2017

An Interactive Tutor for Java Programming

FINAL YEAR PROJECT

STUDENT NAME: [redacted] [redacted] [redacted]

PROJECT SUPERVISOR: PETER MCKENNA

No part of this project has been submitted in support of an application for any other degree or qualification at this or any other institute of learning. Apart from those parts of the project containing citations to the work of others, this project is my own unaided work.

Signed : ____________________

Table of Figures

Chapter 1 – Introduction

1.1 - Project Description

Computer science is a popular subject to study due to how widely required it is in today's business world. Computing requires a large amount of time to be spent working with digital media due to the nature of the course. Understanding a subject requires an ability to both comprehend and complete tasks outlined within that subject area. That is why one of the best methods of understanding the concepts used within computer science is using applications that allow the user to explore the different aspects of that topic.

This report will detail all of the steps taken to produce a piece of interactive teaching software that is aimed at teaching novice users a basic concept of programming. It will combine ideas from previously researched learning assistance concepts to allow the creation of an application that will explain a programming concept in detail.

Below is a description of each chapter that makes up this report, whilst also outlining each of the basic steps that has gone into the creation of the final application.

1.2 - Project Outline

This report will be composed of eight chapters that will discuss the different aspects involved with the research, development and evaluation of this project. Below is a breakdown of the headings used to section up the report into a more readable and organised layout.

1) Introduction – This will provide context to the report, giving a starting point and discussing potential options and ideas to follow up with research on.

2) Literature Review – The research aspect of this project, gathering relevant investigations and research papers from respected sources. This will be used to look at what research has already been completed in the field of e-learning.

3) Design Statement – This is where the foundation of the report will be, detailing initial design ideas for the application. This includes wireframes and pseudo code that is used to shape the creation of the application.

4) Implementation – This will detail the actual creation of the program, discussing the process of creation and relating back to previous research when implementing features.

5) Product Evaluation – An evaluation of the final program to highlight certain aspects of the application and to talk about what worked well and what could be changed.

6) Conclusion – Consolidation of findings from the report, discussing what worked well and what could be improved in future projects.

7) References – A list of referenced papers and work from respected sources.

8) Appendices – Extra documents relating to the report and project itself

1.3 - Aim

To teach a fundamental programming topic in an intuitive and beginner friendly way through the creation of an interactive application.

1.4 - Objectives

- Research existing work in the field of electronic learning to influence the optimal approach for this project.
- Explore programming languages have the most potential to create both an aesthetical and functional program.
- Develop and apply knowledge of programming techniques to produce a robust application.
- Show initiative when combining and expanding features of concepts from previous research.
- Test functionality and the effectiveness of the presented concepts through the use of the application.
- Reflect on previous work and projects to change and improve on the final produced piece of software.
- Evaluate the final application by detailing important findings and experiences that are relevant to the project as a whole.

Chapter 2 – Literature Review

Programming is a logic-based skill that requires an understanding of underlying mathematical concepts to succeed at even the most basic of levels of program creation. It is accepted that the process of learning to program is complicated and difficult, as evidenced by a survey investigating the worldwide failure rates of university and college students in introductory programming (Bennedsen and Caspersen, 2007). The results from this survey showed a 67% pass rate for students, meaning that the remaining 33% of students either failed, skipped of aborted the final exam. The process of teaching programming shares similarities to the process of teaching almost any other subject, meaning that a clear understanding of the cognitive workings of both teaching and learning any information is required as a starting foundation. There is a variety of different teaching methods and techniques that can be used to teach programming, with the most popular being electronic learning (E-learning).

E-learning is best defined as a method of delivering instructional content on digital media devices. It has a range of features available, allowing the creation of software that progressively changes depending on user inputs. The area of interest that this project will be focusing on is interactive elements within e-learning, which is most commonly used by programs. This is because programs allow the combination of presentational media such as videos and audio, whilst also making the user actively recall and apply their knowledge to advance further through the program. One of the biggest benefits to using e-learning it that it can be targeted towards asynchronous or synchronous e-learning in order to match the users preferred learning style. Asynchronous e-learning consists of software designed for independent learning, completely covering specific material in a way that allows the user to work freely at their own pace. Synchronous e-learning on the other hand is supplementary software that is used in conjunction with more typical methods of learning, such as lectures or instructor led tutorial classes. Depending on the users' preferred learning approach, they may favour one style more than they may favour the other.

Two online seminars were monitored to analyse the effectiveness of asynchronous and synchronous learning at a post-secondary level of education (Hrastinski, 2008, p.54). Students were interviewed using a discussion forum and one-to-one conversations to collect opinions on the two types of learning styles. The conclusion from this analysis was that asynchronous e-learning was better at supporting cognitive participation, whereas synchronous e-learning better supported increased motivation from the students. This conclusion matches results discussed in an earlier dissertation from the same author. Hrastinski noted in his 2007 paper that 'Asynchronous communication may induce increased cognitive effort since students have more time for reflection. Synchronous communication may induce increased motivation and decreased ambiguity because of possibilities for immediate feedback' (Hrastinski, 2007, p. 45). This provides evidence to the idea that choice of learning style is dependent on which skills the user would benefit from assistance with.

Users that struggle staying focused and motivated when working appear to benefit from instructional-based content to keep to a steady workflow, preventing distractions and forcing them to match the outlined pace that occurs with a synchronous style of learning. Users that have difficulty remembering and recalling information would appear to benefit from asynchronous learning due to the lack of a generalised pace that is outlined when working with other people, allowing more time to be spent revising over areas of particular difficulty to that specific user. It appears that using a combination of synchronous and asynchronous learning techniques would produce the best results by ensuring that users would get to experience both sets of advantages from each techniques. This is one of the reasons that universities separate the delivery methods of teaching materials into lectures, laboratories and seminars. Lectures mostly cover the synchronous learning style, laboratories mostly cover the asynchronous learning style and seminars exercise both at the same time. By covering all aspects of information delivery methods, students get full advantage of the benefits from each style of learning.

Another aspect of how users take in information through e-learning is detailed in the knowledge construction view, based on principles found through cognitive scientific research. These principles outline three important concepts when users are learning through electronic multimedia. The first principle is that people have separate channels to process visual material and auditory material. This means that three information-processing methods (verbal, pictorial and pre-existing knowledge) can be focused towards learning one particular topic at once. The diagram, adapted from Mayer's work, shows this. Mayer highlighted the importance of combining different methods of presentation, which in this case leads to the modality effect (Mayer et al, 1996). This effect occurs when both visual and auditory information is presented together at the same time, rather than separate. The same information from different types of media is acknowledged by the senses and passed into the working memory, allowing the organisation and combination of information from the verbal channel, pictorial channel and prior knowledge from long-term memory.

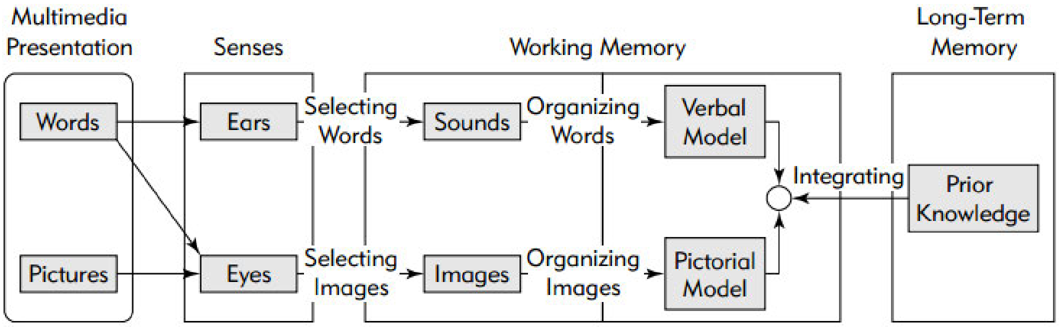


Figure 2.1.1 (Clark and Mayer, 2011) – Cognitive Theory of Multimedia Learning (Adapted from Mayer 2005)

The second principle is that there is a limit to how many pieces of information that can processed at one time. As evidenced previously by Mayer (Figure 2.1.1), there are limited cognitive channels that can only process a finite amount of information at one time. Overexposure to large quantities of information in a small amount of time can have a negative effect on the overall learning process.

The final principle is that learning is most effective when people are engaged in learning relevant and organised material that can be integrated with pre-existing understanding. Structure and planning is important to ensuring that people are learning appropriate and linked information to their previous knowledge.

2.1 E-learning design principles

It is important to understand the most efficient teaching techniques to satisfy people's different learning requirements. Large amounts of resources have been invested into exploring effective techniques that are designed to simplify the learning process. Some of the best examples of this are the design principles that each cover a different aspect of organising taught material in a productive way. An interesting feature of these principles is how easily they can be combined and used together. One person who has covered these principles in detail is Richard E. Mayer, an educational psychologist who has been prominent within the world of educational multimedia. A large number of the following principles have been influenced heavily by his work.

The first and most prominent design principle is the use of multimedia in e-learning, studied in-depth by Richard E. Mayer. This principle states that the combination of text and graphics is more effective at presenting a given topic, rather than using one or the other in isolation. The multimedia effect is a method of simplification that is especially useful for beginners when starting to look into a specific subject. As said previously, Mayer has been incredibly influential to the development of this principle. He has conducted research and experiments into the effectiveness of this concept, where he has noted how significant the multimedia effect is when it is targeted towards learners of a novice skill level. This is shown in his book where he says, "There is evidence that our recommendation to use words and graphics is particularly important for learners who have low knowledge of the domain (whom we can call novices)" (Clark and Mayer, 2011). He stated that experiments performed by him, his colleagues (Gallini and Mayer, 1990) and other researchers (Ollerenshaw, Aidman and Kidd, 1997) reported that in a comparison of learners with high and low knowledge about a given topic, the low knowledge learners performed better with the combination of text and graphics (prominently animation). The argument for using animation as an alternative to static media provides some interesting points from another perspective. An example of this is the quote: "dynamic displays can distort reality in various ways such as slowing down some processes and speeding up others, showing an object or phenomenon from different or changing viewpoints, augmenting the display with cues to draw viewers' attention to the most relevant parts," (Hegarty, 2004). It is correct that dynamic elements provide a different viewpoint by actively changing the presented information; however, it needs to be used in moderation to preserve the reader's attention. A possible solution for this would be the incorporation of forced interaction in order to use the dynamic element, requiring the learner to actively process information to make the element progress (almost like a demonstration).

Visual appeal is one of the biggest factors at obtaining and holding a person's attention. The contiguity, coherence, redundancy and segmentation principles are closely linked as they are all primarily focused on controlling different aspects of the user interface design. Below is a brief outline of each one:

- Contiguity - Alignment of corresponding words and graphics.
- Coherence - Elimination of unnecessary information.
- Redundancy – Use narration and animation instead of text.
- Segmentation - Break down information into bite-sized pieces.

Contiguity can be broken down into two categories. Spatial contiguity is the idea that people learn more deeply from multimedia when corresponding words and pictures are presented in close proximity of each other. Temporal contiguity is the idea that people learn more deeply from multimedia when corresponding animation and narration are presented simultaneously rather than successively (Mayer 2005). This principle relates back to the previously discussed cognitive theory channels that determine what types of media can be processed simultaneously. Making use of text in conjunction with static, dynamic and audial teaching media at the same time provides the optimal setup for information absorption. These media techniques supplements the three features of working memory (dual channels, limited capacity and active processing).

Coherence is the process of guiding the learner through the material in a way that leads towards the most relevant information. Presenting too much information at once can lead to the learner losing concentration and getting confused, a process called information overload. Mayer (2005) summarises the overload problem when he says, "This overload scenario can occur when a multimedia instructional message contains too much detail, embellishment, or gratuitous information or when the layout of material is confusing." It appears that dynamic media would suffer from this problem the most due to how flexible they are, leading to people inserting too much information into the one animation. Applying the coherence principle encourages the consolidation of vital material in a way that enables the learner to focus their cognitive capacity towards essential processing.

Redundancy shares some similarities with the coherence principle; however, it has more of a focus on the removal of text and moves emphasis towards using animations and narration. It makes the point that presenting text and audio simultaneously leads to the text becoming redundant. Repeating near identical narration and text can result in reduced focus on the content as some people struggle to divide attention towards two sources of identical information being presented in different ways (Hoffman, 2006). That means that this principle encourages the removal of text to force focus towards the animations or audio elements that work together in a more fluid way. Out of all the discussed principles, this one has been critiqued by multiple cognitive psychologists. This principle depends heavily on the learners' preference and capabilities. Sorden (2005) debated that using text along with diagrams could be helpful for a complete beginner when trying to understand a concept. On the other hand, he accepted that a more experienced learner might not need the additional text, making it redundant. Both perspectives raise valid points and concerns, so it would be best to assume the effectiveness of this particular principle is heavily dependent on context.

Segmentation is the last principle that has a large effect on the physical presentation of information. It is the idea that splitting up or breaking down information into a series of steps makes allows the learner to work at their own speed to understand the material. It serves a similar purpose to the coherence principle by eliminating the possibility of overloading a learner with too much information at once (Clark and Mayer, 2011). Instead of forcefully guiding them through the material, they are given the choice of what they want to look at based on their specific understanding at that time. The most obvious place that this is used is in websites as it allows in-depth customisation of the layout and provides a method for grouping related information.

The modality principle is different to the visual based principles in the way that it states that there is an advantage to auditory material over text. It encourages focus towards information being presented as speech instead of pure text. This principle appears to be most effective when the material is considered to be challenging for the learner (Tindall-Ford, Chandler, Sweller, 1997). The reason for this is that information is presented at a faster pace, forcing the learner to stay focused to keep up. Retaining focus means that more information is processed, resulting in more content being absorbed in by the learner.

The final design principle to discuss is personalization. This is the concept of engaging with the learner by adopting a more human like tone when presenting information. This is particularly useful when working with asynchronous media, encouraging the user to keep concentration and interaction with the specified media. Clark and Mayer (2011) detailed, "learners know that the character is not really in a conversation with them, but they may be more likely to act as if the character is a conversation partner." The current limitations to technology means that realistic human like software or hardware is hard to develop and use.

It is not expected that the user will believe they are interacting with software that has a personality; however, adopting a human like tone or language would be more familiar to the learner. Talking in a way that simplifies content whilst still providing relevant information makes it easier for the learner to process.

Overall, this section has covered the core concepts that are behind efficiently using multimedia in e-learning. The design principles will be carried across into the final application to ensure that the final product makes use of the most optimal techniques.

Chapter 3 – Design Statement

3.1 - Design Overview

The first decision made during the design process was in regards to which concept would be demonstrated. The target audience for this program was decided to be novice users, as they would get a larger benefit by using an external application. Users more familiar with Java and fundamental programming concepts would be able to pick up new information through experimentation inside an actual programming environment. Novice users would need to understand the concepts before being able to program in a programming environment, meaning that an external application would allow them to focus on a smaller area (Coding and concept wise). It was also decided that this project would act as a way of testing a re-usable presentation method for explaining any fundamental concept.

For the previously mentioned reasons, the chosen concept of focus was arrays. This data structure is a fundamental concept that is simple once understood, however, it could be difficult for novice users to understand completely. An array can be demonstrated in a visual way, making it idea to use as an initial idea.

Based off previous research into e-learning concepts and techniques, it was decided that the application was going to be focused towards an asynchronous structure. The reason for that is that it encourages the user to take the initiative with their learning. They can focus on specific areas that they struggle with by giving them the ability to work at a comfortable pace and to their own time restrictions. Another reason is that it offers a larger amount of flexibility with application design and implementation. Synchronous applications need to work in parallel with other teach material and are harder to change and update, due to how reliant they are on another structure.

3.2 – Functionality design

It was decided that the application would be created using a variety of suitable languages and libraries that are detailed in the next section. The decision was made to create an application with three main interactive points.

The first is an introduction page that will provide a brief rundown of the array structure. The second is a demonstration that will require the user to interact with on-screen elements to display information. The third is a dynamically generated question page. All of these pages were created using a combination of JavaScript, HTML and other relevant libraries. I have very little experience using these languages, so I spent a large amount of time researching and experimenting with their different capabilities.

The main data handling was done using Java, as I was more comfortable using this language for complex data management. The data handling process covered the organisation of the questions for the dynamic question page. It was decided that the questions would be stored using a MySQL database on the university mudfoot server. That means that the questions could be created and modified freely and the application would constantly collect the most up to date questions. This was done using a model view controller (MVC) pattern. The model aspect is the part of the Java application that collects and stores the question data. The view aspect is the HTML and JavaScript display on a web page. The controller is the Java to JavaScript functionality that allows the manipulation of which data is displayed depending on user inputs.

3.3 – MVC Idea Originality and Declaration

The idea of the MVC structure came from an Enterprise Programming assignment completed earlier in the year. A similar structure formed the grounds of that assignment, so I had some understanding of how to create the Java side of the code. All of the code used in the creation of this project and the Enterprise Programming assignment was created by me.

Below is a list of language and library choices along with a description detailing why they were chosen.

3.4 - Language and Library choices

Below is a list of languages and libraries that were used during the development of this application, along with a small description explaining the reason that they were chosen.

Java – The application is teaching this language, so elements of Java would have to be included. Java is also the main language that has been used throughout the majority of programming work throughout units studied at this university. It is a language that it comfortable to work with, so it will be used for some of the more complex and technical background data management.

JavaScript – The project initially detailed that the application would focus on teaching JavaScript programming. It was decided that focusing on Java would be a better choice because of how much more confident I was using it. It would be easier for me to explain concepts in a language that I was more familiar using.

During my course of study, I have not had the opportunity to work with JavaScript in large detail. It is a language that is favoured for its ability to work with HTML elements and create user interfaces. I decided that I would challenge myself and use JavaScript (along with additional libraries) to build the application. JavaScript provides a large amount of functionality and is ideal for this project.

HTML – It was decided that the application would talk the form of a web-application, as it would be easily accessible once hosted. HTML provides the ability to create a web page to interact with the completed application.

JQuery – This JavaScript library provides more functionality when used alongside HTML pages and regular JavaScript. It performs very similar functions as JavaScript in a much more convenient way.

JQuery UI – JQuery UI is an extension of the JQuery library that provides various visual JQuery elements that can be used with HTML. This was used to make the application look more visually impressive and provide additional functionality.

MYSQL – This was used to create the database used to store the questions. It was accessed using JQuery Ajax calls to return requested question data.

GSON – GSON is a Java library that allows the re-formatting of text into JSON format. This is useful when used with JQuery and HTML as JSON one of widely used text formats that websites can display.

3.5 - User Interface Design

Here are the wireframes that were used as initial ideas for the layout of the application.

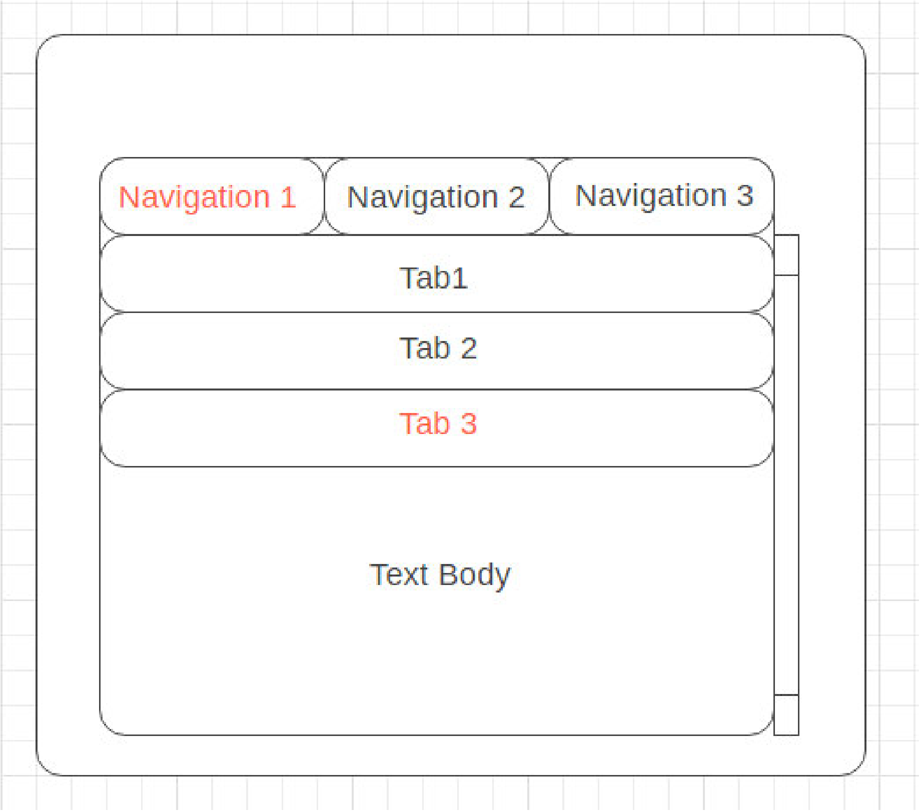


Figure 3.5.1 – Navigation, introduction page design (Created using: https://wireframe.cc/).

This is first and default page that the application opens up with, introducing the array data structure. It will use an accordion tab layout to segment information, allowing the user to focus on one specific area. As found from previous research, segmenting information prevents the overloading of information and provides a natural navigation path that the user will follow.

Initial ideas for the tab contents are:

What are arrays?

Why are they used?

How do they work?

Java example

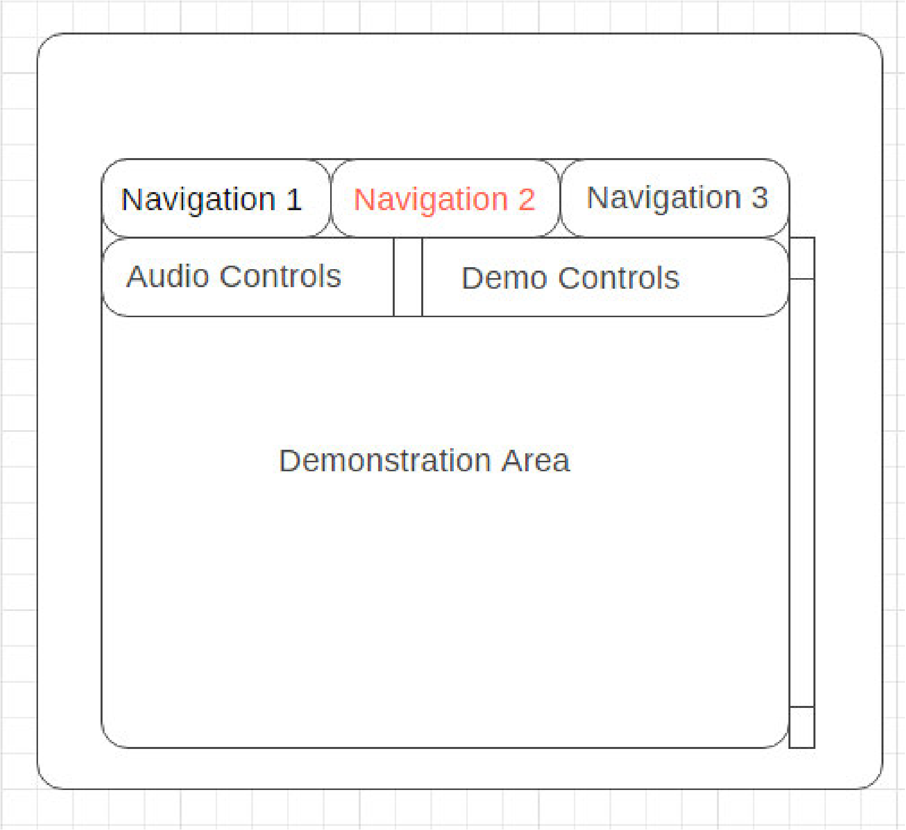


Figure 3.5.2 – Navigation, demonstration page design (Created using: https://wireframe.cc/).

This is the second page of the application that will contain an interactive demonstration for the user to experiment with. The purpose of this page is to visually show how an array functions, showing how changes to the code have an effect on the structure of the array.

This page will also make use of an audio voice over. This is because of the redundancy principle that encourages the usage of animations and audio instead of text. The voice over will explain how the demonstration works and how those changes effect the array.

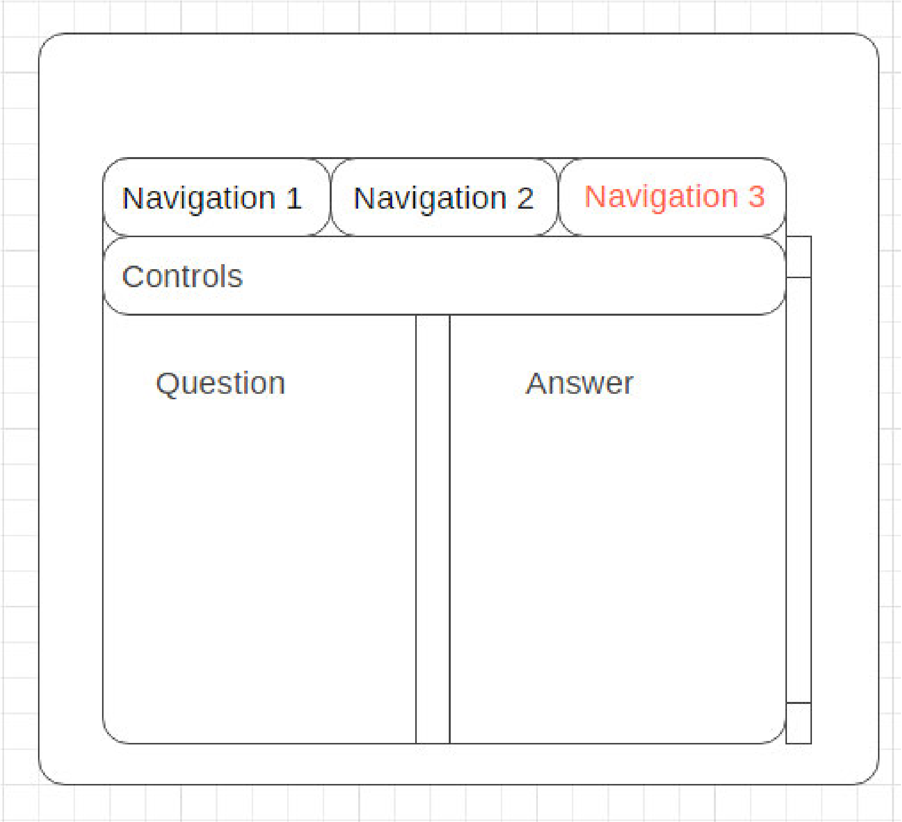


Figure 3.5.3 – Navigation, questions and answers page design (Created using: https://wireframe.cc/).

This page is where the questions and answers will be displayed. The user will select a question difficulty rating and the table will be populated with questions and textboxes for the answers. There will also be a button that will mark the users answers once pressed, displaying the results at an appropriate location on the page. This has not yet been decided as presenting dynamic elements using HTML and JQuery could provide problems with the intended design.

Chapter 4 - Implementation

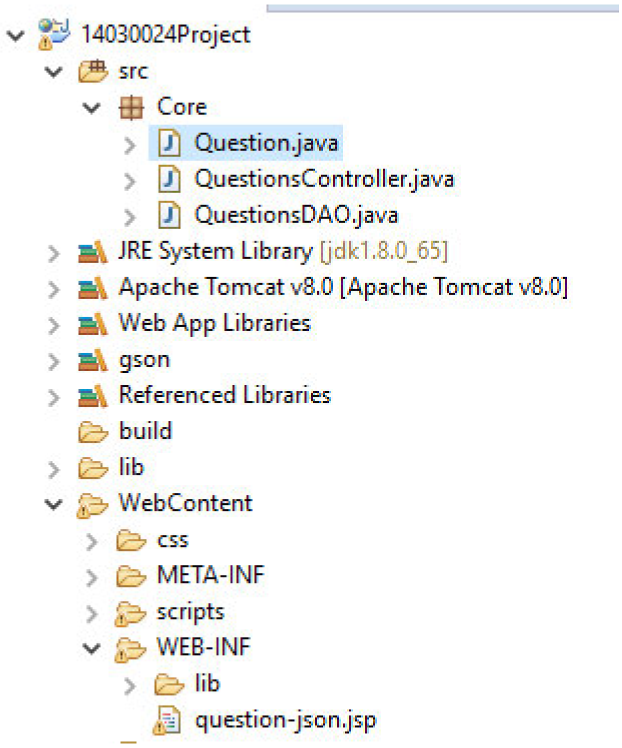
4.1 – Implementation Overview

The process of creating the application took place in two stages. The first was the creation of the Java MVC project, which required the setting up of multiple Java classes and an SQL database. The second stage of implementation was creating the front-end display using JavaScript, HTML, JQuery and JQuery UI.

The following sections are in the order of which they were completed.

4.2 – Java Code

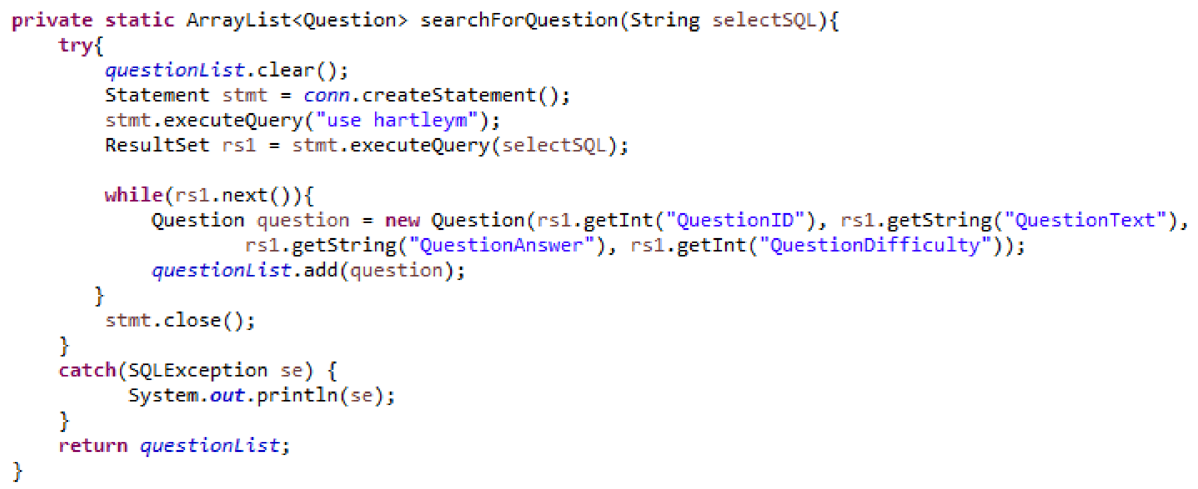
The first task was to create the java classes that would handle the data management of the application. In order to set up the MVC structure, three classes and one .JSP output page were created.



```
14030024Project
    src
        Core
            Question.java
            QuestionsController.java
            QuestionsDAO.java
    JRE System Library [jdk1.8.0_65]
    Apache Tomcat v8.0 [Apache Tomcat v8.0]
    Web App Libraries
    gson
    Referenced Libraries
    build
    lib
    WebContent
        css
        META-INF
        scripts
        WEB-INF
            lib
            question-json.jsp
```

Figure 4.2.1 – Java and JSP file structure.

The “Question.java” class was the first one to be created. Insider are getters and setter methods that are used in alongside a Question object constructor. This structure is used to create each question object that is then saved into an array list in other areas of the program.



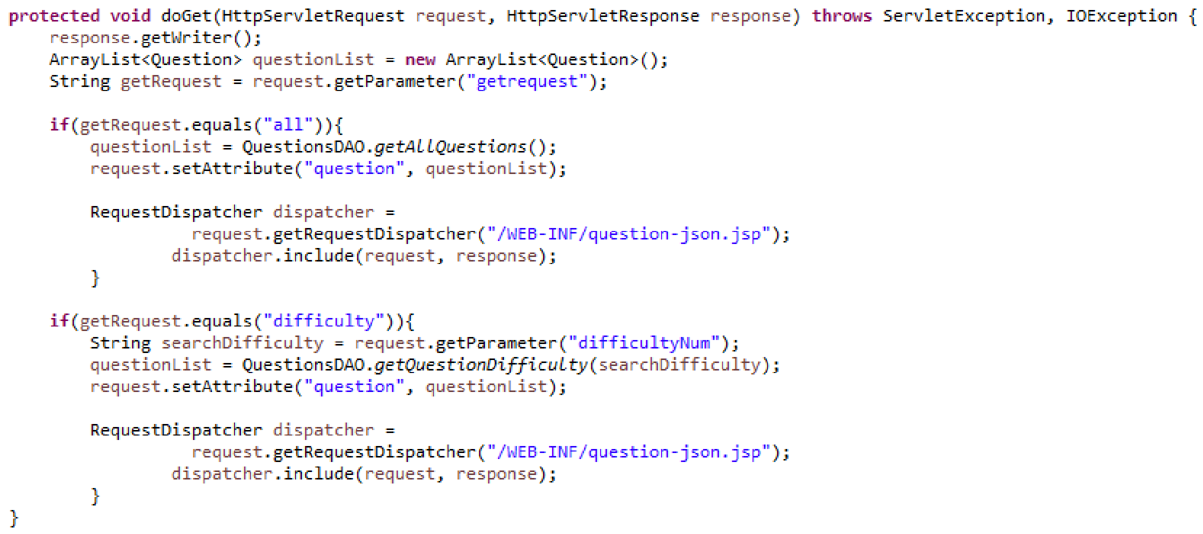
```
private static ArrayList<Question> searchForQuestion(String selectSQL){
    try{
        questionList.clear();
        Statement stmt = conn.createStatement();
        stmt.executeQuery("use hartleym");
        ResultSet rs1 = stmt.executeQuery(selectSQL);

        while(rs1.next()){
            Question question = new Question(rs1.getInt("QuestionID"), rs1.getString("QuestionText"),
                    rs1.getString("QuestionAnswer"), rs1.getInt("QuestionDifficulty"));
            questionList.add(question);
       }
        stmt.close();
    }
    catch(SQLException se) {
          System.out.println(se);
    }
    return questionList;
}
```

Figure 4.2.2 – "Search for questions" method in QuestionsDAO class.

The second class was "QuestionsDAO.Java", which is responsible for collecting the questions from the SQL database. It uses a connect method to establish a connection to the database, using supplied log in details and a database URL.

The next method in this class is used to search for a question. Upon call, this method takes in a SQL select statement in the form of a string. It then calls the connect method to get access to the database and executes the query. The result of the query is returned, where it is broken up into a question object by using the constructor from the "Question.Java" class. Each object is added onto an array list, which is returned to the method that initially called for the questions.

The other two methods are used to construct the SQL statement based on which was called. One statement is used to collect every question from the database and the other statement collects questions that match a specified difficulty rating. This is because the question page provides the option to filter and display certain difficulty questions.



```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    response.getWriter();
    ArrayList<Question> questionList = new ArrayList<Question>();
    String getRequest = request.getParameter("getrequest");

    if(getRequest.equals("all")){
        questionList = QuestionsDAO.getAllQuestions();
        request.setAttribute("question", questionList);

        RequestDispatcher dispatcher =
                  request.getRequestDispatcher("/WEB-INF/question-json.jsp");
                dispatcher.include(request, response);
        }

    if(getRequest.equals("difficulty")){
        String searchDifficulty = request.getParameter("difficultyNum");
        questionList = QuestionsDAO.getQuestionDifficulty(searchDifficulty);
        request.setAttribute("question", questionList);

        RequestDispatcher dispatcher =
                  request.getRequestDispatcher("/WEB-INF/question-json.jsp");
                dispatcher.include(request, response);
        }
}
```

Figure 4.2.3 – Question Controller class.

The final Java class of the Java project is the "QuestionsController.Java". This is used to pass the question list across to the display page. Different functions are performed depending on the address that is called from within the application. The address that is passed to this class as the "getRequest" variable can take two forms:

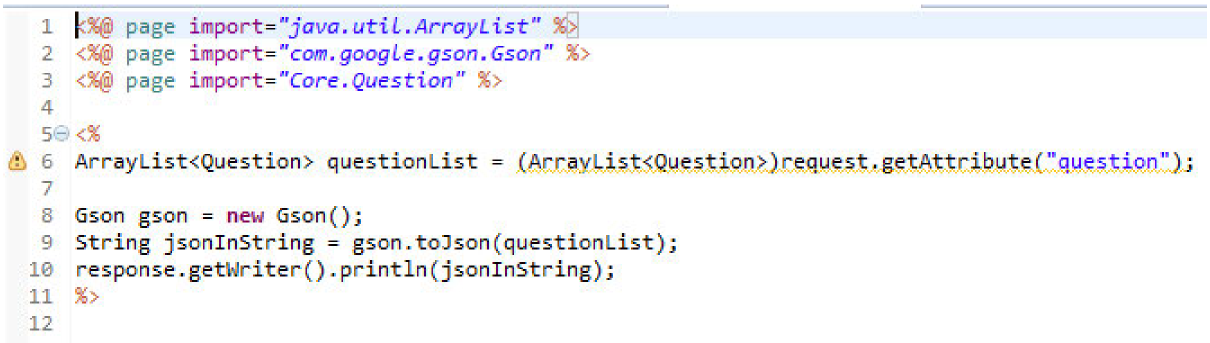
http://localhost:8080/[redacted]QuestionsController?getrequest=all

Or

http://localhost:8080/[redacted]QuestionsController?getrequest=difficulty&difficultyNum=1

The parameters at the end of the address specify which method is used in the question controller class. In the first address, the request is for "all" questions. The second address is requesting for questions with a difficulty number of "1". This number will change depending on what the user has selected when generating the question page.

Once the request has been processed and the array list with the correct questions in has been returned, the list is passed to the "question-json.jsp" page to be formatted.



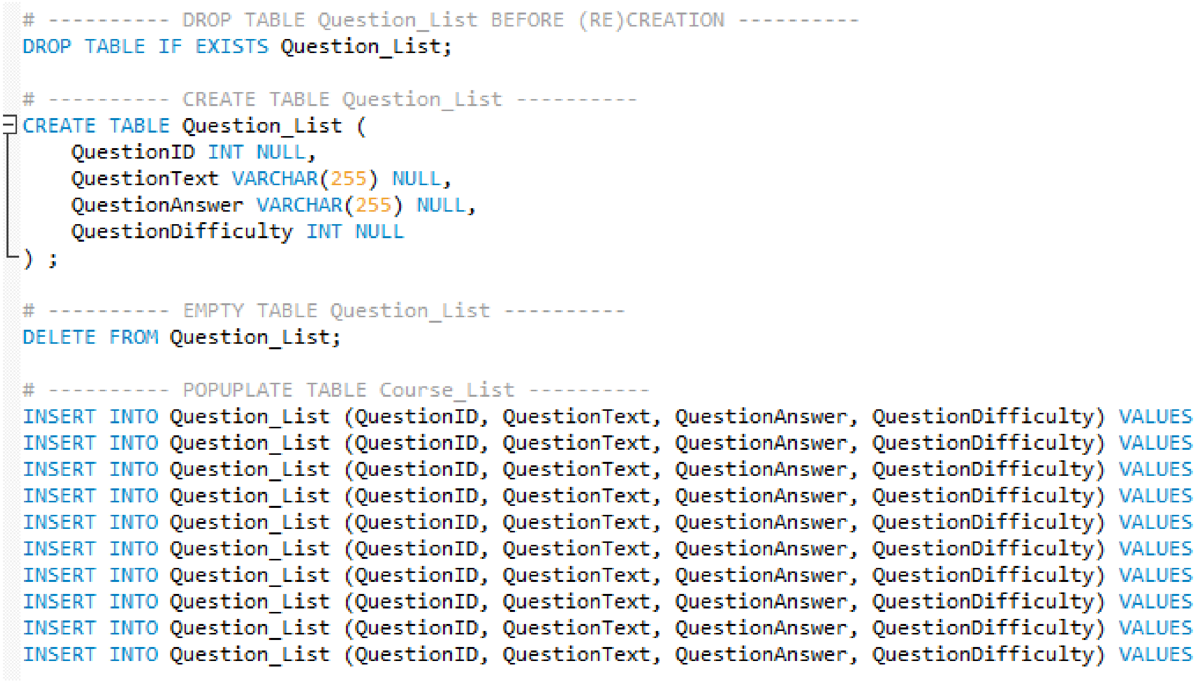
```
<%@ page import="java.util.ArrayList" %>
<%@ page import="com.google.gson.Gson" %>
<%@ page import="Core.Question" %>

<%
ArrayList<Question> questionList = (ArrayList<Question>)request.getAttribute("question");

Gson gson = new Gson();
String jsonInString = gson.toJson(questionList);
response.getWriter().println(jsonInString);
%>

```

Figure 4.2.4 – Question-Json.jsp page.

The JSP page uses the GSON library mentioned previously to convert the plain string into JSON format. The GSON object handles all of the conversion process, which is the main reason why the GSON library was used.

4.3 – Additional Media

A variety of different media was used for this project in addition to the Java and JQuery/HTML code.



```
# ---------- DROP TABLE Question_List BEFORE (RE)CREATION ----------
DROP TABLE IF EXISTS Question_List;

# ---------- CREATE TABLE Question_List ----------
CREATE TABLE Question_List (
    QuestionID INT NULL,
    QuestionText VARCHAR(255) NULL,
    QuestionAnswer VARCHAR(255) NULL,
    QuestionDifficulty INT NULL
) ;

# ---------- EMPTY TABLE Question_List ----------
DELETE FROM Question_List;

# ---------- POPUPLATE TABLE Course_List ----------
INSERT INTO Question_List (QuestionID, QuestionText, QuestionAnswer, QuestionDifficulty) VALUES
INSERT INTO Question_List (QuestionID, QuestionText, QuestionAnswer, QuestionDifficulty) VALUES
INSERT INTO Question_List (QuestionID, QuestionText, QuestionAnswer, QuestionDifficulty) VALUES
INSERT INTO Question_List (QuestionID, QuestionText, QuestionAnswer, QuestionDifficulty) VALUES
INSERT INTO Question_List (QuestionID, QuestionText, QuestionAnswer, QuestionDifficulty) VALUES
INSERT INTO Question_List (QuestionID, QuestionText, QuestionAnswer, QuestionDifficulty) VALUES
INSERT INTO Question_List (QuestionID, QuestionText, QuestionAnswer, QuestionDifficulty) VALUES
INSERT INTO Question_List (QuestionID, QuestionText, QuestionAnswer, QuestionDifficulty) VALUES
INSERT INTO Question_List (QuestionID, QuestionText, QuestionAnswer, QuestionDifficulty) VALUES
INSERT INTO Question_List (QuestionID, QuestionText, QuestionAnswer, QuestionDifficulty) VALUES
```

Figure 4.3.1 – SQL database.

The first item in this list is the SQL database, created using MySQL workbench. It involved the creation of a Question list that had attributes relating to a question. Each individual question was then inserted into the database below. The main reason a SQL database was used for this project is that it allows the questions to dynamically change when needed. There is no need to access the application code at all, which means that the user of the application could swap which database is being accessed when needed.

“This is an example of an array in action. You can control this demonstration using the plus and minus buttons in the control box. As you use these buttons, you will see how the structure changes as you increase or decrease the initialization value for the array. The piece of Java code responsible for that particular array layout is visible next to the control buttons after you press one of them.

The onscreen display will show you the index value of each element. The main thing you want to get your head around is that arrays start at zero. It's stressed so much because it’s the easiest thing to forget when you start working with this data structure.”

Figure 4.3.2 – Audio Script.

This is the script used to record the audio file for the demonstration. The main thing to note is the tone and language used. This reflects back on the personalisation principle researched during the literature review. Informal language helps to give the application a more casual and relaxed feel, assisting with the learning process.

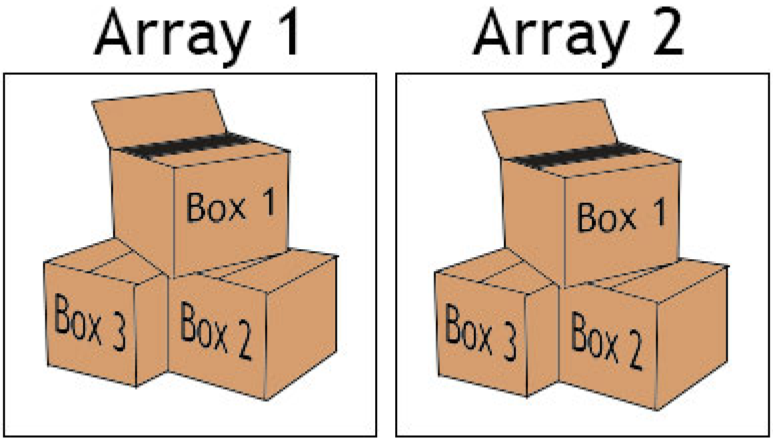
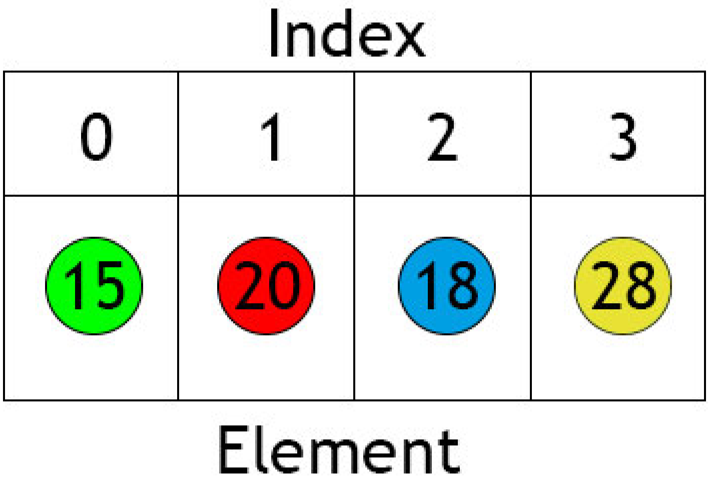


Figure 4.3.3 – Introduction Image 1.



Index

0	1	2	3
15	20	18	28

Element

Figure 4.3.4 – Introduction Image 2.

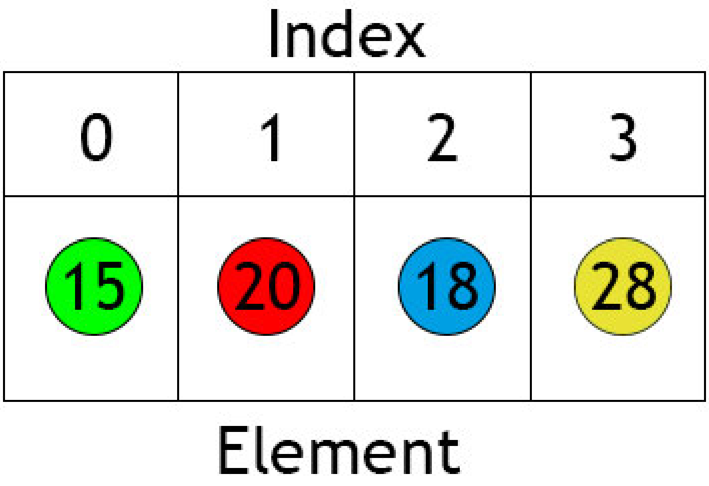


Figure 4.3.5 – Introduction Image 3.

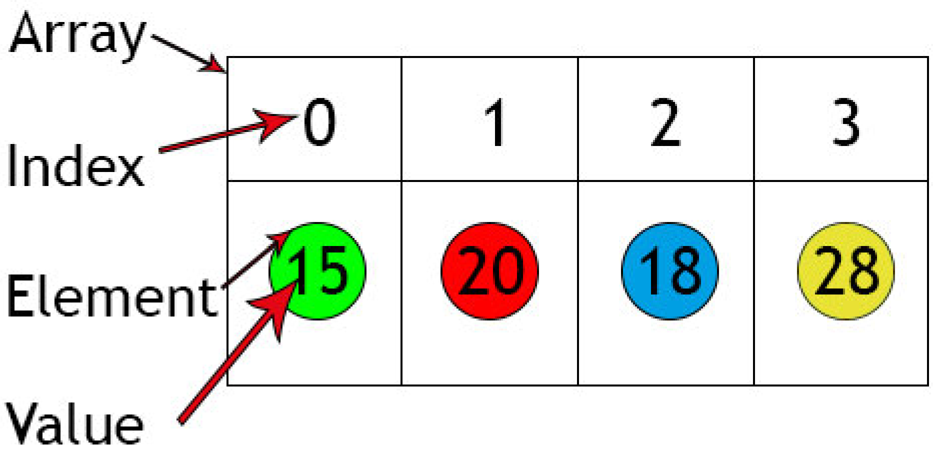
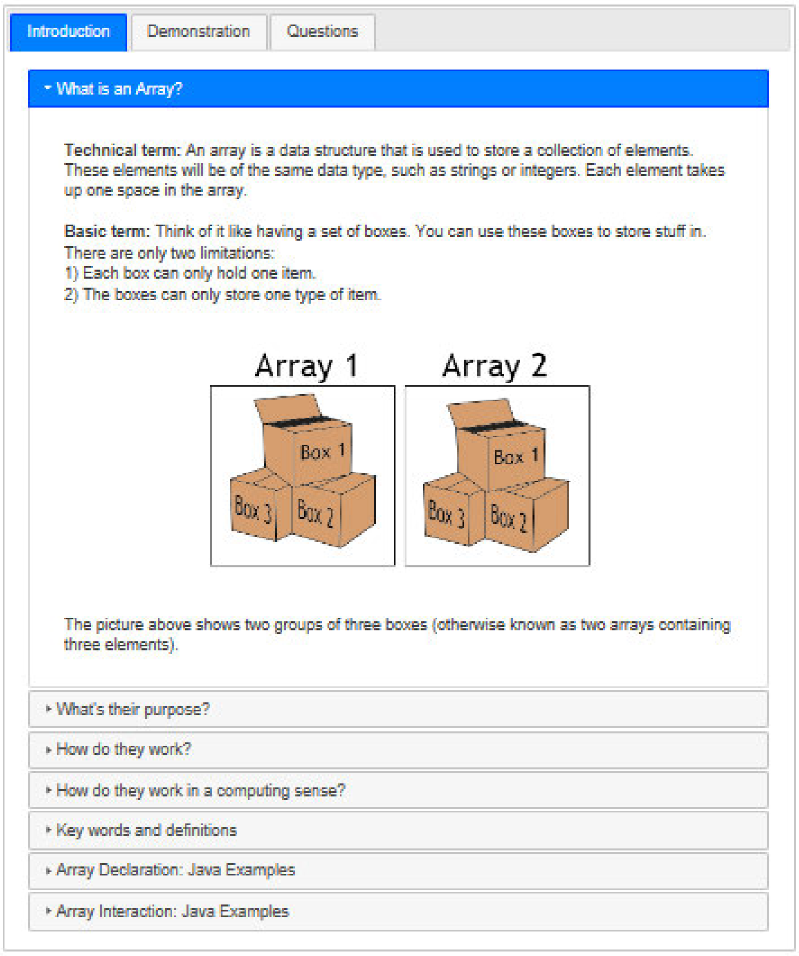


Figure 4.3.6 – Introduction Image 4.

The final additional media that was used in this project is the images for the demonstration. Each of these images was hand-drawn using Adobe Photoshop. The purpose of the images is to make use of the multimedia effect, by combining both text and images onto the demonstration. This breaks up the page, preventing too much text being displayed at once.

4.4 – HTML, JavaScript, JQuery and JQueryUI



▾ What is an Array?

Technical term: An array is a data structure that is used to store a collection of elements. These elements will be of the same data type, such as strings or integers. Each element takes up one space in the array.

Basic term: Think of it like having a set of boxes. You can use these boxes to store stuff in.
There are only two limitations:
1) Each box can only hold one item.
2) The boxes can only store one type of item.

The picture above shows two groups of three boxes (otherwise known as two arrays containing three elements).

▸ What's their purpose?

▸ How do they work?

▸ How do they work in a computing sense?

▸ Key words and definitions

▸ Array Declaration: Java Examples

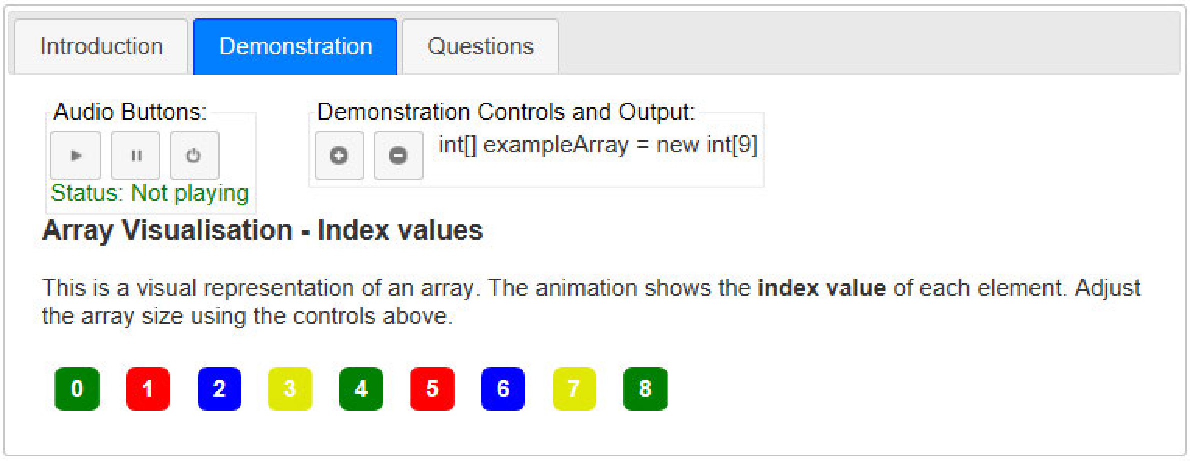
▸ Array Interaction: Java Examples

Figure 4.4.1 – Introduction Form.

The second half of this project is the HTML, JavaScript, JQuery and JQueryUI elements. The first page of the application is the introduction form. The actual layout of the page remained similar to what was outlined in the wireframe designs. The only presentational change was the additional heading names that were added after evaluating how much detail the array element had to be covered using.

This page makes use of an informal language structure to talk the user through the features of an array, similarly to the audio recording. All of the text is hard-coded into the HTML page to help with formatting and positioning. The accordion tab style comes from the JQuery UI library, allowing small amounts of information to be displayed at once.

There are a few important features to this page. The first feature is that the accordion structure makes use of the segmentation principle to break up information into smaller, more readable blocks of text. Secondly, the tabs are intuitive to work through because of the portrait layout, which provides a natural reading order for the user to follow. The third feature is that text is kept compact and segmented, creating groups of information that are easier for the user to process. It also makes use of images with labels to provide a visual break for the reader as they progress down the page. Finally, the colour choice draws the users' attention towards the important elements on the page.



int[] exampleArray = new int[9]

Array Visualisation - Index values

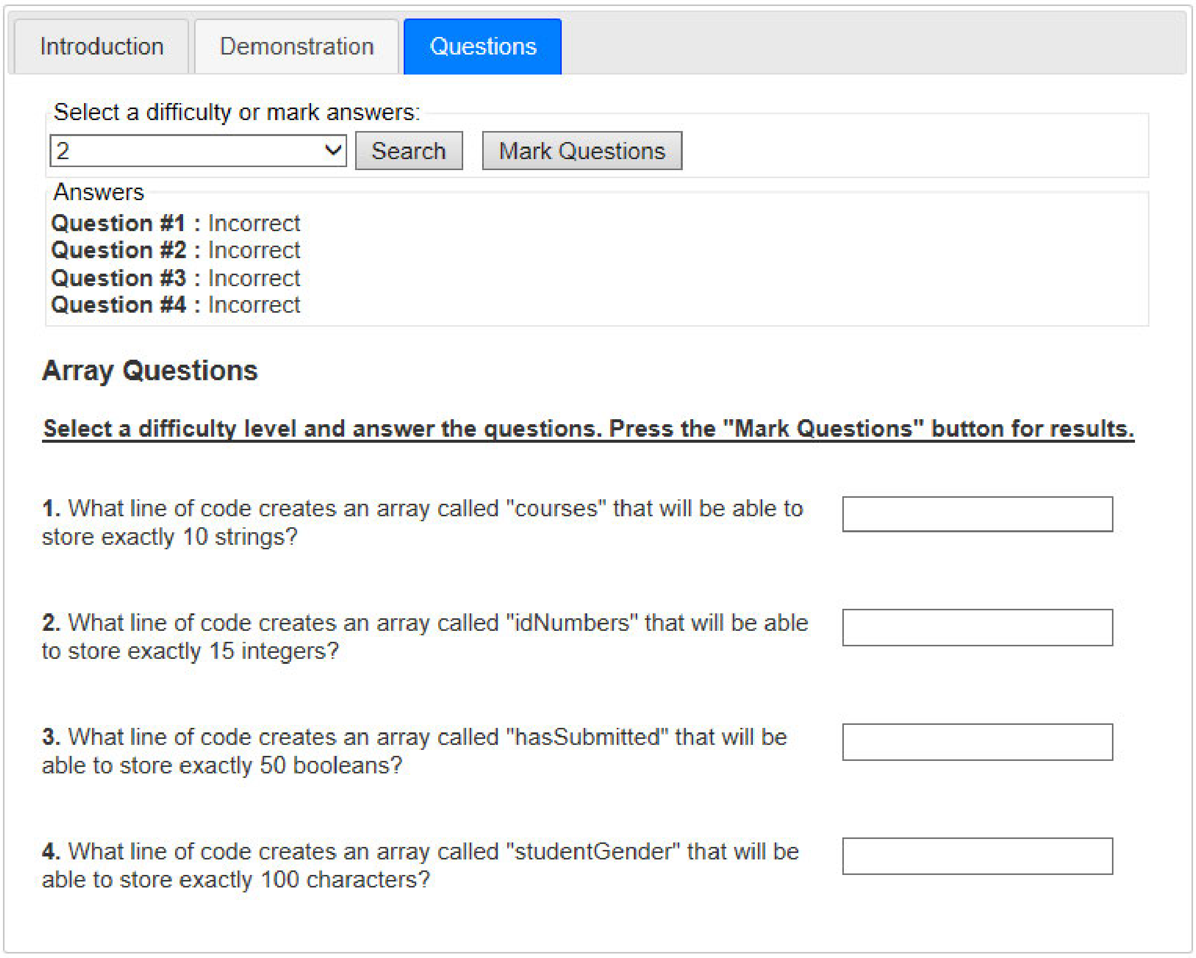
This is a visual representation of an array. The animation shows the **index value** of each element. Adjust the array size using the controls above.

Figure 4.4.2 – Demonstration Form.

The demonstration form has two key elements. The first is the audio capability that allows the user to listen to a description of the demonstration as the use it. This makes use of the redundancy and modality principles, encouraging the user to both listen and experiment with the demonstration.

The second element of this page is the demonstration itself. By using the buttons at the top of the page, the user can experiment with adjusting the size of the array. The Java equivalent code is displayed alongside the controls to show the user what code would create that particular array. The demonstration makes use of vibrant colours to draw attention towards each element. Inside each element is the Index location.

In order to function, the HTML page is linked to a Data.js file that contains all of the relevant functions. The demonstration page makes use of two groups of JQuery functions. Each function is called when a button element is pressed.



Select a difficulty or mark answers:

2

Answers

Question #1 : Incorrect
Question #2 : Incorrect
Question #3 : Incorrect
Question #4 : Incorrect

Array Questions

Select a difficulty level and answer the questions. Press the "Mark Questions" button for results.

1. What line of code creates an array called "courses" that will be able to store exactly 10 strings?

2. What line of code creates an array called "idNumbers" that will be able to store exactly 15 integers?

3. What line of code creates an array called "hasSubmitted" that will be able to store exactly 50 booleans?

4. What line of code creates an array called "studentGender" that will be able to store exactly 100 characters?

Figure 4.4.3 – Questions Form.

The final page of the application is the Questions page. This makes use of Ajax calls to the Java MVC model to populate the questions div container. Each question is generated dynamically based on what the user choses using the difficulty selector. The user inputs their answers into the text boxes and clicks on the “Mark Questions” button. An answers div is then created, which informs the user what questions they got correct or incorrect.

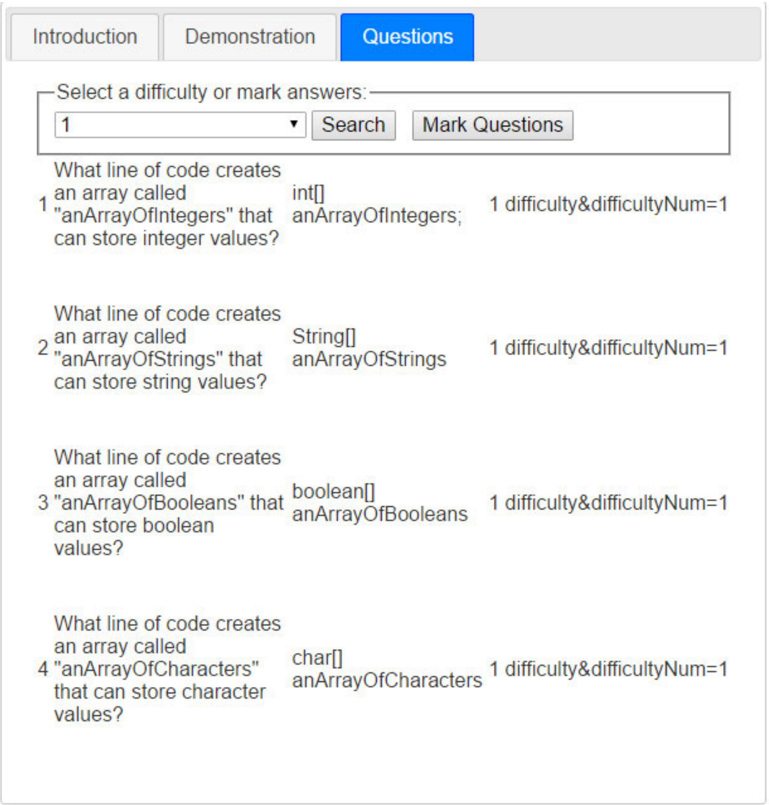
Design changes were kept to a minimum due to issues with the formatting from the dynamic generation of the table. A large amount of time was spent trying to position these elements correctly.

Chapter 5 - Product Evaluation

5.1 – Evaluation Overview

This section will describe some of the testing performed, along with examples of problems that were encountered during the development process. The majority of problems encountered during the application creation stemmed from the CSS and HTML files. This meant that many issues were corrected whilst the code was still being written. Below are some of the larger issues that were encountered.

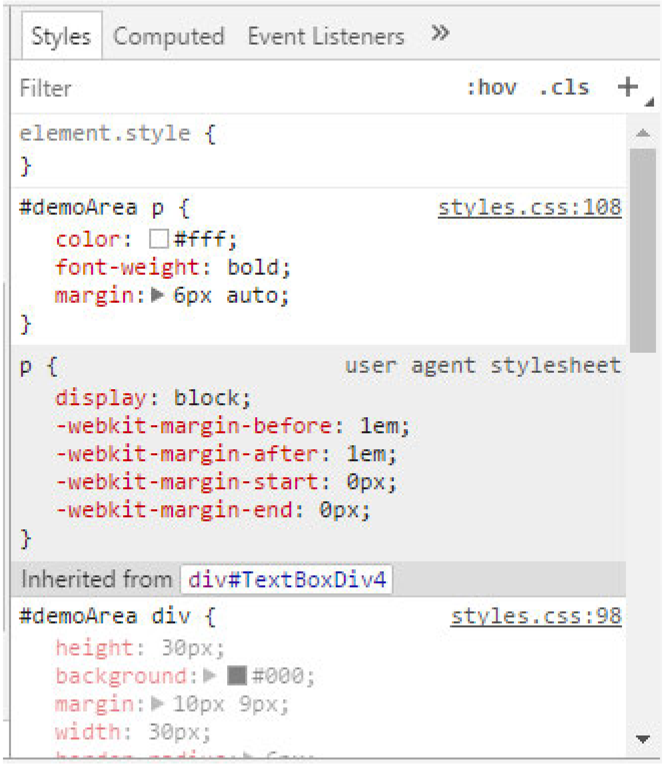
5.2 – Testing



1	What line of code creates an array called "anArrayOfIntegers" that can store integer values?	int[] anArrayOfIntegers;	1 difficulty&difficultyNum=1
2	What line of code creates an array called "anArrayOfStrings" that can store string values?	String[] anArrayOfStrings	1 difficulty&difficultyNum=1
3	What line of code creates an array called "anArrayOfBooleans" that can store boolean values?	boolean[] anArrayOfBooleans	1 difficulty&difficultyNum=1
4	What line of code creates an array called "anArrayOfCharacters" that can store character values?	char[] anArrayOfCharacters	1 difficulty&difficultyNum=1

Figure 5.2.1 – Question Array Formatting.

The biggest issue encountered was the formatting of the questions. When the questions are requested using the Ajax call, a collection of JSON objects are returned containing multiple variables. The number of questions returned can change depending on the difficulty requested. That means that the textboxes and display structure need to be generated dynamically after each request. The image below shows how much information needed to be processed an organised before the textboxes were generated. The main issue encountered was displaying the right question in the right location, whilst also being able to track which textbox needed to match that specific question.



```
element.style {
}
#demoArea p {                                    styles.css:108
   color: □#fff;
   font-weight: bold;
   margin:▶ 6px auto;
}
p {                                        user agent stylesheet
   display: block;
   -webkit-margin-before: 1em;
   -webkit-margin-after: 1em;
   -webkit-margin-start: 0px;
   -webkit-margin-end: 0px;
}
Inherited from div#TextBoxDiv4
#demoArea div {                                   styles.css:98
   height: 30px;
   background:▶ ■#000;
   margin:▶ 10px 9px;
   width: 30px;
```

Figure 5.2.2 – Console CSS.

As mentioned previously, CSS provided the largest number of issues and consumed the most amount of time out of this entire project. The reason for that is that it requires the tweaking and adjusting every single element as it was added to the page. Considering how important the visual aspect of this project was, it was vital that all elements aligned correctly and interacted in a smooth way.

Chapter 6 – Conclusion

6.1 - Project Review

The objective of this project was to create an interactive tutoring application that would teach a novice user a fundamental computing concept. This objective was completed and a final web application has been created.

6.2 – Project Limitations

The main limitation of this project was that only one concept was covered and explained. It would have ideal to cover more fundamental computing topics such as stacks, queues or pointers. The application completes its intended purpose; however, it would have been interesting to see how it could have been adapted to display additional concepts.

Another limitation was in regards to the choice and familiarity of the languages. Over the course of this project, I had to learn how to create an application using a completely new set of programming languages. This required a large amount of experimentation and some compromises in regards to some of the applications features.

6.3 – Potential Improvements

One feature that would have been ideal to include in this project would be the ability to compile and run code inside the application. This compiler would allow the user to freely experiment with coding from inside the program, instead of having to use another piece of software. It would also be able to provide syntactical checking, which is one of the most useful features when beginning to code. Coding a complier from scratch is a very difficult task and would have taken far too long to complete.

A more realistic improvement would be the addition of syntactical checking when using the question form. Providing detailed feedback about the user's inputs would allow them to learn from their mistakes in a more convenient way. Faster and accurate feedback would increase the overall effectiveness of the application.

6.4 – Project Summary

This project required a large amount of research into the field of e-learning and experimentation with unfamiliar programming languages. Taking the time to do all of this has allowed the completion of a working application that performs its required task in an efficient way, whilst also having potential for further expansion through the incorporation of additional computing concepts.

Overall, the project was successful at completing its intended purpose of teaching novice users about using a fundamental computing function in Java. It has a variety of different interactive features that enable the user to get instant feedback of their understanding of the topic. It has made use of all the principles researched during the literature review, showing that it takes a variety of learning styles into account. The techniques used to explain arrays could easily be replicated and modified to suit any other data structure or computing concept.

Reference List

Allinjawi.A and Al-Nuaim.H (2014). *An Achievement Degree Analysis Approach to Identifying Learning Problems in Object-Oriented Programming*. King Abdulaziz University.

Bennedsen.J and Caspersen M. E (2007). *Failure Rates in Introductory Programming*, IT University West, Aarhus V, Denmark: ACM SIGCSE Bulletin, p.32-36.

Clark, R. C, Mayer, R. E. (2011), *E-Learning and the Science of Instruction*, 3rd edition, John Wiley & Sons, San Fransisco.

Hegarty, M. (2004). *Dynamic visualizations and learning: Getting to the difficult questions*. Learning and Instruction, 14, p. 345.

Hoffman, B. (2006). *The Encyclopedia of Educational Technology*. San Diego, CA: Montezuma Press.

Hrastinski, S. (2007). *Participating in synchronous online education*. Unpublished doctoral dissertation, Lund University, Germany.

Hrastinski, S. (2008). *Asynchronous and synchronous e-learning*. EDUCAUSE Quarterly, Issue 4, p.51–55.

Mayer, R. E., Bove, W., Bryman, A., Mars, R., & Tapangco, L. (1996). *When less is more: Meaningful learning from visual and verbal summaries of science textbook lessons*. Journal of Educational Psychology*, Vol 88*, University of California, Santa Barbara, p.64–73.

Mayer, R. E., Gallini, J.K. (1990). *When is an illustration worth ten thousand words?* Journal of Educational Psychology*, 88, p.64 – 73.*

Mayer, R.E. (Ed.). (2005). *The Cambridge handbook of multimedia learning,* New York: Cambridge University Press, P.184

Milne.I and Rowe.G. (2002). *Difficulties in Learning and Teaching Programming – Views of Students and Tutors*. University of Dundee, Dundee: Kluwer Academic, p.58.

Ollerenshaw, A., Aidman, E., & Kidd, G. (1997). *Is an illustration always worth ten thousand words? Effects of prior knowledge, learning style, and multimedia illustrations on text comprehension*. International Journal of Instructional Media, 24, p.227–238.

Sorden S. D. (2005). *A cognitive approach to instructional design for multimedia learning*. Information Science Journal, 8(1), p. 263-278.

Tindall-Ford, S., Chandler, P., & Sweller, J. (1997). *When two sensory modes are better than one*. Journal of Experimental Psychology*: Applied, 3,* p. 257-286

Bibliography

JQuery Library:

http://jquery.com/

JQuery API:

http://api.jquery.com/

JQuery UI Library:

https://jqueryui.com/

JQuery UI API:

http://api.jqueryui.com/

GSON Library:

http://repo1.maven.org/maven2/com/google/code/gson/gson/2.8.0/

Dynamic textbox creation tutorial:

http://www.mkyong.com/jquery/how-to-add-remove-textbox-dynamically-with-jquery/

CSS tutorial:

https://www.w3schools.com/css/

Appendices

Appendix A – Terms of Reference

Project Title: An Interactive tutor for Programming

Student: [redacted] **Supervisor**: Peter Mckenna

Course-Specific Learning Outcomes

On successful completion of this unit, the student will be able to:

1. Independently plan, manage and successfully complete a project of substantial size in an area that is relevant to their degree programme.
2. Demonstrate that they have the capacity to gain new skills and knowledge independently of teaching.
3. Critically reflect and evaluate existing work and their own work.
4. Integrate the learning obtained from other units taken on their degree programme. (MMU,2016)
5. Investigate the interaction between hardware and software and the influence of this interaction on the design of computer systems. (Costen,N., 2016)

Project Background

Programming as a skill set is vital to the growth of new technology and software used in all fields of work. The development skills brought by computer scientists who specialize in this area are essential to the development of modern businesses. In recognition of this, the frequency of students learning how to program is constantly increasing at all levels of education. Prospective programmers will start with primary concepts to act as a foundation, allowing them to expand and learn new information to suit their future requirements.

Even at the most basic of levels, programming contains a substantial number of topics. Learning new concepts requires a strong foundation of core programming terms and functions. Many students believe that they understand certain topics to a high level. However, findings from a report that investigated some of the difficulties encountered when learning to program states that "upon detailed examination or one-to-one querying from a tutor it turns out the they are often wrong in their belief" (Milne.I and Rowe.G., 2002). This shows that although people may believe they have enough knowledge in regards to a particular subject, they actually only have a limited overview and require alternative methods of concept representation to fully understand how to efficiently use and explain it.

This project will consist of researching, developing and testing a piece of interactive software that will assist users when learning a specific core programming topic. This could act as a supplementary learning resource or as a completely independent program to teach a particular topic that novice programmers may struggle to completely understand. The program will have to make use of an array of techniques to be effective due to the variety of different learning styles and teaching methods that currently exist.

Aim

To teach a fundamental programming topic in an intuitive and beginner friendly way through the creation of an interactive application.

Objectives

- Explore existing research into teaching and learning techniques for both programming and non-programming topics (books and journal articles from online library resources).
- Determine a relevant area of programming to focus the development of the application on by using results from previous research and investigations.
- Outline a program design using influence from previous research to ensure that the application will be optimal for its intended purpose.
- Evaluate the effectiveness of the completed application by collecting and comparing the results of users interacting with it.
- Produce a summary upon completion of the project to present results and offer suggestions for future related work.

Deliverables

Over the course of this project, a variety of documentation will be compiled to track the development of the application. This will include:

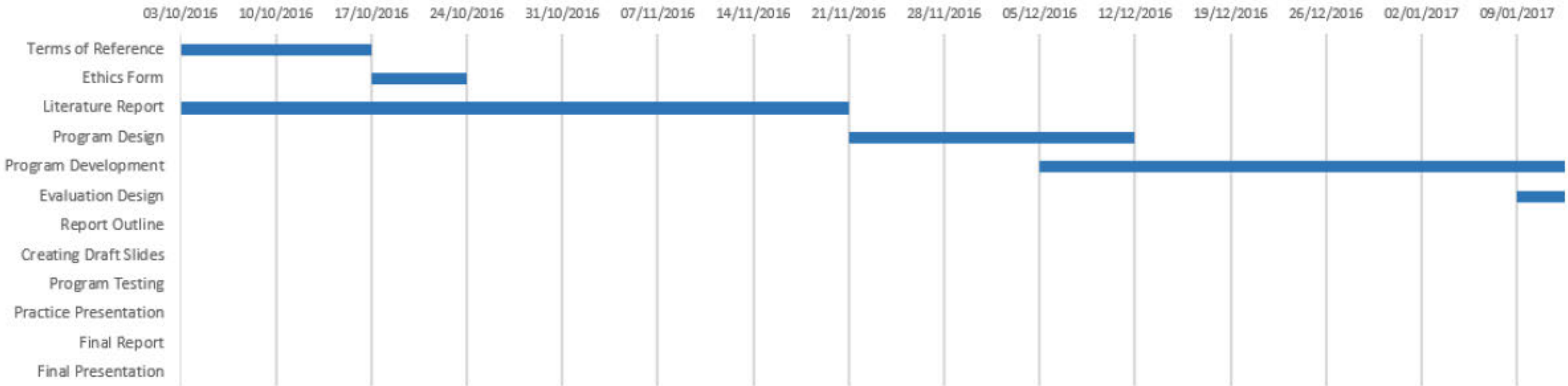
- Terms of Reference: A document containing course learning outcomes, a description of the project, a list of possible issues, a structured schedule, aims and objectives.
- Literature Review: A report containing relevant and reputable literature research, consisting predominantly of information related to a similar problem.
- Final Application: The fully completed, working application.
- Presentation and Slides: A demonstration of the application and evidence of important findings obtained from the project.
- Final Project Report: The final report containing all work and research completed over the course of the project.

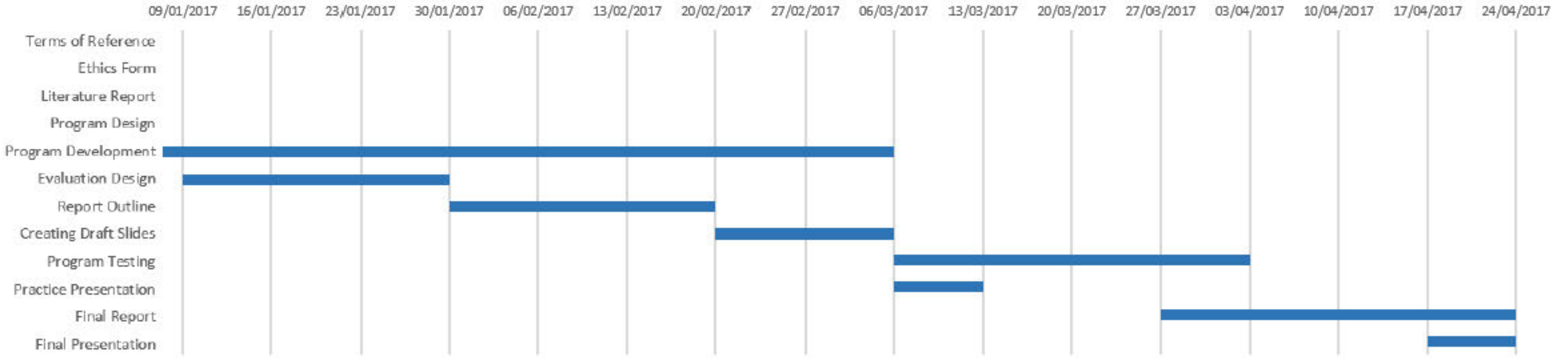
Problems

- Topic complexity: Finding the correct balance of topic choice. Selecting a topic that is difficult to grasp but not too complex that the user has to use additional methods of learning to gain full use out of the application.
- Choice of programming language: Establishing that the chosen programming language has sufficient tools and features to allow the creation of a suitable application.

Schedule

Task	Duration (Weeks)	Start Date	End Date
Terms of Reference	2	03/10/2016	17/10/2016
Ethics Form	1	17/10/2016	24/10/2016
Literature Report	7	03/10/2016	21/11/2016
Program Design	3	21/11/2016	12/12/2016
Program Development	13	05/12/2016	06/03/2017
Evaluation Design	3	09/01/2017	30/01/2017
Report Outline	3	30/01/2017	20/02/2017
Creating Draft Slides	2	20/02/2017	06/03/2017
Program Testing	4	06/03/2017	27/03/2017
Practice Presentation	1	06/03/2017	13/03/2017
Final Report	4	27/03/2017	24/04/2017
Final Presentation	1	17/04/2017	24/04/2017

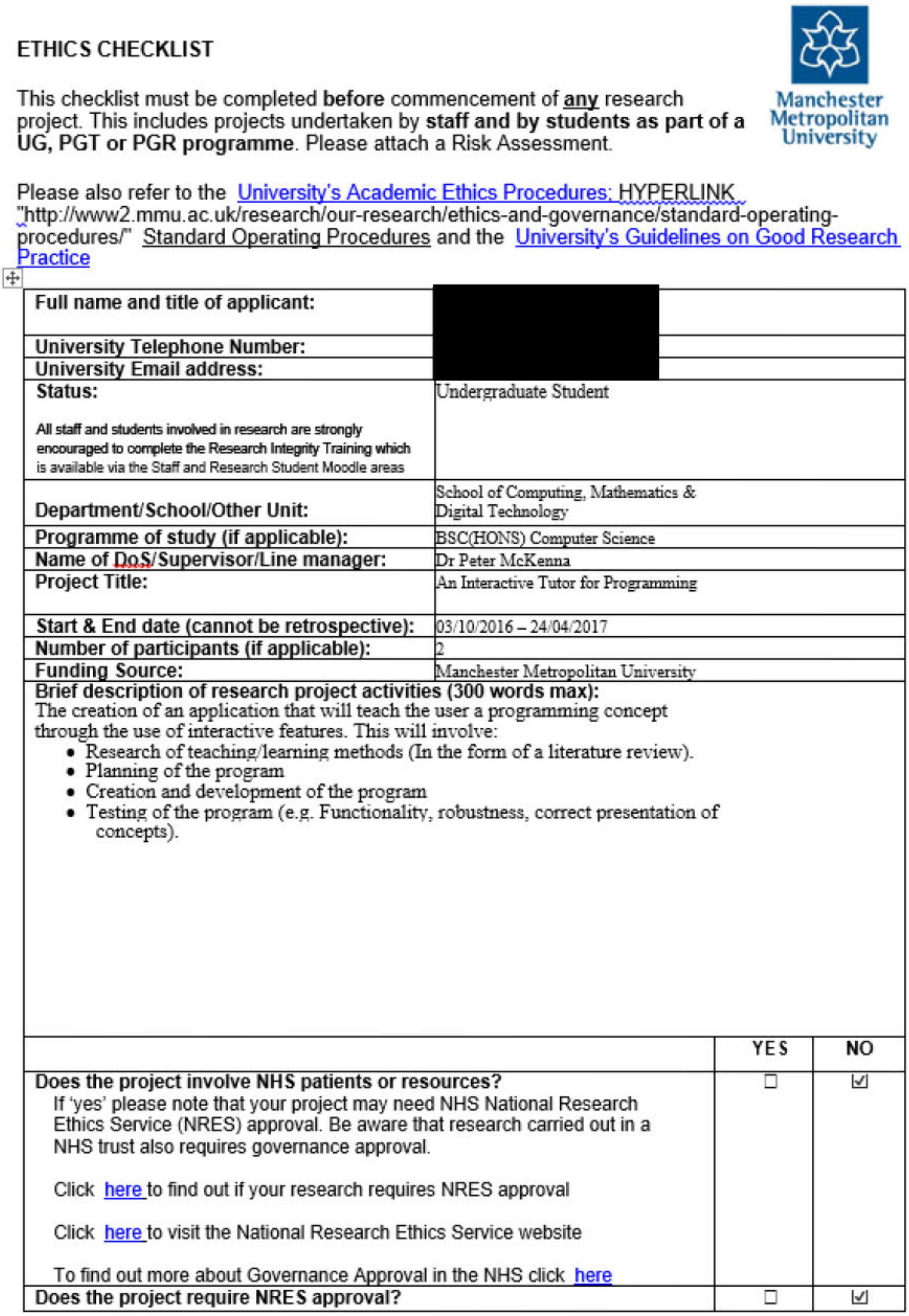




Required Resources:

- Computer with Eclipse Java IDE
- Web Browser
- Access to online library resources (MMU library portal, Google scholar)
- Text editors (Microsoft word, Google documents)

Appendix B – Ethics form



ETHICS CHECKLIST

This checklist must be completed **before** commencement of **<u>any</u>** research project. This includes projects undertaken by **staff and by students as part of a UG, PGT or PGR programme**. Please attach a Risk Assessment.

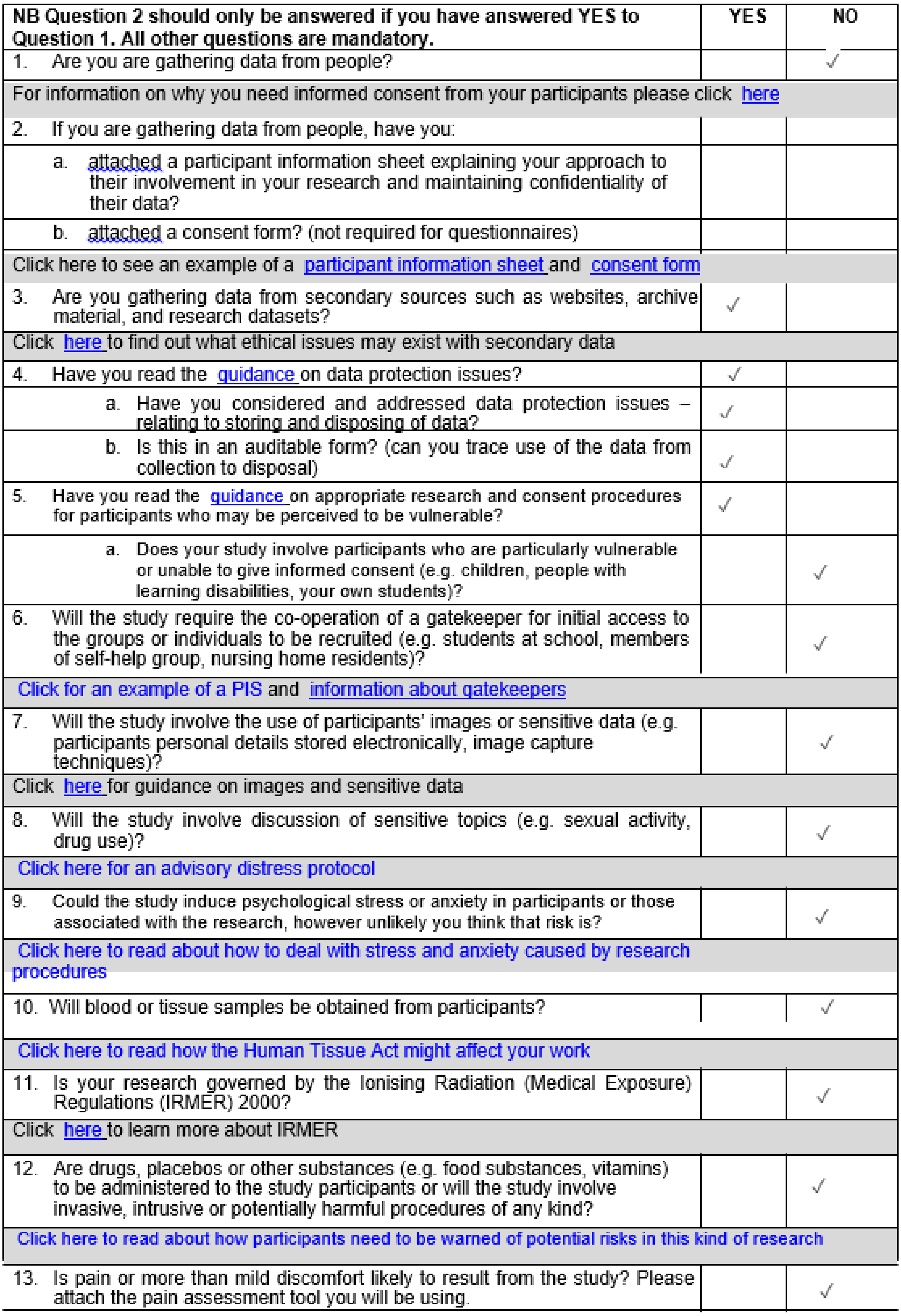
Please also refer to the University's Academic Ethics Procedures: HYPERLINK "http://www2.mmu.ac.uk/research/our-research/ethics-and-governance/standard-operating-procedures/" Standard Operating Procedures and the University's Guidelines on Good Research Practice

Full name and title of applicant:	[redacted]
University Telephone Number:	[redacted]
University Email address:	[redacted]
Status: All staff and students involved in research are strongly encouraged to complete the Research Integrity Training which is available via the Staff and Research Student Moodle areas	Undergraduate Student
Department/School/Other Unit:	School of Computing, Mathematics & Digital Technology
Programme of study (if applicable):	BSC(HONS) Computer Science
Name of DoS/Supervisor/Line manager:	Dr Peter McKenna
Project Title:	An Interactive Tutor for Programming
Start & End date (cannot be retrospective):	03/10/2016 – 24/04/2017
Number of participants (if applicable):	2
Funding Source:	Manchester Metropolitan University

Brief description of research project activities (300 words max):
The creation of an application that will teach the user a programming concept through the use of interactive features. This will involve:

- Research of teaching/learning methods (In the form of a literature review).
- Planning of the program
- Creation and development of the program
- Testing of the program (e.g. Functionality, robustness, correct presentation of concepts).

	YES	NO
Does the project involve NHS patients or resources? If 'yes' please note that your project may need NHS National Research Ethics Service (NRES) approval. Be aware that research carried out in a NHS trust also requires governance approval. Click here to find out if your research requires NRES approval Click here to visit the National Research Ethics Service website To find out more about Governance Approval in the NHS click here	□	☑
Does the project require NRES approval?	□	☑



NB Question 2 should only be answered if you have answered YES to Question 1. All other questions are mandatory.	**YES**	**NO**
1. Are you are gathering data from people?		✓
For information on why you need informed consent from your participants please click here		
2. If you are gathering data from people, have you:		
a. attached a participant information sheet explaining your approach to their involvement in your research and maintaining confidentiality of their data?		
b. attached a consent form? (not required for questionnaires)		
Click here to see an example of a participant information sheet and consent form		
3. Are you gathering data from secondary sources such as websites, archive material, and research datasets?	✓	
Click here to find out what ethical issues may exist with secondary data		
4. Have you read the guidance on data protection issues?	✓	
a. Have you considered and addressed data protection issues – relating to storing and disposing of data?	✓	
b. Is this in an auditable form? (can you trace use of the data from collection to disposal)	✓	
5. Have you read the guidance on appropriate research and consent procedures for participants who may be perceived to be vulnerable?	✓	
a. Does your study involve participants who are particularly vulnerable or unable to give informed consent (e.g. children, people with learning disabilities, your own students)?		✓
6. Will the study require the co-operation of a gatekeeper for initial access to the groups or individuals to be recruited (e.g. students at school, members of self-help group, nursing home residents)?		✓
Click for an example of a PIS and information about gatekeepers		
7. Will the study involve the use of participants' images or sensitive data (e.g. participants personal details stored electronically, image capture techniques)?		✓
Click here for guidance on images and sensitive data		
8. Will the study involve discussion of sensitive topics (e.g. sexual activity, drug use)?		✓
Click here for an advisory distress protocol		
9. Could the study induce psychological stress or anxiety in participants or those associated with the research, however unlikely you think that risk is?		✓
Click here to read about how to deal with stress and anxiety caused by research procedures		
10. Will blood or tissue samples be obtained from participants?		✓
Click here to read how the Human Tissue Act might affect your work		
11. Is your research governed by the Ionising Radiation (Medical Exposure) Regulations (IRMER) 2000?		✓
Click here to learn more about IRMER		
12. Are drugs, placebos or other substances (e.g. food substances, vitamins) to be administered to the study participants or will the study involve invasive, intrusive or potentially harmful procedures of any kind?		✓
Click here to read about how participants need to be warned of potential risks in this kind of research		
13. Is pain or more than mild discomfort likely to result from the study? Please attach the pain assessment tool you will be using.		✓

Click here to read how participants need to be warned of pain or mild discomfort resulting from the study and what do about it.		
14. Will the study involve prolonged or repetitive testing or does it include a physical intervention?		✓
Click here to discover what constitutes a physical intervention and here to read how any prolonged or repetitive testing needs to managed for participant wellbeing and safety		
15. Will participants to take part in the study without their knowledge and informed consent? If yes, please include a justification.		✓
Click here to read about situations where research may be carried out without informed consent		
16. Will financial inducements (other than reasonable expenses and compensation for time) be offered to participants?		✓
Click here to read guidance on payment for participants		
17. Is there an existing relationship between the researcher(s) and the participant(s) that needs to be considered? For instance, a lecturer researching his/her students, or a manager interviewing her/his staff?		✓
Click here to read guidance on how existing power relationships need to be dealt with in research procedures		
18. Have you undertaken Risk Assessments for each of the procedures that you are undertaking?		
19. Is any of the research activity taking place outside of the UK?		✓
20. Does your research fit into any of the following security sensitive categories: • commissioned by the military • commissioned under an EU security call • involve the acquisition of security clearances • concerns terrorist or extreme groups If Yes, please complete a Security Sensitive Information Form		✓

I understand that if granted, this approval will apply to the current project protocol and timeframe stated. If there are any changes I will be required to review the ethical consideration(s) and this will include completion of a 'Request for Amendment' form.

✓ I have attached a Risk Assessment

✓ I have attached an Insurance Checklist

If the applicant has answered **YES** to **ANY** of the questions **5a – 17** then they must complete the MMU Application for Ethical Approval.

Signature of Applica[redacted]____Date: 24/10/2016 (DD/MM/YY)

Independent Approval** for the above project is (please check the appropriate box):* ***Granted

☐ I confirm that there are no ethical issues requiring further consideration and the project can commence.

Not Granted

☐ I confirm that there are ethical issues requiring further consideration and will refer the project protocol to the Faculty Research Group Officer.

Signature: ____________________________________ Date: _____________ (DD/MM/YY)

Print Name: ___________________________________ Position: ______________________

Approver: Independent Scrutiniser for UG and PG Taught/ PGRs RD1 Scrutiniser/ Faculty Head of Ethics for staff.

Appendices References

Costen,N. (2016). *MMU Course-Specific Unit Outcomes*. [online] Available at: https://moodle.mmu.ac.uk/pluginfile.php/2115169/mod_resource/content/1/6Z1001_CourseOutcomes_2016_v1.pdf [Accessed 10 Oct. 2016].

Milne.I and Rowe.G. (2002). *Difficulties in Learning and Teaching Programming – Views of Students and Tutors*. University of Dundee, Dundee: Kluwer Academic, p.58.

MMU. (2016). *MMU Unit Specification*. [online] Available at: https://moodle.mmu.ac.uk/pluginfile.php/2092744/mod_resource/content/1/6G6Z1001%20Unit%20Specification.pdf [Accessed 10 Oct. 2016].