortak, J., Richardt, G. and Bonnemeier, H., 2007. Iyengar Yoga Increases Cardiac Parasympathetic Nervous Modulation among Healthy Yoga Practitioners. *Evidence-Based Complementary and Alternative Medicine*, 4(4), pp.511-517.

Kolasinski, S., Garfinkel, M., Tsai, A., Matz, W., Dyke, A. and Schumacher, H., 2005. Iyengar Yoga for Treating Symptoms of Osteoarthritis of the Knees: A Pilot Study. *The Journal of Alternative and Complementary Medicine*, 11(4), pp.689-693.

Malhotra, M., 2018. *7 R easons To Choose Facebook's Reactjs In 2018*. [online] Valuecoders. Available at: <https://www.valuecoders.com/blog/technology-and-apps/5-reasons-choose-facebooks-reactjs/> [Accessed 25 August 2020].

Marlynn Wei, J., 2016. *New Survey Reveals The Rapid Rise Of Yoga — And Why Some People Still Haven't Tried It*. [online] Harvard Health Blog. Available at: <https://www.health.harvard.edu/blog/new-survey-reveals-the-rapid-rise-of-yoga-and-why-some-people-still-havent-tried-it-201603079179> [Accessed 11 August 2020].

Molina, M. and Sundar, S., 2018. Can Mobile Apps Motivate Fitness Tracking? A Study of Technological Affordances and Workout Behaviors. *Health Communication*, 35(1), pp.65-74.

Montigny, D., 2020. *The Unstoppable Trend Of Yoga*. [online] YOGI TIMES. Available at: <https://www.yogitimes.com/article/unstoppable-trend-yoga-infographic-business> [Accessed 11 August 2020].

Pascoe, M. and Bauer, I., 2015. A systematic review of randomised control trials on the effects of yoga on stress measures and mood. *Journal of Psychiatric Research*, 68, pp.270-282.

PNG MART, 2018. *Yoga Pose PNG Transparent Images*. [image] Available at: <http://www.pngmart.com/image/130433> [Accessed 3 September 2020].

React, 2020. *React – A Javascript Library For Building User Interfaces*. [online] Available at: <https://reactjs.org/> [Accessed 13 August 2020].

Ross, A. and Thomas, S., 2010. The Health Benefits of Yoga and Exercise: A Review of Comparison Studies. *The Journal of Alternative and Complementary Medicine*, 16(1), pp.3-12.

Studio Po, 2015. *Postings & Ponderings: Urdhva Baddhanguliyasana*. [online] Available at: <https://www.studiopoyoga.com/blog/2015/9/1/postings-ponderings-urdhva-baddhanguliyasana> [Accessed 14 September 2020].

Tummee, n.d. *Yoga Sequence Builder*. [online] Available at: <https://www.tummee.com> [Accessed 13 September 2020].

Williams, K., Petronis, J., Smith, D., Goodrich, D., Wu, J., Ravi, N., Doyle, E., Juckett, G., Kolar, M., Gross, R. and Steinberg, L., 2005. Effect of Iyengar yoga therapy for chronic low back pain. *Pain*, 115(1), pp.107-117.

Win, B., Piessens, F., Joosen, W. and Verhanneman, T., 2002. *On The Importance Of The Separation-Of-Concerns Principle In Secure Software Engineering*. Katholieke Universiteit Leuven, Dept. of Computer Science.

Yoga Alliance, 2016. *2016 Yoga In America Study Conducted By Yoga Journal & Yoga Alliance*. [online] Available at: <https://www.yogaalliance.org/2016YogaInAmericaStudy> [Accessed 13 September 2020].

Yoga Journal, 2020. *Yoga Poses*. [online] Available at: <https://www.yogajournal.com/poses> [Accessed 14 September 2020].

Yoga Selection, 2019. *Iyengar Yoga Online*. [online] Available at: <https://yogaselection.com/> [Accessed 13 September 2020].

Yogic Way of Life, 2019. *Paschima Namaskarasana - Reverse Prayer Pose*. [online] Available at: <https://www.yogicwayoflife.com/paschima-namaskarasana-reverse-prayer-pose/> [Accessed 14 September 2020].

YouTube, n.d. *Copyright*. [online] Available at: <https://www.youtube.com/howyoutubeworks/policies/copyright/#fair-use> [Accessed 24 August 2020].

YouTube Help, 2020. *Fair Use On Youtube*. [online] Available at: <https://support.google.com/youtube/answer/9783148?hl=en> [Accessed 24 August 2020].

Appendix A User Stories

For the following user stories 1 ideal day is the equivalent to 8 ideal hours and the priority values are 1 - “must have”, 2 - “should have” and 3 - “will not have”.

1. As a user I can select my preferences for the poses and submit the form.
Estimated Ideal Days: 8
Priority: 1

Acceptance Test: The user needs to be able to open the dropdown menus, click on a preference for each option and submit the overall form which will display the generated class underneath.

Details: The selected poses will be extracted from the database using all preferences where possible and then removing props preference, then type and then difficulty.

2. As a user, upon submitting the form I will receive a class which will be for my desired length preference.
Estimated Ideal Days: 5
Priority: 2

Acceptance Test: Upon submission, the generated class’ combined pose length will not be longer than the user’s class preference.

Details: The program will loop through the filtering process to select poses with as many preferences as possible until the combined pose length attribute values reach the user’s preference length.

3. As a user, upon resubmitting the form, I will have a new randomized class generated for me using the new preferences.
Estimated Ideal Days: 7
Priority: 1

Acceptance Test: Upon resubmitting, the user can see that the poses have changed with the exception of the constant default poses for beginning a class subpart and the overall end of the class.

Details: The fixed poses will be always added to the class outside of the loop for the rest of the classes to ensure they are always there. Upon resubmitting the form, an entirely new handleSubmit function will be triggered causing the generation of a brand-new class.

4. As a user, I can see the English and Sanskrit names, the exercise group, the type, the difficulty, the length and the props of each pose after the class is generated.
Estimated Ideal Days: 4

Priority: 1

Acceptance Test: Upon submitting, the user can see each pose's details.

Details: The pose details will be stored as attributes as part of the database, so upon API calls from the database, the details will be included.

5. As a user, I can see a visual representation of the pose in the form of an animation.
Estimated Ideal Days: 10
Priority: 1

Acceptance Test: Upon submitting the form, each pose will include an animation to aid its understanding.

Details: Each pose will have a URL as an attribute which will lead to a gif for its animation.

6. As a user, I can see the description of each pose when the form is submitted.
Estimated Ideal Days: 3
Priority: 2

Acceptance Test: Upon submitting the form, a description of the pose will be shown.

Details: This will be done via a card flip and the description will be on the back side of the card.

7. As a user, I can request music to be played as part of the form submission and can select from multiple types.
Estimated Ideal Days: 3
Priority: 2

Acceptance Test: Upon submitting the form, if a music type is chosen, music will be played from YouTube.

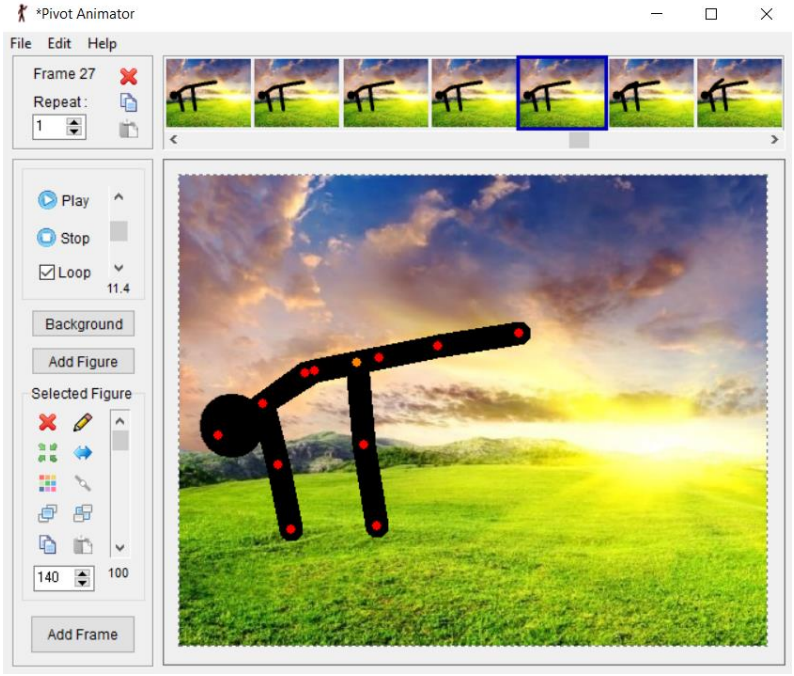
Details: The YouTube player will select a random URL from a JSON file storing all the links for the different types. Multiple URLs will be stored for each type.

8. As a user, I can control the volume, pause and unpause the music if I had selected a music type before submitting the form.
Estimated Ideal Days: 1
Priority: 2

Acceptance Test: Upon playing music, the user can control the player.

Details: The react-player functionality will ensure these actions are possible.

Appendix B Pivot Animator



Appendix C Additional Sources of Guidance for Animations and Pose Descriptions

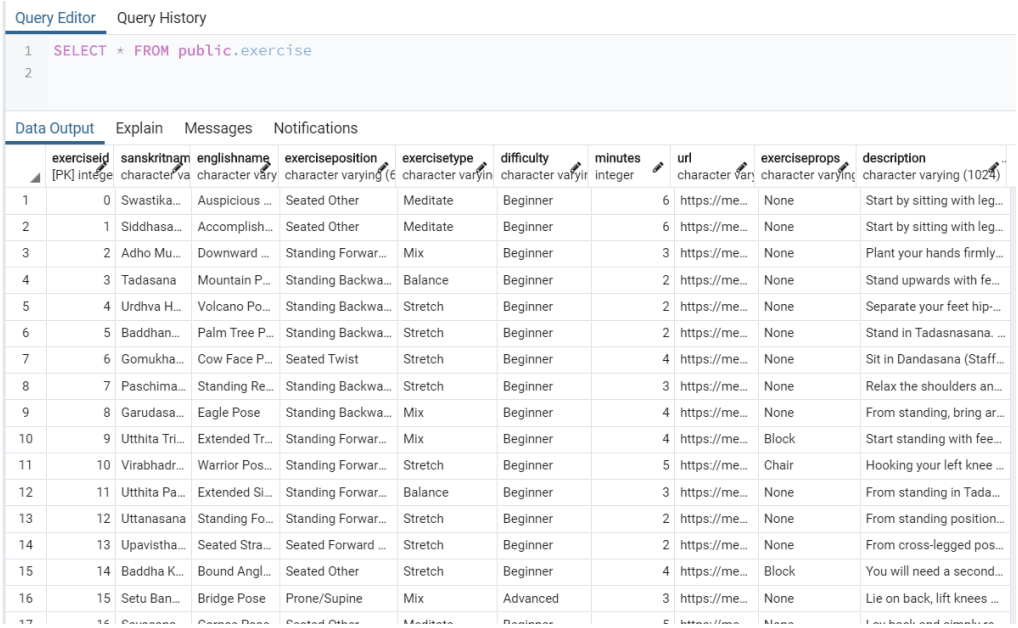
Online Yoga Teaching	https://www.youtube.com/user/onlineyogateaching
Helen and Julia Yoga	https://www.youtube.com/channel/UCGRkBmmlbG3ENDo7wr3hLQQ
VENTUNO YOGA	https://www.youtube.com/user/VentunoYoga
KinoYoga	https://www.youtube.com/c/KinoYoga
Roads To Bliss	https://www.youtube.com/c/RoadsToBliss
Howcast	https://www.youtube.com/c/howcast
Yoga by Candace	https://www.youtube.com/c/YogabyCandace1
Yoga Infusion	https://www.youtube.com/c/YogaInfusion
Alo Moves - Online Yoga Videos	https://www.youtube.com/channel/UCsksmxdgJtJp18iMYQpJg0A
YogaVibes	https://www.youtube.com/channel/UCdgifK61C5OtbWjmlwM7r1w
Bodhi Yoga Training Academy	https://www.youtube.com/c/BodhiYogaTrainingAcademy
Timiti	https://www.youtube.com/channel/UCHilgfaDiq2RPr7j2mJHt2g
Yoga With Tim	https://www.youtube.com/channel/UCciuZl2ydLCvN5txlLW0rIg
iHanuman Yoga Media	https://www.youtube.com/channel/UCHqprHA1rDds4CV4QdvUwyg
Sikana English	https://www.youtube.com/c/SikanaEN
Linda Black	https://www.youtube.com/channel/UCw04kYTE0eftZZNDmiJl2lw
The Art of Living	https://www.youtube.com/channel/UCvtVGPJsz1AST42hcgPMMqA
VINDHYACHAL YOGA SADHNA Yogi Aditya Shrivas	https://www.youtube.com/channel/UC_7JVp6CLnqvojlq2J6YDWg
essentialyoga	https://www.youtube.com/channel/UClGIa_QyvkJgW6zAHSZv5Yg
IYASE NEWS	https://www.youtube.com/c/IYASENEWS
Yoga Journal	https://www.youtube.com/channel/UC-QFhQGiB-WqOjRR98XZrkA
Iyengar Yoga Association of Greater New York	https://www.youtube.com/c/IyengarYogaAssociationofGreaterNewYork

Appendix D pgAdmin CREATE TABLE

CREATE TABLE exercise(

exerciseid INT NOT NULL,
sanskritname character varying(64) NOT NULL,
englishname character varying(64) NOT NULL,
exerciseposition character varying(64) NOT NULL,
exercisetype character varying(64) NOT NULL,
difficulty character varying(16) NOT NULL,
minutes INT NOT NULL,
url character varying (1024) NOT NULL,
exerciseprops character varying(64),
description character varying (1024) NOT NULL,
PRIMARY KEY (exerciseid));

Appendix E pgAdmin Database



Query Editor | Query History

```
1  SELECT * FROM public.exercise
2
```

Data Output | Explain | Messages | Notifications

	exerciseid [PK] integer	sanskritnam character va	englishname character vary	exerciseposition character varying (6	exercisetype character varyin	difficulty character varyir	minutes integer	url character vary	exerciseprops character varying	description character varying (1024)
1	0	Swastika...	Auspicious ...	Seated Other	Meditate	Beginner	6	https://me...	None	Start by sitting with leg...
2	1	Siddhasa...	Accomplish...	Seated Other	Meditate	Beginner	6	https://me...	None	Start by sitting with leg...
3	2	Adho Mu...	Downward ...	Standing Forwar...	Mix	Beginner	3	https://me...	None	Plant your hands firmly...
4	3	Tadasana	Mountain P...	Standing Backwa...	Balance	Beginner	2	https://me...	None	Stand upwards with fe...
5	4	Urdhva H...	Volcano Po...	Standing Backwa...	Stretch	Beginner	2	https://me...	None	Separate your feet hip-...
6	5	Baddhan...	Palm Tree P...	Standing Backwa...	Stretch	Beginner	2	https://me...	None	Stand in Tadasnasana. ...
7	6	Gomukha...	Cow Face P...	Seated Twist	Stretch	Beginner	4	https://me...	None	Sit in Dandasana (Staff...
8	7	Paschima...	Standing Re...	Standing Backwa...	Stretch	Beginner	3	https://me...	None	Relax the shoulders an...
9	8	Garudasa...	Eagle Pose	Standing Backwa...	Mix	Beginner	4	https://me...	None	From standing, bring ar...
10	9	Utthita Tri...	Extended Tr...	Standing Forwar...	Mix	Beginner	4	https://me...	Block	Start standing with fee...
11	10	Virabhadr...	Warrior Pos...	Standing Forwar...	Stretch	Beginner	5	https://me...	Chair	Hooking your left knee ...
12	11	Utthita Pa...	Extended Si...	Standing Forwar...	Balance	Beginner	3	https://me...	None	From standing in Tada...
13	12	Uttanasana	Standing Fo...	Standing Forwar...	Stretch	Beginner	2	https://me...	None	From standing position...
14	13	Upavistha...	Seated Stra...	Seated Forward ...	Stretch	Beginner	2	https://me...	None	From cross-legged pos...
15	14	Baddha K...	Bound Angl...	Seated Other	Stretch	Beginner	4	https://me...	Block	You will need a second...
16	15	Setu Ban...	Bridge Pose	Prone/Supine	Mix	Advanced	3	https://me...	None	Lie on back, lift knees ...

Appendix F Google Form

1	Open the webapp from https://iyengaryogamain.herokuapp.com. Choose any preferences you wish including for music and click on submit. Does the webapp generate a class consisting of multiple poses for you?
2	What is the first pose name?
3	What is the second pose name?
4	What is the last pose name?
5	What length preference did you choose? * in minutes
6	How many poses ended up being displayed?
7	Can you see the stickfigure animation of the first pose?
8	What is the third pose's Difficulty and Length?
9	What music is playing at the moment?
10	At what % is the volume roughly set at the moment?
11	Can you read the description of the 4th pose by positioning the cursor over the animation?
12	Choose pose preference as "None" for everything and music preference as "Disabled" and submit the form. Is there any music played/displayed now?
13	How many poses are displayed for this class now?

14	Have the poses generally changed with the exception of Tadasana, Dandasana, Sarvangasana and Savasana?

	Answer from 1 (Strongly Disagree) to 5 (Strongly Agree).
1	I think that I would like to use this system frequently.
2	I found the system unnecessarily complex.
3	I thought the system was easy to use.
4	I think that I would need the support of a technical person to be able to use this system.
5	I found the various functions in this system were well integrated.
6	I thought there was too much inconsistency in this system.
7	I would imagine that most people would learn to use this system very quickly.
8	I found the system very cumbersome to use.
9	I felt very confident using the system.
10	I needed to learn a lot of things before I could get going with this system.
11	I found the system elegant and well designed.
12	I found the descriptions and animations to be straightforward.
13	I think that the system will encourage users to try Yoga.
14	I found my preferences to be applied by the webapp successfully or at least partially for each pose (for example, each pose has at least the preferred difficulty).
15	(Optional and applicable only if Yoga experienced) I found the sequence of poses to be logical and to represent a balance of different pose types.
16	(Optional and applicable only if Yoga experienced) I found the pose specifications accurate.
17	(Optional and applicable only if Yoga experienced) I found the descriptions to be accurate for the poses.
18	(Optional and applicable only if Yoga experienced) I found the animations to be accurate for the poses.
19	(Optional and applicable only if Yoga experienced) Do the beginning and ending exercise seem appropriate for a class?

Based on original SUS system (Bangor, Kortum and Miller, 2008).

Appendix G Ethics Checklist

Introduction: Thank you for participating in the survey. Please note that all survey information will be anonymous, stored securely and will be deleted from Google Forms after the project is submitted. This survey will cover usability and functionality feedback on using the Iyengar Yoga Class generator which is a part of Ventsislav Antov's MSc Software Development Dissertation Project for Glasgow University. The survey will first ask you to complete tasks on the Iyengar Yoga Class Generator webapp and to answer questions related to those tasks. The survey will then ask you some additional questions regarding your overall experience with the webapp - some of these will be for yoga-experienced users only, so if you haven't practiced yoga, simply ignore them. You can withdraw your application at any time. The purpose of the webapp is to generate unique classes based upon your selected preferences. Note that you must consent to participating before starting the tasks.

Debriefing: Your contribution will be used to evaluate the webapp and make improvements where necessary. If you need to contact me, please do so through my email which is ven.antov@gmail.com. The project supervisor is Ornela Dardha and her email is ornela.dardha@glasgow.ac.uk.

School of Computing Science
University of Glasgow

Ethics checklist form for assessed exercises (at all levels)

This form is only applicable for assessed exercises that use other people ('participants') for the collection of information, typically in getting comments about a system or a system design, or getting information about how a system could be used, or evaluating a working system.

If no other people have been involved in the collection of information, then you do not need to complete this form.

If your evaluation does not comply with any one or more of the points below, please contact the Department Ethics Committee for advice.

If your evaluation does comply with all the points below, please sign this form and submit it with your assessed work.

---

1. Participants were not exposed to any risks greater than those encountered in their normal working life.

   Investigators have a responsibility to protect participants from physical and mental harm during the investigation. The risk of harm must be no greater than in ordinary life. Areas of potential risk that require ethical approval include, but are not limited to, investigations that occur outside usual laboratory areas, or that require participant mobility (e.g. walking, running, use of public transport), unusual or repetitive activity or movement, that use sensory deprivation (e.g. ear plugs or blindfolds), bright or flashing lights, loud or disorienting noises, smell, taste, vibration, or force feedback

2. The experimental materials were paper-based, or comprised software running on standard hardware.

   Participants should not be exposed to any risks associated with the use of non-standard equipment: anything other than pen-and-paper, standard PCs, mobile phones, and PDAs is considered non-standard.

3. All participants explicitly stated that they agreed to take part, and that their data could be used in the project.

   If the results of the evaluation are likely to be used beyond the term of the project (for example, the software is to be deployed, or the data is to be published), then signed consent is necessary. A separate consent form should be signed by each participant.

   Otherwise, verbal consent is sufficient, and should be explicitly requested in the introductory script.

4. No incentives were offered to the participants.

   The payment of participants must not be used to induce them to risk harm beyond that which they risk without payment in their normal lifestyle.

5. No information about the evaluation or materials was intentionally withheld from the participants.
   Withholding information or misleading participants is unacceptable if participants are likely to object or show unease when debriefed.

6. No participant was under the age of 16.
   Parental consent is required for participants under the age of 16.

7. No participant has an impairment that may limit their understanding or communication.
   Additional consent is required for participants with impairments.

8. Neither I nor my supervisor is in a position of authority or influence over any of the participants.
   A position of authority or influence over any participant must not be allowed to pressurise participants to take part in, or remain in, any experiment.

9. All participants were informed that they could withdraw at any time.
   All participants have the right to withdraw at any time during the investigation. They should be told this in the introductory script.

10. All participants have been informed of my contact details.
    All participants must be able to contact the investigator after the investigation. They should be given the details of both student and module co-ordinator or supervisor as part of the debriefing.

11. The evaluation was discussed with all the participants at the end of the session, and all participants had the opportunity to ask questions.
    The student must provide the participants with sufficient information in the debriefing to enable them to understand the nature of the investigation.

12. *All the data collected from the participants is stored in an anonymous form.*
    All participant data (hard-copy and soft-copy) should be stored securely, and in anonymous form.

Module and Assessment Name MSc Development Project for IT+ Dissertation

Student's Name Ventsislav Antov

Student's Registration Number 2504299a

Student's Signature 

Date 26.08.2020