Edge Hill University

CIS2301 – Server and Client Side Scripting

BSc Web Design & Development

Server and Client Side Scripting

Author:
Jacob Cleaver

Tutor:
Dave Walsh

May 5, 2016

Contents

1 Introduction **4**

2 System Planning and Research **4**

2.1 Literature Review 5

2.2 Desk Research 7

2.3 System Modelling 10

2.3.1 Use-Case Scenarios 11

2.3.2 Use-Case Diagrams 12

2.3.3 Class Diagram 13

2.3.4 ERD Diagram 14

3 Technology Choices **16**

4 Software Development Process **17**

5 Application Testing **17**

6 Questionnaire Design **20**

7 Project Post-Mortem **24**

7.1 Positives 24

7.2 Negatives 25

7.3 Improvements 26

8 Appendices **29**

8.1 Appendix A - UML Diagram Planning 29

8.2 Appendix B - Low Fidelity Prototype 1 34

8.3 Appendix C - Low Fidelity Prototype 2 39

List of Tables

1 Desk Research carried out focussing on the top three most popular survey websites (Survey Monkey, Typeform and Google Forms) according to (Marrs, 2014). 10

2 Application testing carried to ensure the system functions as intended, reducing and mending all the errors that can occur during this process 20

List of Figures

1 This diagram demonstrates how each user will interact with the web-based system. 13
2 Here is a class diagram focusing on the database infrastructure. 14
3 This entity relationship diagram focuses on the relationships throughout the database including the primary and foreign keys. 15
4 This shows the list of five questions created for the survey. . . 21
5 This shows the responses the respondents provided for the first question. 22
6 Responses for question two of five. 22
7 Here are the responses the respondents provided for the third question. 23
8 These are the responses for question four. 23
9 This shows the responses the respondents provided for the fifth question. 24
10 Stage one of the use-case diagram implementation. 29
11 This has been defined and condensed which was used to create the digitalised version. 30
12 This is the planning for the class diagram showing the database structure. 31
13 Stage one of the ERD diagram. 32
14 Finalising the ERD ready for digitalising. 33
15 This images shows the first design of creating a survey. 34
16 This design is for the adding of questions to the survey. 35
17 This is a list of all the user's surveys they have created. 36
18 This shows the responses gathered from the respondents. . . . 37
19 Here is the front-end of the survey that the respondents will access via the URL web link. 38
20 This entity relationship diagram focuses on the relationships throughout the database including the primary and foreign keys. 39

21 This is the finalised design of the My Surveys page for the Admin user to create, view and see responses for all their surveys. 40
22 This is the create survey/questionnaire form the user will complete. 41
23 Here is the dialogue confirmation box the Admin user will be faced with when they are trying to delete a survey. 42
24 This image displays how the Admin user will see the responses initiated from the respondents responses to the questions. . . . 43
25 This is final design of the front-end view for the respondents to answer the question. 44

Abstract

Questionnaires have allowed for broader and specified data gathering based on desired demographics. This report will be studying questionnaires gathering research on the benefits and optimised features of questions to assemble the best results. Once the analysis of research has been discussed the web-based questionnaire system will be planned using UML diagrams and then built using a Laravel framework. Application testing will be used to improve the build before the final test is carried out; creating an optimised questionnaire and sending to a small sample to receive genuine results.

1 Introduction

Questionnaires have advanced the gathering of data to be used in many ways with the overall aim being improvement. This report will be in alliance with a Laravel project studying questionnaires and various survey websites to produce an optimised web-based questionnaire system extracted from the research gathered within this report. The report will consist of several stages and areas covering everything from the planning and research to the technology choices and an overall project post-mortem. Once the documentation and the system has been completed, five questionnaires will be produced by the Admin user and it will be sent to five respondents to gather a small variety of actual data that can be analysed from the response data built into the system.

2 System Planning and Research

Planning and research is vital to providing efficient, effective and complex web-based systems. This section will involve a literature review focussing on questionnaires/surveys; looking into how they are produced, effectiveness, essential and optimal features, benefits and drawbacks. In addition, desk research will be included comparing and contrasting the features from three popular survey websites. Once the research is fully understood the planning can take place in the form of Unified Modelling Language (UML) diagrams such as; use-case, class and entity-relationship diagrams.

2.1 Literature Review

According to (KirkleesCouncil, 2015) "a questionnaire is simply a 'tool' for collecting and recording information about a particular issue of interest. (BusinessDictionary, 2016) furthers this by saying a questionnaire is a "list of a research or survey questions asked to respondents, and designed to extract specific information." (BusinessDictionary, 2016) also stated that questionnaires serve four basic purposes: (1) collecting appropriate data, (2) making data comparable and amendable to analysis, (3) minimising bias in formulating and asking questions, (4) making questions engaging and varied.

There are many benefits to questionnaires such as being able to gather a wide array of data at a relatively low cost (KirkleesCouncil, 2015). However, with the benefits also comes drawbacks. The next stage of the research will be examining the pros and cons of questionnaires. In addition to the benefit mentioned previously, questionnaires are widely beneficial due to the ability to send/share/spread across a wide geographical area including remote locations. This allows for respondents in many different situations to share their interests and opinions. Some people may see questionnaires as a hassle, especially over the telephone, but this capability supplies disabled people or people without internet access with the chance to answer the surveys that may concern themselves. There is also another aspect that needs to be considered when creating questionnaires; selecting the correct demographics. This can be accomplished by handing the surveys out in person giving the surveyor the choice of who to hand it to. As the internet has evolved, survey websites now supply the option to choose the target audience, this can be seen in Pollfish's survey website (Pollfish, 2016). Furthermore, (Google, 2016) provided the information that online surveys can provide researchers with the ability to mass share via many different platforms such as email, URL links, embedded HTML in websites or social media such as; Facebook, Google+ and Twitter.

However, questionnaires do have negatives that need to be discussed to provide a complete overview. The main drawback to questionnaires is the stereotype they have attached, being a hassle and wasting the respondents time. This could lead to the high refusal rate of telephone questioning as respondents may not want to spend their time answering over the phone especially if they are asked by different callers multiple times in one day. Another drawback, this time of face-to-face questioning is it requires trained interviewers to ensure the questions are asked correctly and not leading the respondents to any specific answer creating bias data. This also requires higher costs due to the time consuming and labour intensive nature behind the method.

Creating questionnaires or surveys requires a sturdy thought and planning process to ensure the design is optimised to gather the maximum amount of reliable data. In order for questionnaires to produce optimum results providing effective results the design, content and layout are vital. After studying hundreds of methodology textbooks that offered various versions of conventional wisdom about optimal question design (Krosnick and Presser, 2009) summarised the content into the following information. The first thing to work on is the wording because this will effect the way the questions are answered which in turn will supply false or accurate data depending. Therefore, it is necessary to ensure that technical terms and slang are avoided and to instead use wording that all respondents will interpret in the same way. Also, it needs to be kept in mind that loaded or leading questions will provide biased data due to the nature of the question pushing respondents towards a certain answer. Another way to avoid confusion with the respondents answer is to avoid the use of double entendre questions, sticking to the point will result in definite answers.

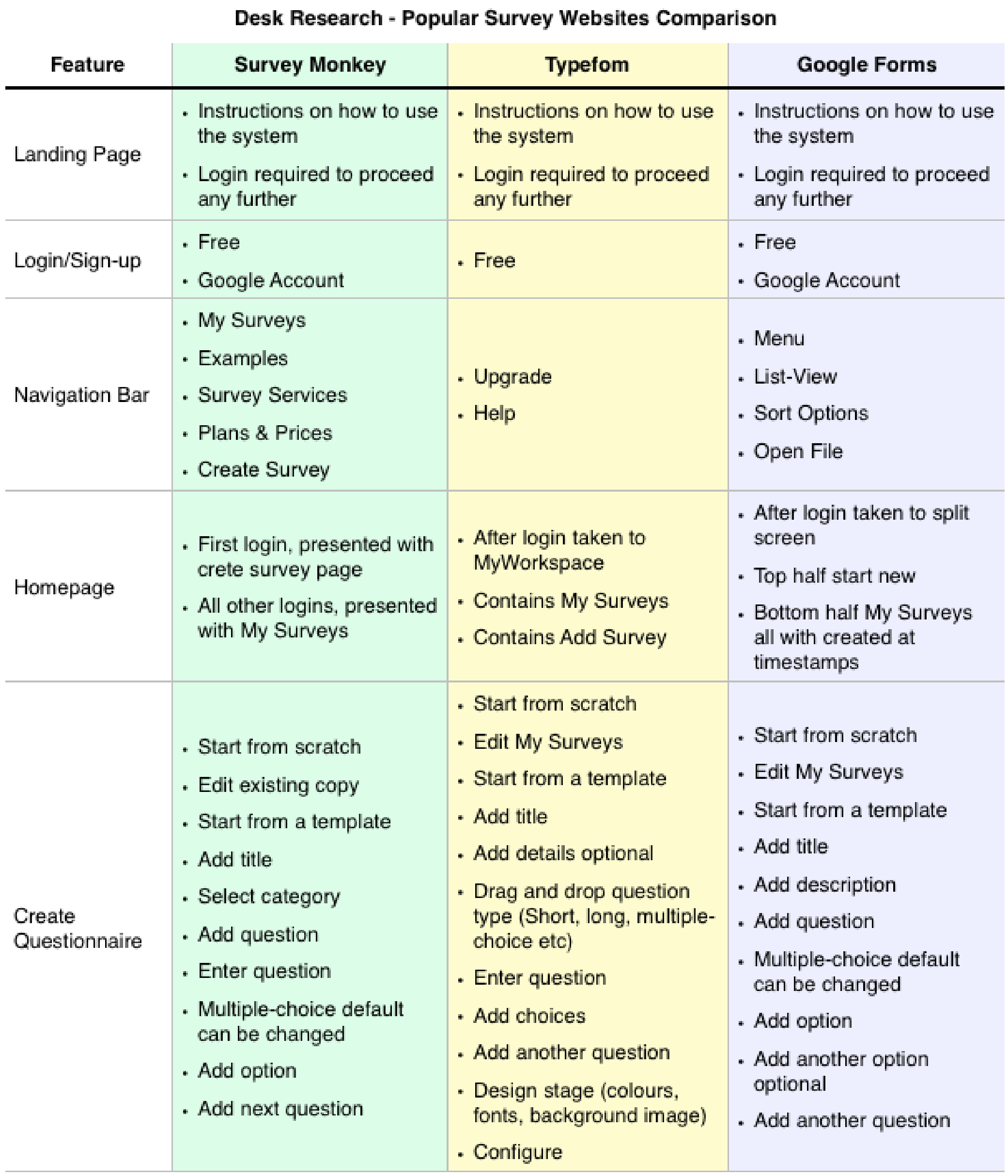
It is crucial to optimise the questionnaire length and question order, focusing on the users interest, availability of time and attention span. According to (Leung, 2001) 'there are no universal agreements about the optimal length of questionnaires'. However, it was found during a BMJ study that 'short simple questionnaires usually attract higher response rates than long complex ones'(Leung, 2001).

Ordering the questions is also vital to entice the respondents and engage them with the questions, to give the best possible results. In an article by (Leung, 2001) it was declared that the question order should begin generalised and end particular to the research area. The reason behind starting general and becoming more specified is false intention, the respondent will begin the questionnaire thinking the questions are easy and simple continuing to answer the remaining questions. Another way (Leung, 2001) says is a clever method of inciting the user throughout the questionnaire in order to gather reliable data is to use semantic differential scales. For example, and answer of 1 being strongly agree through to 5 being strongly disagree. This method might allow for the user to think more about the answer they are giving instead of ticking the same box.

After gathering the information present within this literature review, it is clear to see the main features that need to be included to ensure the survey system is optimised as much as possible.

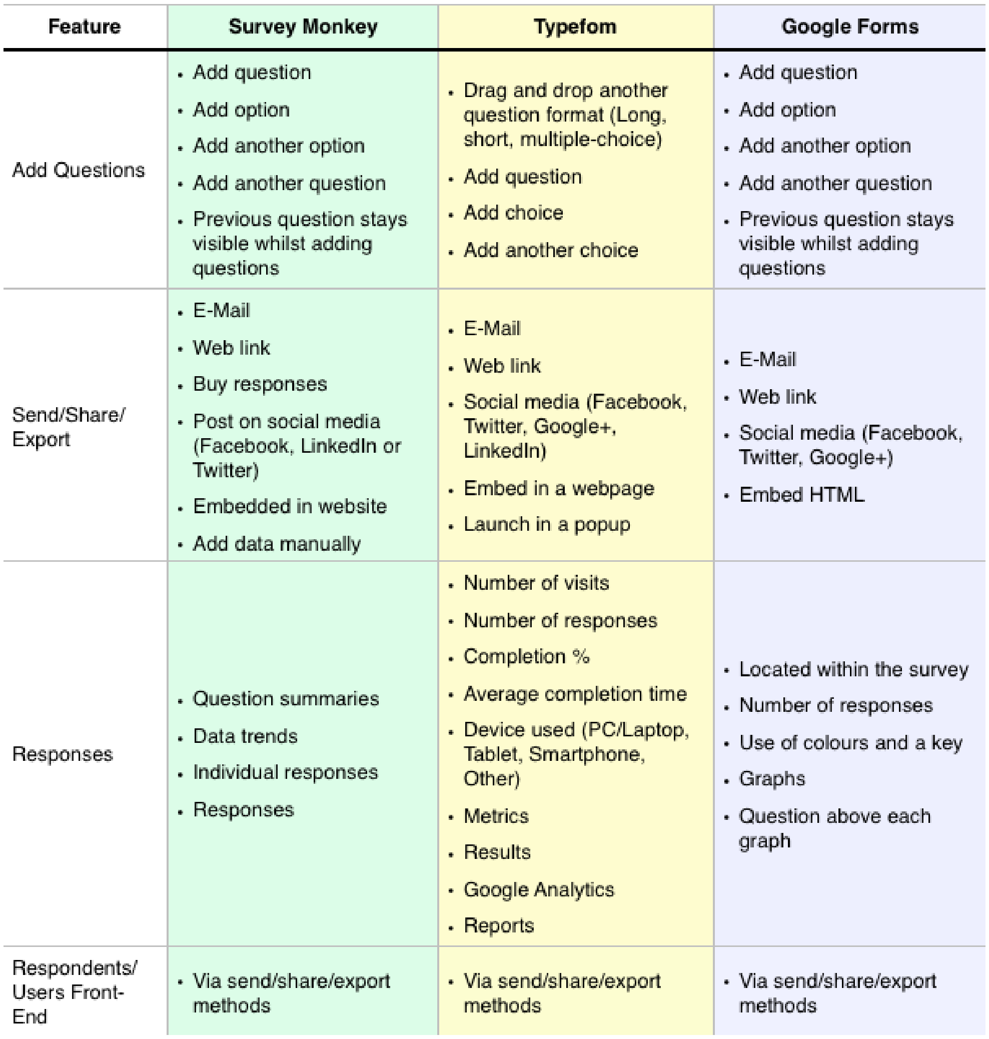
2.2 Desk Research

Desk Research is a favourable methods for gathering information and comparing against the opposition, in this case it will be focussed on survey websites.



Desk Research - Popular Survey Websites Comparison

Feature	Survey Monkey	Typefom	Google Forms
Landing Page	- Instructions on how to use the system - Login required to proceed any further	- Instructions on how to use the system - Login required to proceed any further	- Instructions on how to use the system - Login required to proceed any further
Login/Sign-up	- Free - Google Account	- Free	- Free - Google Account
Navigation Bar	- My Surveys - Examples - Survey Services - Plans & Prices - Create Survey	- Upgrade - Help	- Menu - List-View - Sort Options - Open File
Homepage	- First login, presented with crete survey page - All other logins, presented with My Surveys	- After login taken to MyWorkspace - Contains My Surveys - Contains Add Survey	- After login taken to split screen - Top half start new - Bottom half My Surveys all with created at timestamps
Create Questionnaire	- Start from scratch - Edit existing copy - Start from a template - Add title - Select category - Add question - Enter question - Multiple-choice default can be changed - Add option - Add next question	- Start from scratch - Edit My Surveys - Start from a template - Add title - Add details optional - Drag and drop question type (Short, long, multiple-choice etc) - Enter question - Add choices - Add another question - Design stage (colours, fonts, background image) - Configure	- Start from scratch - Edit My Surveys - Start from a template - Add title - Add description - Add question - Multiple-choice default can be changed - Add option - Add another option optional - Add another question



Feature	Survey Monkey	Typefom	Google Forms
Add Questions	• Add question • Add option • Add another option • Add another question • Previous question stays visible whilst adding questions	• Drag and drop another question format (Long, short, multiple-choice) • Add question • Add choice • Add another choice	• Add question • Add option • Add another option • Add another question • Previous question stays visible whilst adding questions
Send/Share/ Export	• E-Mail • Web link • Buy responses • Post on social media (Facebook, LinkedIn or Twitter) • Embedded in website • Add data manually	• E-Mail • Web link • Social media (Facebook, Twitter, Google+, LinkedIn) • Embed in a webpage • Launch in a popup	• E-Mail • Web link • Social media (Facebook, Twitter, Google+) • Embed HTML
Responses	• Question summaries • Data trends • Individual responses • Responses	• Number of visits • Number of responses • Completion % • Average completion time • Device used (PC/Laptop, Tablet, Smartphone, Other) • Metrics • Results • Google Analytics • Reports	• Located within the survey • Number of responses • Use of colours and a key • Graphs • Question above each graph
Respondents/ Users Front-End	• Via send/share/export methods	• Via send/share/export methods	• Via send/share/export methods

Table 1: Desk Research carried out focussing on the top three most popular survey websites (Survey Monkey, Typeform and Google Forms) according to (Marrs, 2014).

Desk research has allowed for a much improved knowledge on popular survey websites. The desk research carried out in Table 1 has produced clear information that displays the similarities and differences between the three websites. The first similarity that was noticed across all three websites was the landing page containing instructions on how that particular website process runs.

In order to proceed any further and carry out tasks registration or login was required. Once logged in there was a significant difference within the navigation bar. Survey Monkey and Google Forms had various buttons to choose between linking to different features, whereas Typeform adopted more of a mobile appearance with a minimal navigation bar containing two buttons; upgrade and help. The rest of the navigation was either within the main container or the drop-down menu.

When creating a survey all three processes contained many similarities, the main feature that stood out was the default setting of multiple-choice answers. According to (Qualtrics, 2016) "This is the most common question type due to its simplicity and ease of use for both the survey creator and the survey taker".

Another similarity highlighted during the desk research was the positioning of questions when more questions are being created. When an additional question was being created the previous question remained above allowing the user to see their current questions and avoid duplication or overlap of questions.

Sufficient similarities such as this allow for the essential design features to be implemented into the system.

2.3 System Modelling

(Green, 2016) says that 'system modelling is an essential tool for understanding interoperability of the components in the system, managing trade-offs and mitigating risks'. In addition, system modelling is vital to ensure the software architecture is understood and prepared producing UML diagram. This section will be focussing on these essential factors, producing use-case scenarios, use-case diagram, class diagram and an entity relationship diagram. The diagrams below will demonstrate the users tasks and capabilities along side the database infrastructure.

2.3.1 Use-Case Scenarios

According to (Elenburg, 2005) 'use cases originated as a requirements modelling technique within the object-oriented (OO) software development community'. Furthermore (Elenburg, 2005) also talked about use cases in the mid 1990s being formalised as part of the Unified Modelling Language (UML) specification. Later, UML became the most-used specification and therefore, the defacto standard for modelling application structure. In this case, use-case scenarios will demonstrate verbally how each user will interact with the system, which will outline all of the required tasks. The two users present for this system are the Admin who acts as the researcher for the questionnaires and the Respondent who interacts with the questionnaires created by the Admin.

Admin/Researcher
As an Admin Researcher I want to be able to login to a secure system which allows more tasks than responding to questionnaires.
As an Admin Researcher I want to be able to create questionnaires.
As an Admin Researcher I want to be able to add questions to the questionnaire.
As an Admin Researcher I want to be able to add options to the questions in the form of multiple-choice.
As an Admin Researcher I want to be able to read the questionnaires.
As an Admin Researcher I want to be able to delete questionnaires including a confirm deletion.
As an Admin Researcher I want to be able to view the responses gathered from the questionnaires answered by the respondents.
As an Admin Researcher I want to be able to send questionnaires via a generated web link.

Respondent
As a Respondent I want to be able to view a survey that has been shared with me via a URL link.

As a Respondent I want to be able to answer the questions in the survey/questionnaire.
As a Respondent I want to be able to submit the answers.

2.3.2 Use-Case Diagrams

Use-case diagrams work hand-in-hand with the use-case scenarios, demonstrating the list of dialogue in a diagram format. The diagrams supply efficient detail including features such as to confirm and delete and also requiring specific users to login before allowing permissions. This then begins the hierarchy of the system building up the authorisation of different users. The diagram produced for the system (Figure 1) contains two actors/users; Admin/Researcher and the respondent. The Admin for the system requires a login to carry out any tasks such as creating surveys, which can be seen to include the creation of questions which includes adding options. Another inclusion the diagram shows is the confirm delete which is included as part of the deleting survey task. On the other hand, the respondent doesn't need as many abilities because as the desk researched showed; all the respondents do not need to login to answer the questions. Therefore, all the respondent needs to be able to do is view surveys, answer questions and submit the answers.

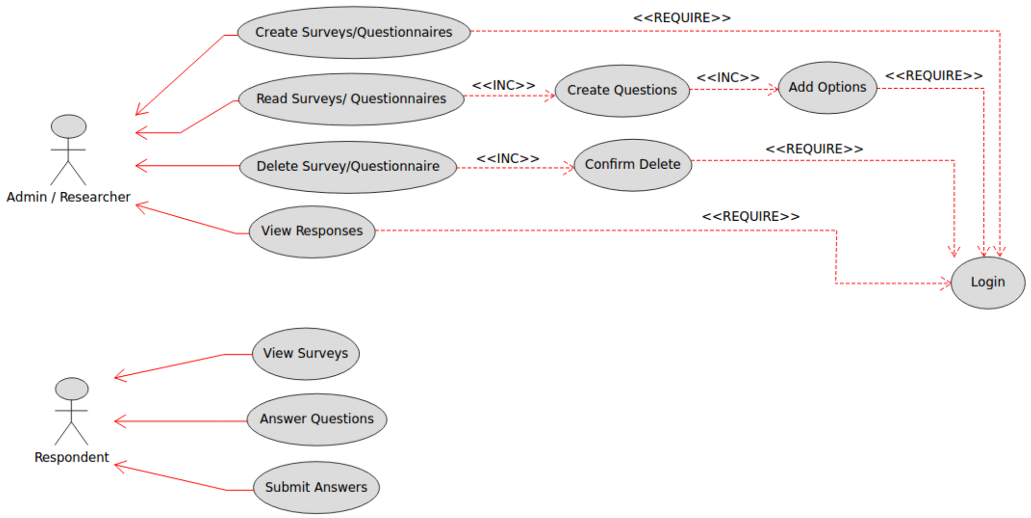


Figure 1: This diagram demonstrates how each user will interact with the web-based system.

2.3.3 Class Diagram

(Kusek et al., 2001) states that 'class diagrams describe and demonstrate a set of classes, interfaces and their relationships between one another showing a view of the system's static design which is very useful in modelling object-oriented systems'. The class diagram seen below in Figure 2 demonstrates a set of four classes and the relationship between each of the classes. The diagram shows the hierarchy and format of the system. The user class shows the attributes stored within the class to associate each user with the correct login in the form of name and email. This will then allow for the Admin to carry out tasks within a secure infrastructure as per the requirements. The diagram shows how the survey calls the questions and how the responses are linked; that is via the class identification code foreign keys that will be examined in more detail within the entity relationship diagram. The relationship between each class shows that the users and survey class have an exactly one-to-one dependency on each other. This is the same relationship throughout each class; survey to questions and questions to responses.

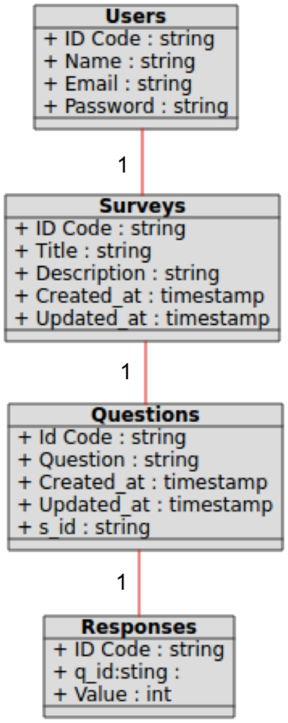


Figure 2: Here is a class diagram focusing on the database infrastructure.

2.3.4 ERD Diagram

Entity relationship diagrams, also know as ERD, also focus on the database structure and the relationship between each entity. However, this time it included primary and foreign key working through the stages from first normal form all the way to third, showing in-depth detail and complexity to the database. 'All entities within an entity set have the same set of attributes; each entity set has a key and each attribute has a domain' (Sengupta, 2016). These features benefit the use of entity relationship diagrams in ways such as; visual representation, effective communication, simplicity in understanding and high flexibility rate (Daniel, 2012).

The ERD produced in Figure 3 allows for easy understanding of the database structure, as mentioned by (Sengupta, 2016), demonstrating how each will link and call each other within the code. The first two entities depend solely on themselves dealing with the authorisation of the system, storing all the registrations including the password resets. The main sector of this diagram is the focus of the surveys which include questions then options for the multiple-choice, finalised by the storing of the responses. The way this diagram has been implemented clearly shows a logical method behind the database allowing for simpler communication between the system and the database. In addition, the relationships are based on one survey having one or more questions and each question due to multiple-choice will have many options which will have many responses by different respondents. Furthermore, the primary and foreign keys, presented as PK and FK respectively, show how the data within each entity is linked throughout the database. For example, the primary key in the survey is associated with the questions entity in the form of a foreign key allowing for the questions to be saved to the survey. This process works in the manner across the all entities with the foreign key being the primary key of the outer entity.

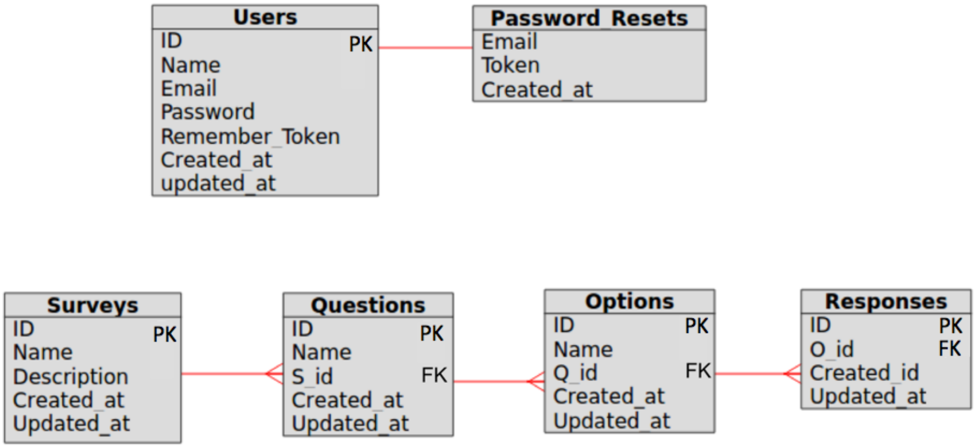


Figure 3: This entity relationship diagram focuses on the relationships throughout the database including the primary and foreign keys.

3 Technology Choices

Technology choices had to be made throughout implementation of the system including the appropriate set of framework(s) for the structure and styling of the web-based questionnaire system. The structural frameworks consisted of Laravel and AngularJS, or both, and the styling frameworks were between Bootstrap and Foundation.

Laravel is framework that covers all stages; from the first step all the way to finalising the styling of the system, whereas AngularJS is solely based on the front-end. AngularJS is a structural framework for dynamic web applications, which lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly (AngularJS, 2016). Laravel is also a web based application framework with expressive, elegant syntax (Laravel, 2016). Laravel and AngularJS could be used succinctly, using Laravel as the back-end and AngularJS as the front. However, as mentioned by (mnemon1ck, 2015) AngularJS is a very complex framework that takes time to understand and learn the specific way the code is used for the front-end.

This being said the chosen structural framework for the system was Laravel because it allows for all of the system requirements to be met and produced. For example, to be able to produce a system with a secure Admin back-end that will allow for different authority to be associated between different users. It also allows for a front-end to be implemented with styling frameworks to produce dynamic web-based systems.

The next decision was based upon the best suited styling framework for the system. The choice was narrowed down to Bootstrap and Foundation. According to (Bootstrap, 2016) it is 'the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web'. However, (ZurbFoundation, 2016) states it isn't necessarily the most popular but it is 'the most advanced responsive front-end framework in the world'.

After studying the availabilities and features available of both frameworks the chosen styling framework to compliment Laravel was Bootstrap. This was due to two reasons; the first was down to the availability, cooperativeness and resourcefulness of the Bootstrap website which would benefit greatly when styling the front-end of the system. The second reason was based upon the compiling issue of SaSS which is eliminated via the use of CSS which Bootstrap provided.

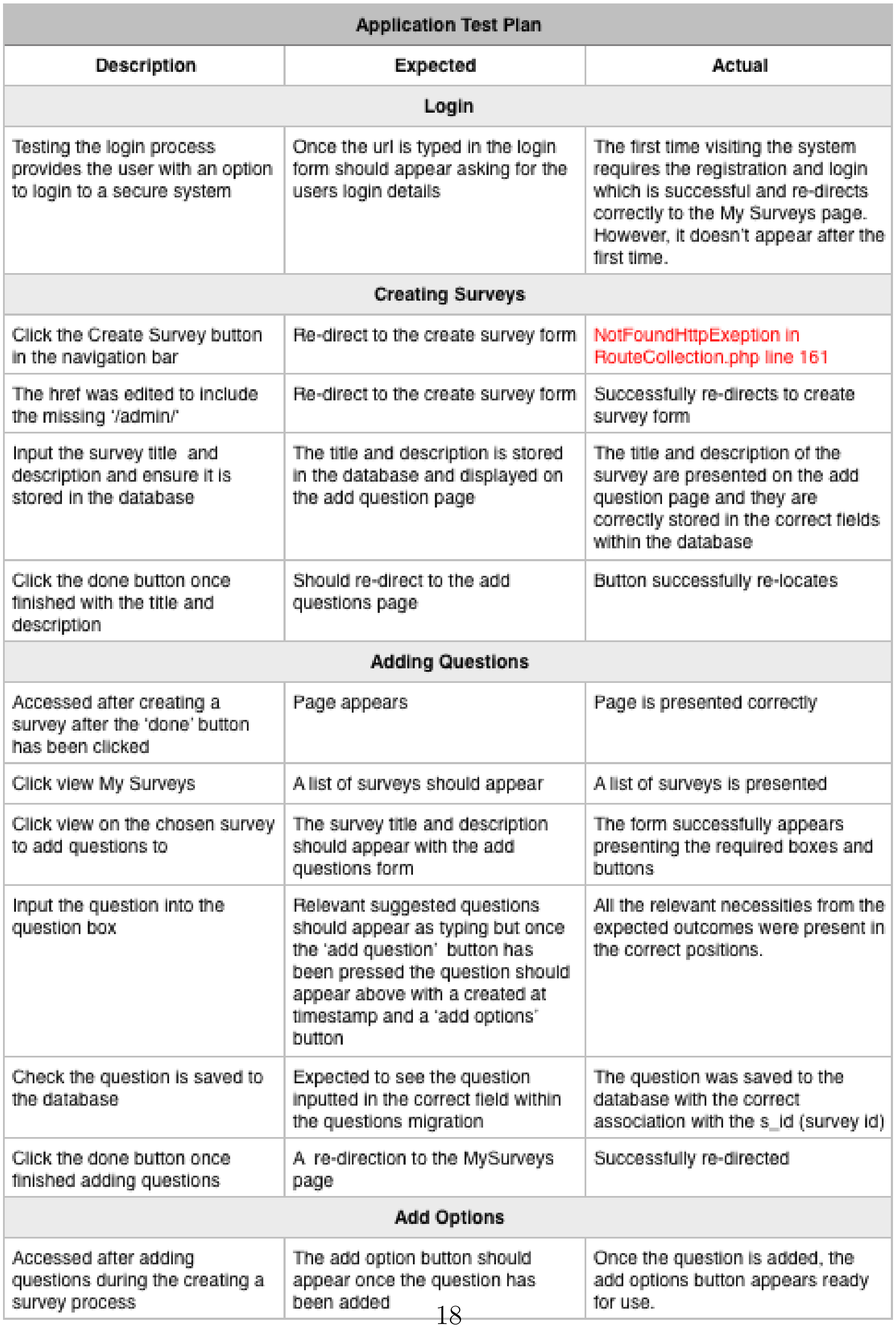
4 Software Development Process

The low fidelity prototypes allow for different designs to be drawn up quickly and receive feedback for improvement. Creating high fidelity prototypes primarily would lengthen this process dramatically, wasting time throughout the overall process. Once the low fidelity prototypes have been finalised then the next stage is to turn them into high fidelity providing a better understanding for the client, implementers and designers. The low fidelity prototypes for this web-based system can be seen in Appendix B through C. Appendix B contains all the figures to provide the first of the two low fidelity prototypes based on the information gathered during the system planning and research section of the report. Appendix C shows the final low fidelity design that the implementation was based upon.

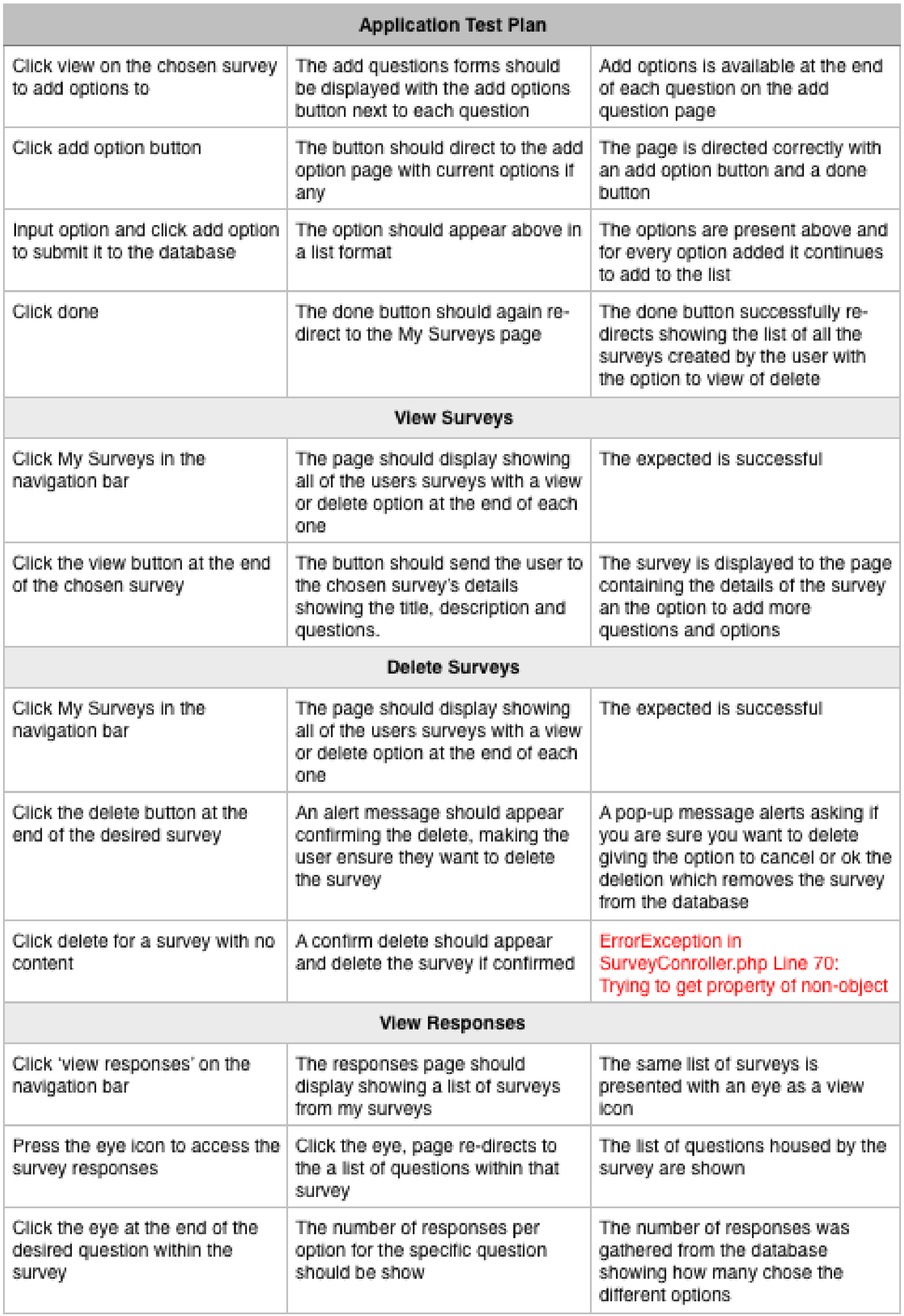
Each development process has a method to be followed providing structure to the project. The development process method followed throughout this project was the waterfall method because of the short requirement stage. The waterfall method works on a flowing diagram that doesn't return to stages, once one stage is complete the next begins. This is the same process for this project as the requirements have been set via the module handbook, meaning the required content is readily available at all stages of the project and therefore the requirement process doesn't need to be re-visited. Another reason why the waterfall method suits the process of this project is because of the single individual carrying out different tasks showing one task will be completed and executed at a time. Furthermore, the structure associated with this method will provide a structure for the individual to work to instead of carrying out tasks from many different stages at once.

5 Application Testing

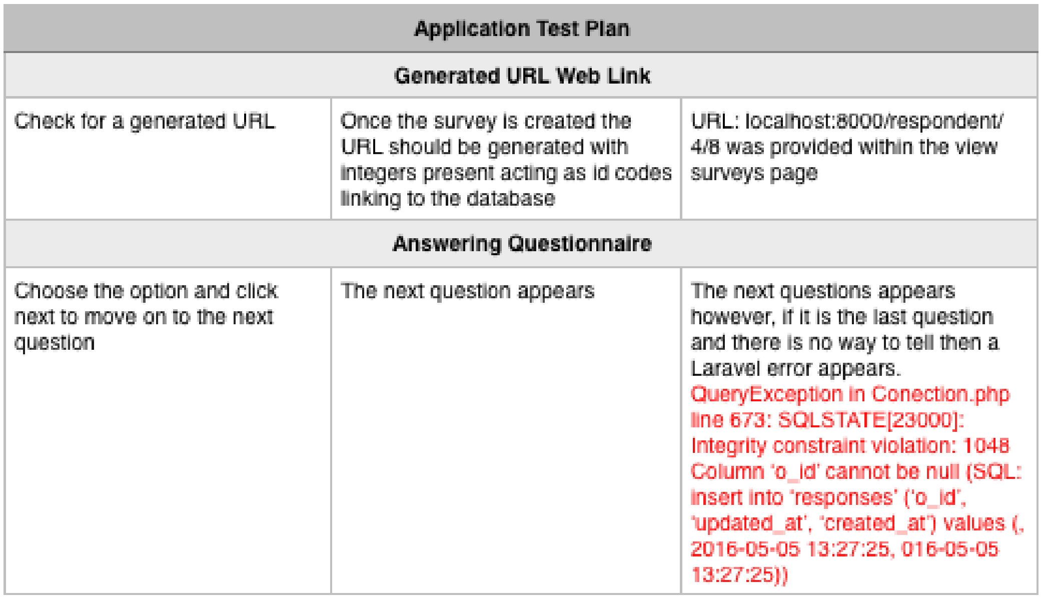
This section will consist of the testing; testing each process for the abilities of the system. The testing will be presented in a tabular format describing the test, stating the expected outcome and displaying the actual outcomes. Each actual outcome that fails to meet the expected outcome will be altered and re-tested until it meets the expected action.



Application Test Plan		
Description	**Expected**	**Actual**
Login		
Testing the login process provides the user with an option to login to a secure system	Once the url is typed in the login form should appear asking for the users login details	The first time visiting the system requires the registration and login which is successful and re-directs correctly to the My Surveys page. However, it doesn't appear after the first time.
Creating Surveys		
Click the Create Survey button in the navigation bar	Re-direct to the create survey form	NotFoundHttpException in RouteCollection.php line 161
The href was edited to include the missing '/admin/'	Re-direct to the create survey form	Successfully re-directs to create survey form
Input the survey title and description and ensure it is stored in the database	The title and description is stored in the database and displayed on the add question page	The title and description of the survey are presented on the add question page and they are correctly stored in the correct fields within the database
Click the done button once finished with the title and description	Should re-direct to the add questions page	Button successfully re-locates
Adding Questions		
Accessed after creating a survey after the 'done' button has been clicked	Page appears	Page is presented correctly
Click view My Surveys	A list of surveys should appear	A list of surveys is presented
Click view on the chosen survey to add questions to	The survey title and description should appear with the add questions form	The form successfully appears presenting the required boxes and buttons
Input the question into the question box	Relevant suggested questions should appear as typing but once the 'add question' button has been pressed the question should appear above with a created at timestamp and a 'add options' button	All the relevant necessities from the expected outcomes were present in the correct positions.
Check the question is saved to the database	Expected to see the question inputted in the correct field within the questions migration	The question was saved to the database with the correct association with the s_id (survey id)
Click the done button once finished adding questions	A re-direction to the MySurveys page	Successfully re-directed
Add Options		
Accessed after adding questions during the creating a survey process	The add option button should appear once the question has been added	Once the question is added, the add options button appears ready for use.



Application Test Plan		
Click view on the chosen survey to add options to	The add questions forms should be displayed with the add options button next to each question	Add options is available at the end of each question on the add question page
Click add option button	The button should direct to the add option page with current options if any	The page is directed correctly with an add option button and a done button
Input option and click add option to submit it to the database	The option should appear above in a list format	The options are present above and for every option added it continues to add to the list
Click done	The done button should again re-direct to the My Surveys page	The done button successfully re-directs showing the list of all the surveys created by the user with the option to view of delete
View Surveys		
Click My Surveys in the navigation bar	The page should display showing all of the users surveys with a view or delete option at the end of each one	The expected is successful
Click the view button at the end of the chosen survey	The button should send the user to the chosen survey's details showing the title, description and questions.	The survey is displayed to the page containing the details of the survey an the option to add more questions and options
Delete Surveys		
Click My Surveys in the navigation bar	The page should display showing all of the users surveys with a view or delete option at the end of each one	The expected is successful
Click the delete button at the end of the desired survey	An alert message should appear confirming the delete, making the user ensure they want to delete the survey	A pop-up message alerts asking if you are sure you want to delete giving the option to cancel or ok the deletion which removes the survey from the database
Click delete for a survey with no content	A confirm delete should appear and delete the survey if confirmed	ErrorException in SurveyConroller.php Line 70: Trying to get property of non-object
View Responses		
Click 'view responses' on the navigation bar	The responses page should display showing a list of surveys from my surveys	The same list of surveys is presented with an eye as a view icon
Press the eye icon to access the survey responses	Click the eye, page re-directs to the a list of questions within that survey	The list of questions housed by the survey are shown
Click the eye at the end of the desired question within the survey	The number of responses per option for the specific question should be show	The number of responses was gathered from the database showing how many chose the different options



Application Test Plan		
Generated URL Web Link		
Check for a generated URL	Once the survey is created the URL should be generated with integers present acting as id codes linking to the database	URL: localhost:8000/respondent/4/8 was provided within the view surveys page
Answering Questionnaire		
Choose the option and click next to move on to the next question	The next question appears	The next questions appears however, if it is the last question and there is no way to tell then a Laravel error appears. QueryException in Conection.php line 673: SQLSTATE[23000]: Integrity constraint violation: 1048 Column ‘o_id’ cannot be null (SQL: insert into ‘responses’ (‘o_id’, ‘updated_at’, ‘created_at’) values (, 2016-05-05 13:27:25, 016-05-05 13:27:25))

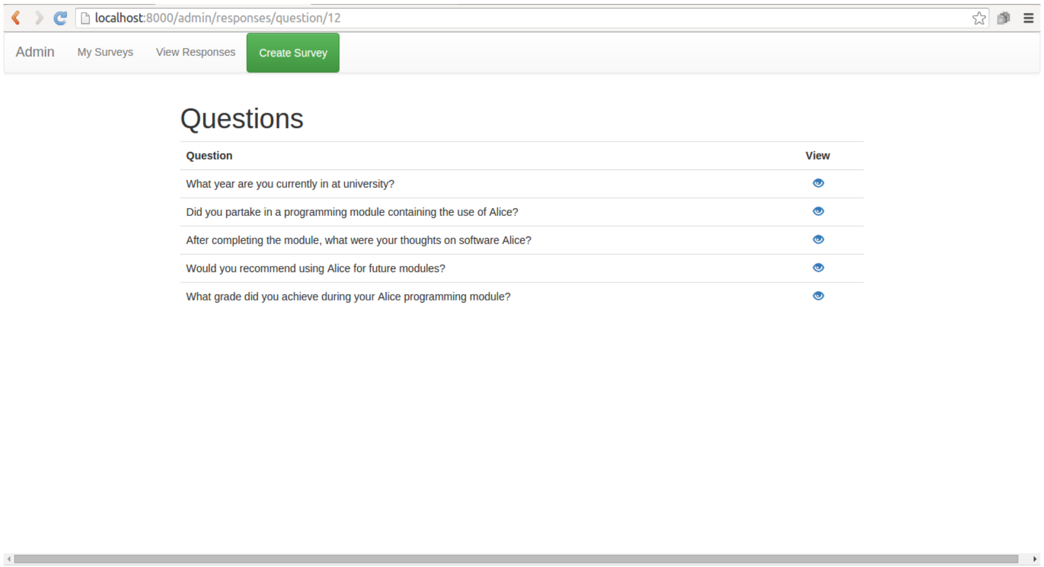
Table 2: Application testing carried to ensure the system functions as intended, reducing and mending all the errors that can occur during this process

6 Questionnaire Design

As gathered from the literature review the design of the questionnaires is vital to produce optimised results. The questionnaire being created to test the system and produce a true set of responses will be focussed at first year computing students based on the Alice programming application. The target audience is first year computing students because of their recent up-to-date knowledge of the application due to their current module assessments.

The questionnaire will consist of five questions and will be answered by five respondents. Each question will be multiple choice, which will compliment the five question formulae to keep the attention span focussed on the questions for a short period of time. The questions will begin very basic and general and increase in depth towards the end as the more specific and technical questions are asked.

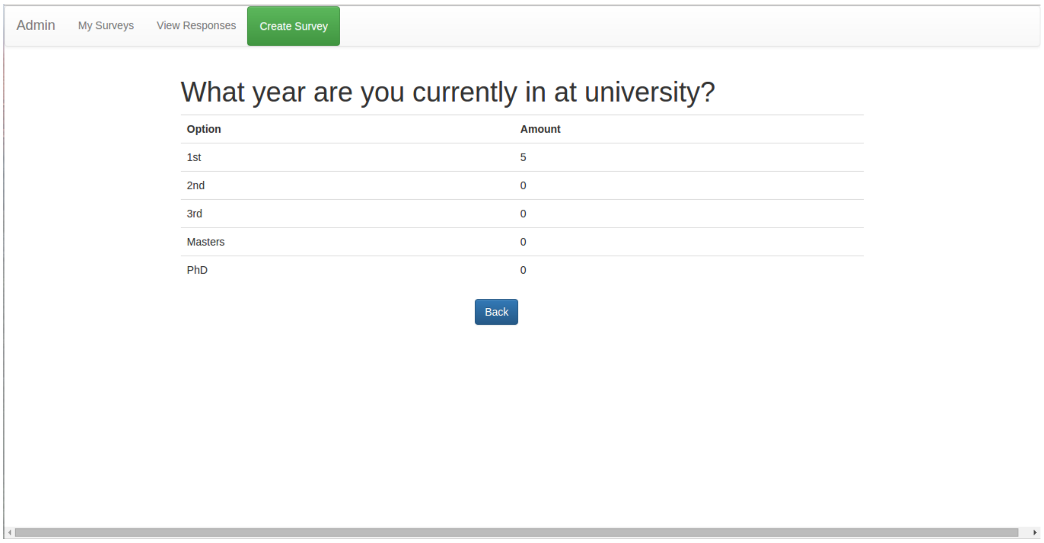
There are no implementations within the code to limit the number of questions or responses that can be used per questionnaire. Therefore, the system relies on the researcher to have an efficient and effective understanding on the optimum questionnaire design and formulation to produce the best for their research.



Questions

Question	View
What year are you currently in at university?	👁
Did you partake in a programming module containing the use of Alice?	👁
After completing the module, what were your thoughts on software Alice?	👁
Would you recommend using Alice for future modules?	👁
What grade did you achieve during your Alice programming module?	👁

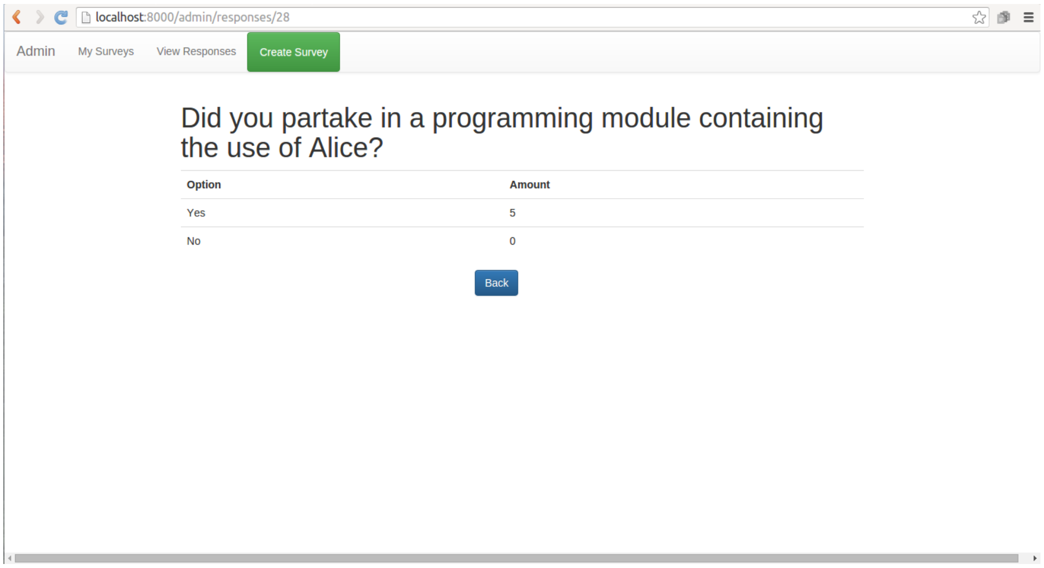
Figure 4: This shows the list of five questions created for the survey.



What year are you currently in at university?

Option	Amount
1st	5
2nd	0
3rd	0
Masters	0
PhD	0

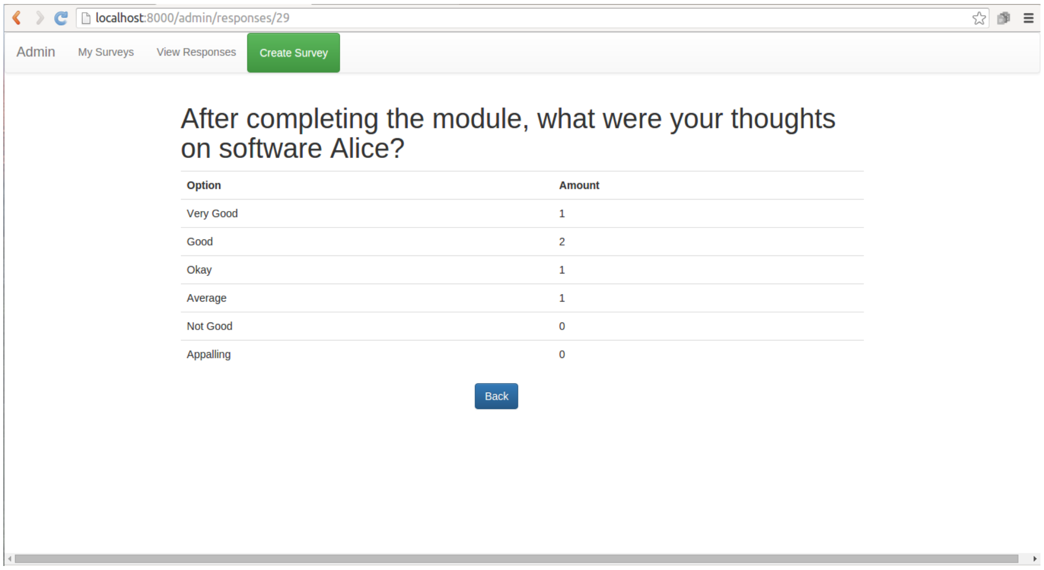
Figure 5: This shows the responses the respondents provided for the first question.



Did you partake in a programming module containing the use of Alice?

Option	Amount
Yes	5
No	0

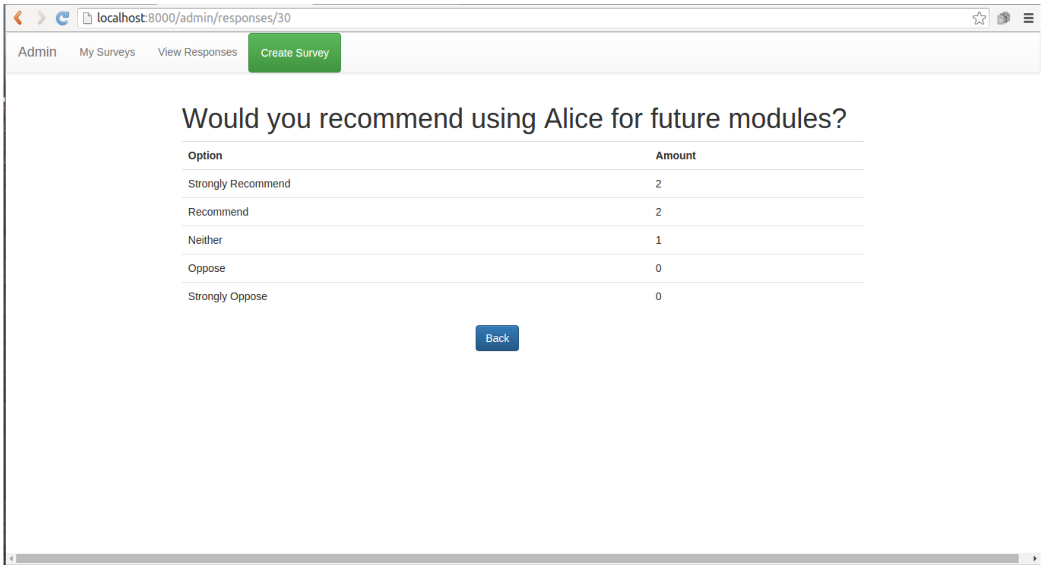
Figure 6: Responses for question two of five.



After completing the module, what were your thoughts on software Alice?

Option	Amount
Very Good	1
Good	2
Okay	1
Average	1
Not Good	0
Appalling	0

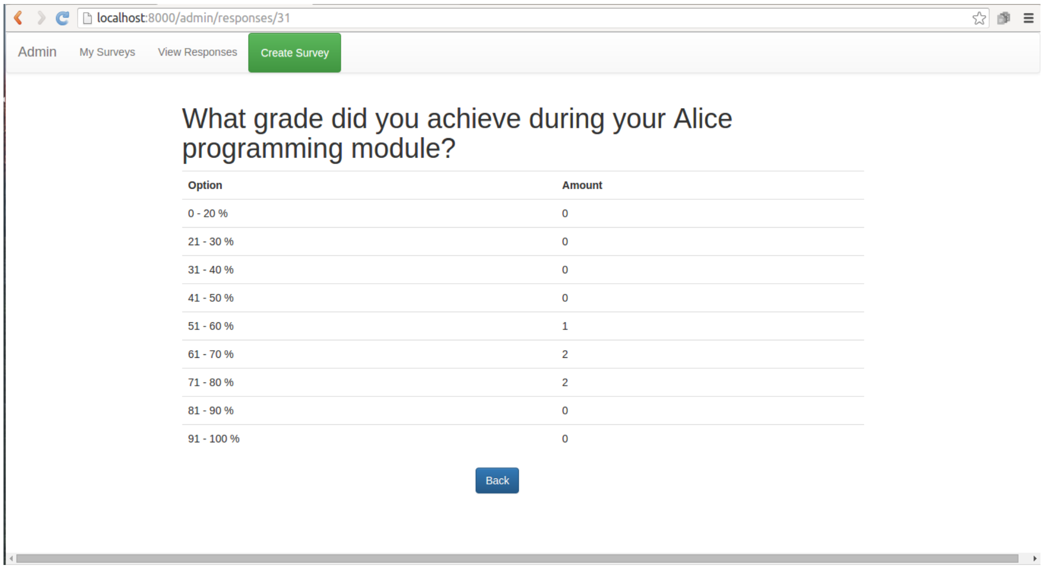
Figure 7: Here are the responses the respondents provided for the third question.



Would you recommend using Alice for future modules?

Option	Amount
Strongly Recommend	2
Recommend	2
Neither	1
Oppose	0
Strongly Oppose	0

Figure 8: These are the responses for question four.



What grade did you achieve during your Alice programming module?

Option	Amount
0 - 20 %	0
21 - 30 %	0
31 - 40 %	0
41 - 50 %	0
51 - 60 %	1
61 - 70 %	2
71 - 80 %	2
81 - 90 %	0
91 - 100 %	0

Figure 9: This shows the responses the respondents provided for the fifth question.

7 Project Post-Mortem

Analysing completed work such as projects is a valuable way to highlight improvements to further benefit future projects allowing learning, knowledge and capabilities to be increased and expanded upon. This section will be divided into three main areas; positives, negatives and improvements. Each section will lead up to discussing the improvements that could be made if the project was to be completed again.

7.1 Positives

The positives of the project begin with involving the system planning and research sector, which helped layout the stages and content needed for the project. Planning is vital to reducing errors and saving time, therefore, drawing the diagrams on paper first (Appendix A) allowed for errors to made and changed in a much quicker time frame than using software such as Umbrello to constantly edit the digitalised version. In addition, planning on paper first allows for multiple variations to be thought without hindering the time frame. Designing on paper also worked very well for the low-fidelity prototypes enabling different designs to be created in a much shorter space of time.

Another positive to the project would be studying the question design. This not only helped understand the logic behind the structure, format, question length, but also helped with the designing the appearance of the site. The information gathered was invaluable as users are more likely to respond well to an aesthetically pleasing website. (Alsudani and Casey, 2009) states that 'experiments have shown that users can judge a web site's credibility in as little as 3.42 seconds merely on the basis of its aesthetic appeal'.

The final positive to be discussed is the methodical working following the waterfall method. This process was most beneficial during the coding allowing for a better understanding of the code and how it functions, setting in-putted data to variable which allows for easier calling between the front end and the database.

7.2 Negatives

Negatives unfortunately are the key to improvements. The negatives found throughout the project consisted of the struggle with third normal form and code inabilities. Third normal form can be complicated to understand for the first time, during this project was the first real use of third normal form within the entity relationship diagram which displayed its issues to begin with. This could be resolved by carrying out more practice creating mock up database entity relation ship diagrams in first, second and third normal form to improve and gain an overall knowledge on the topic area allowing for an better understanding of the different normal forms.

Another negative would certain imperfections within the build of the web-base system. The first one would be based on the deleting of surveys; when deleting a full created survey the deletion process works effortlessly. However, if a survey has been created but doesn't contain any options within the questions then the system fails to delete the survey. This is a poor implementation that failed to be improved and fixed.

In addition, the inability to comply with the research and provide the respondents with an implementing progress such as question numbers is a negative within the code of the project. This failed to be implemented due to the lack of understanding of how to efficiently associate and count the questions added to the survey.

7.3 Improvements

The final stage of the project post-mortem and the most influential is the improvements that could be made to further advance the project. The improvements will be discussed in two sections; planning and code implementation.

In order to improve the planning of the project to help the build, it could be suggested that knowledge on third normal form could be greatly improved. This in turn would increase the efficiency of the database infrastructure. The second improvement could be to further the UML diagrams. For example, process modelling could be used to include activity diagrams which would outlay the process for each task. Again this would help the efficiency of the implementation as the process to be followed would be clear. Another way the project could be improved is in regards to the front-end design of the system, which if given extra time could be tweaked and re-designed based on user experience and interaction. The suggestion of carrying out a small user experience (UX) test on the engagement and experience with the different variations of the low-fidelity prototypes could improve the decision of the finalised design.

Coding is very time consuming especially when it involves researching resources to help implement the various require features. If the chance was given again to improve the project the capabilities and efficiency of the code would greatly benefit the overall system. In addition, the deletion process could be perfected to delete surveys with no options and incorporate the addition of numbers for each question. This would increase the appeal of the system. Another upgrade that could be made to the system is the implementation of a finish button and page to improve the dialogue between the respondent and the system. This should then remove the 'QueryException' error in 'Connection.php' because there should be a re-direction. The final improvement to be suggested is setting limitations within the system, this would increase the amount of optimised questionnaires being created by the Admin users. If the user has a lack of questionnaire design knowledge then the limitation would counteract and allow for the best possible results to be gathered.

References

F. Alsudani and M. Casey. The effect of aesthetics on web credibility. *People and Computers XXIII – Celebrating people and technology*, pages 512–513, 2009.

AngularJS. What is angular?, 2016. URL `https://docs.angularjs.org/guide/introduction`. Accessed: 2016-3-30.

Bootstrap. Bootstrap is the most popular html, css, and js framework for developing responsive, mobile first projects on the web, 2016. URL `http://getbootstrap.com`. Accessed: 2016-4-19.

BusinessDictionary. Questionnaire, 2016. URL `http://www.businessdictionary.com/definition/questionnaire.html`. Accessed: 2016-3-18.

H. Daniel. Benefits of erd. 2012.

D. Elenburg. Use cases: Background, best practices, and benefits. *MKS White Paper*, pages 1–3, 2005.

Google, 2016. URL `https://www.google.com/forms/about/`. Accessed: 2016-3-18.

A. Green. Benefits of system modelling. *PS Enterprise*, 2016.

KirkleesCouncil. Research & consultation guidelines. pages 1–6, 2015.

J. A. Krosnick and S. Presser. Question and questionnaire design. pages 1–4, 2009.

M. Kusek, S. Desic, and D. Gvozdanovic. Uml based object-oriented development: Experience with inexperienced developers. *International Conference on Telecommunications*, 2001.

Laravel. Laravel philosophy, 2016. URL `https://laravel.com/docs/4.2/introduction`. Accessed: 2016-3-30.

W. C. Leung. How to design a questionnaire. *Student BMJ*, 13:187–189, 2001.

M. Marrs. 7 best survey tools: Create awesome surveys for free!, 2014. URL `https://www.citethisforme.com/cite/website`. Accessed: 2016-3-20.

mnemon1ck. Why you should not use angularjs, 2015. URL `https://medium.com/@mnemon1ck/why-you-should-not-use-angularjs`. Accessed: 2016-4-19.

Pollfish, 2016. URL `https://www.pollfish.com`. Accessed: 2016-3-18.

Qualtrics. Multiple choice, 2016. URL `https://www.qualtrics.com/support/survey-platform/edit-survey/editing-questions/question-types-guide/standard-content/multiple-choice/`. Accessed: 2016-3-20.

A. Sengupta. Mis710 module 1a data and process modeling. *Wright State University*, 2016.

ZurbFoundation. Foundation - the most advanced responsive front-end framework in the world, 2016. URL `http://foundation.zurb.com`. Accessed: 2016-4-19.

8 Appendices

8.1 Appendix A - UML Diagram Planning

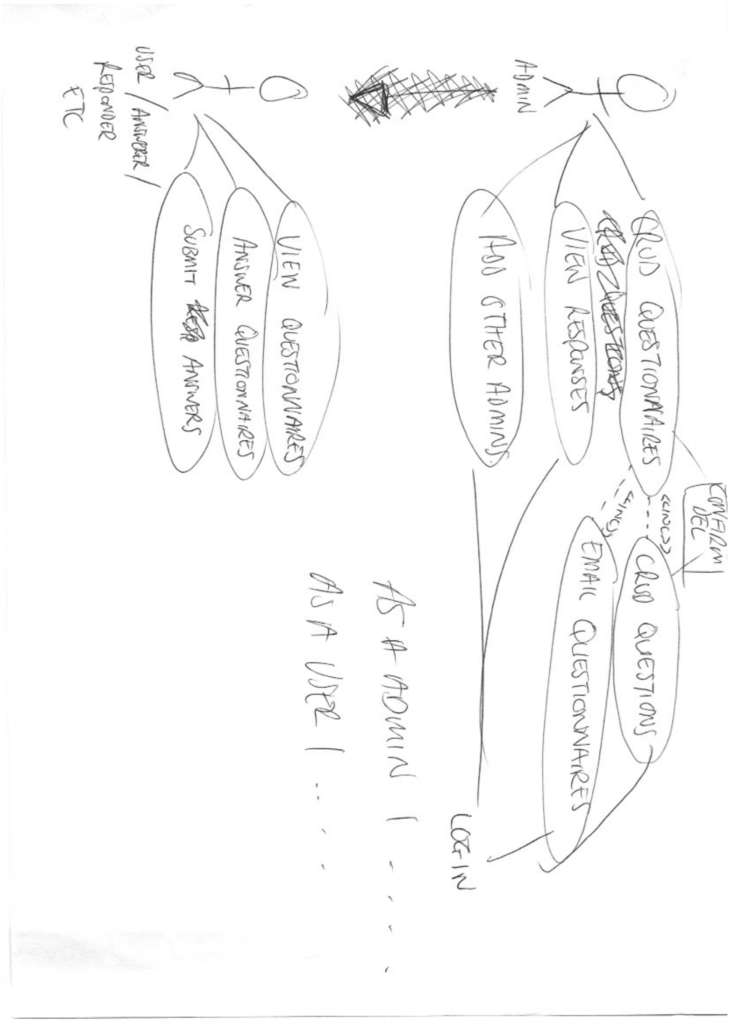


Figure 10: Stage one of the use-case diagram implementation.

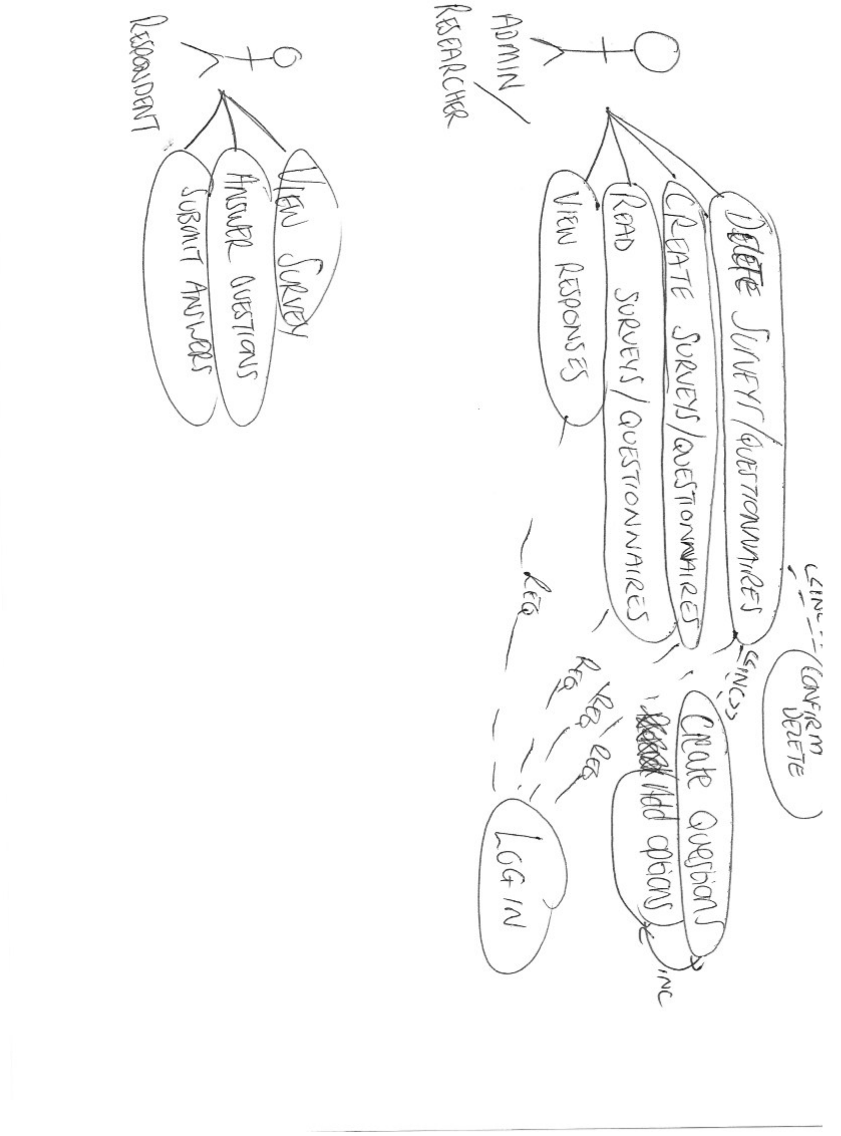


Figure 11: This has been defined and condensed which was used to create the digitalised version.

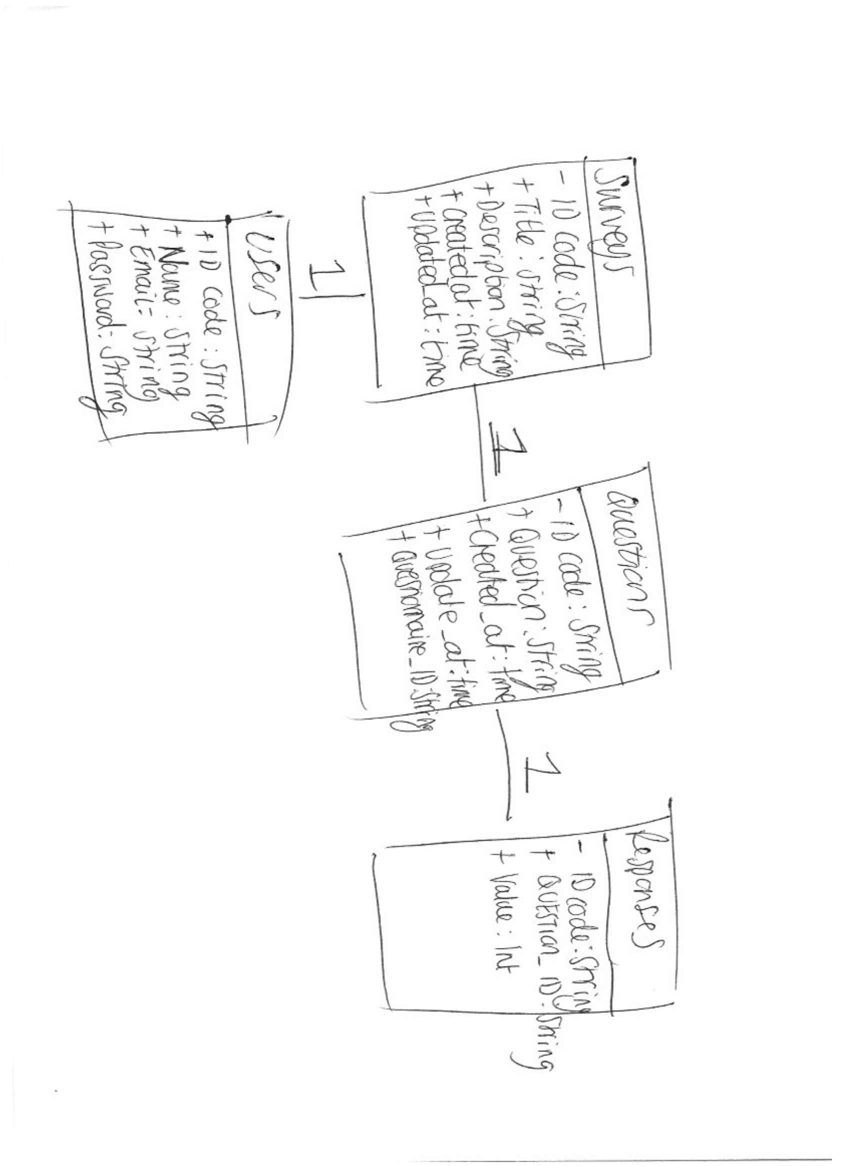


Figure 12: This is the planning for the class diagram showing the database structure.

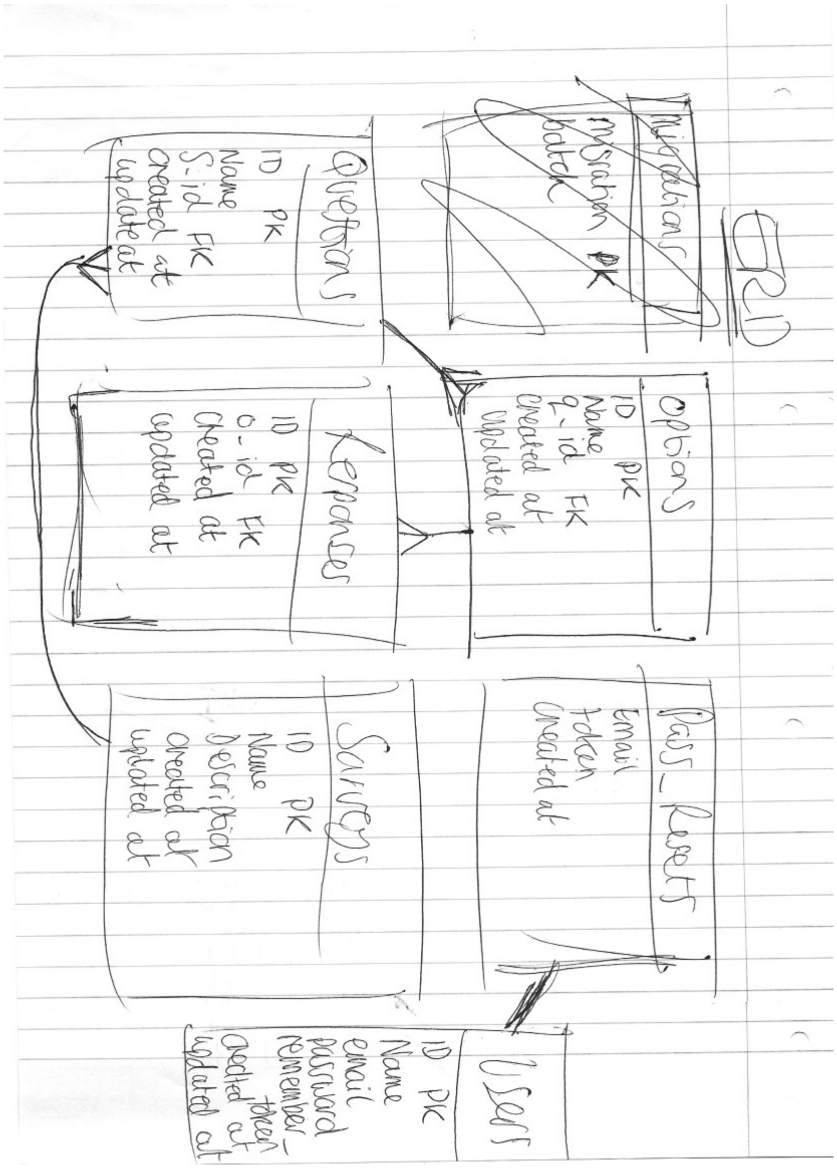


Figure 13: Stage one of the ERD diagram.

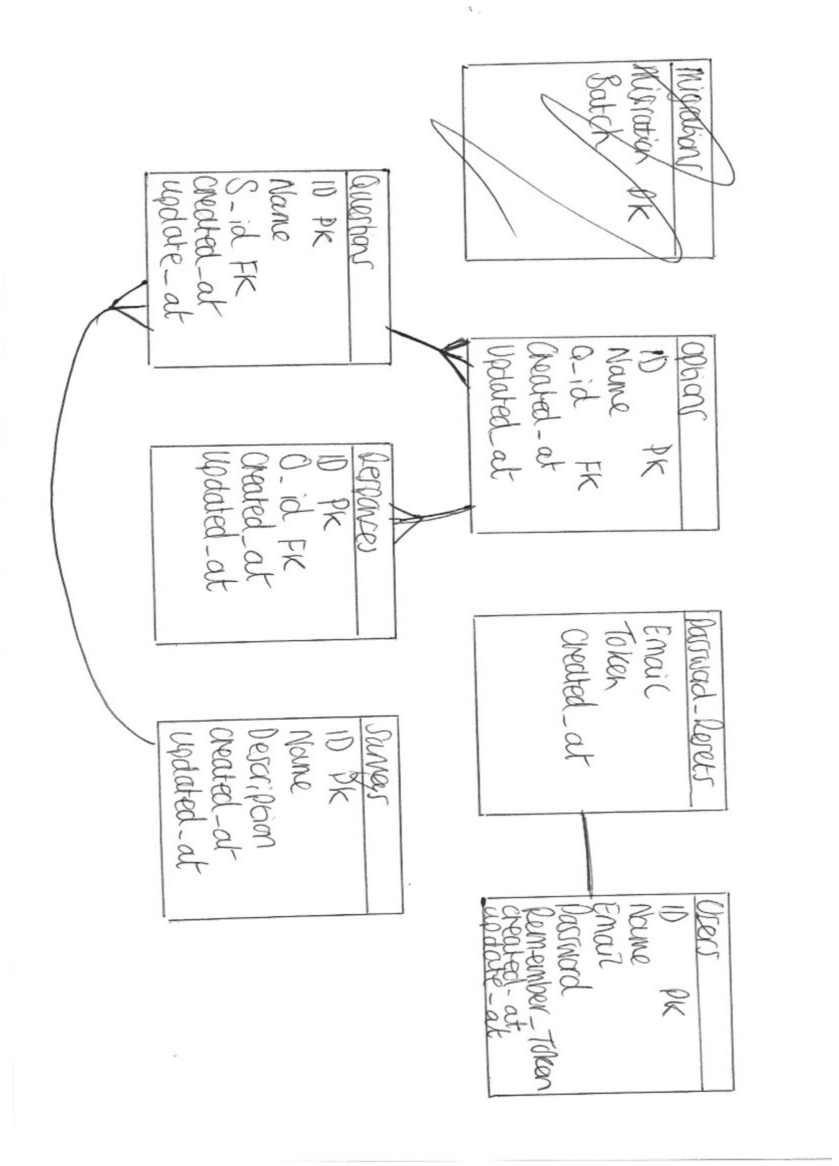


Figure 14: Finalising the ERD ready for digitalising.

8.2 Appendix B - Low Fidelity Prototype 1

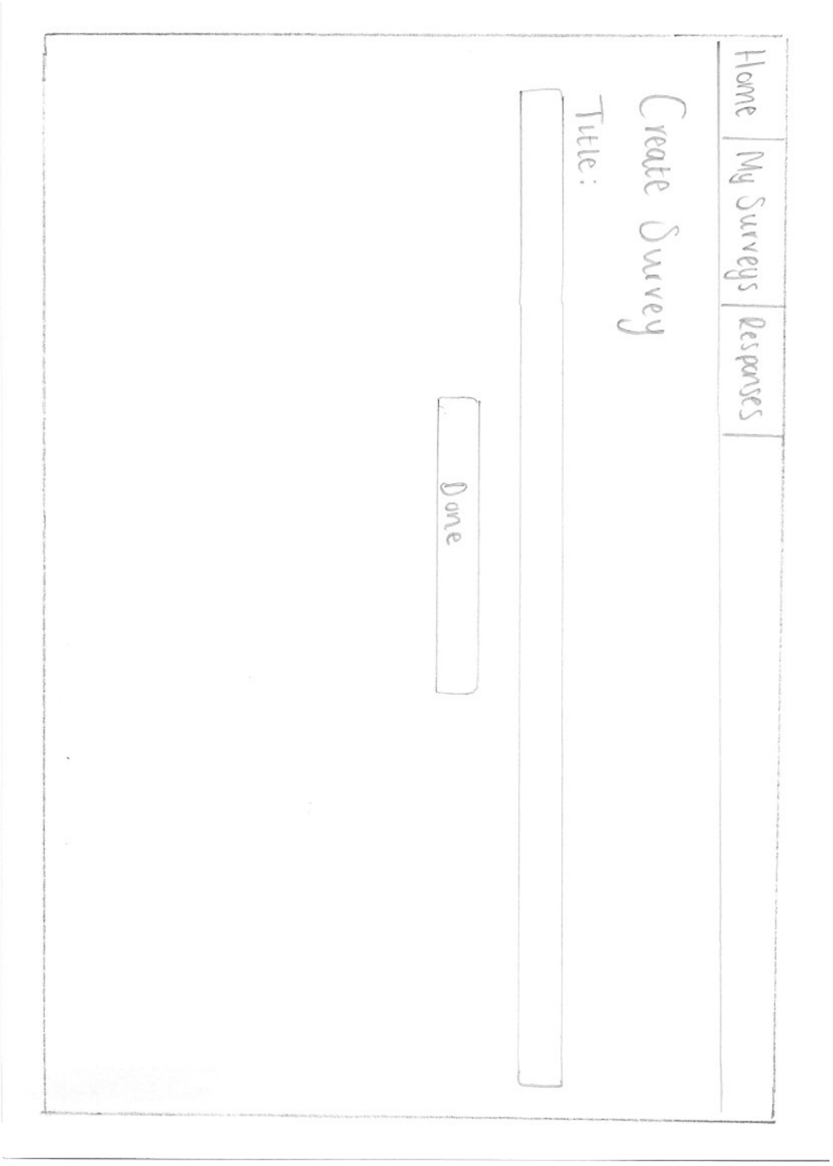


Figure 15: This images shows the first design of creating a survey.

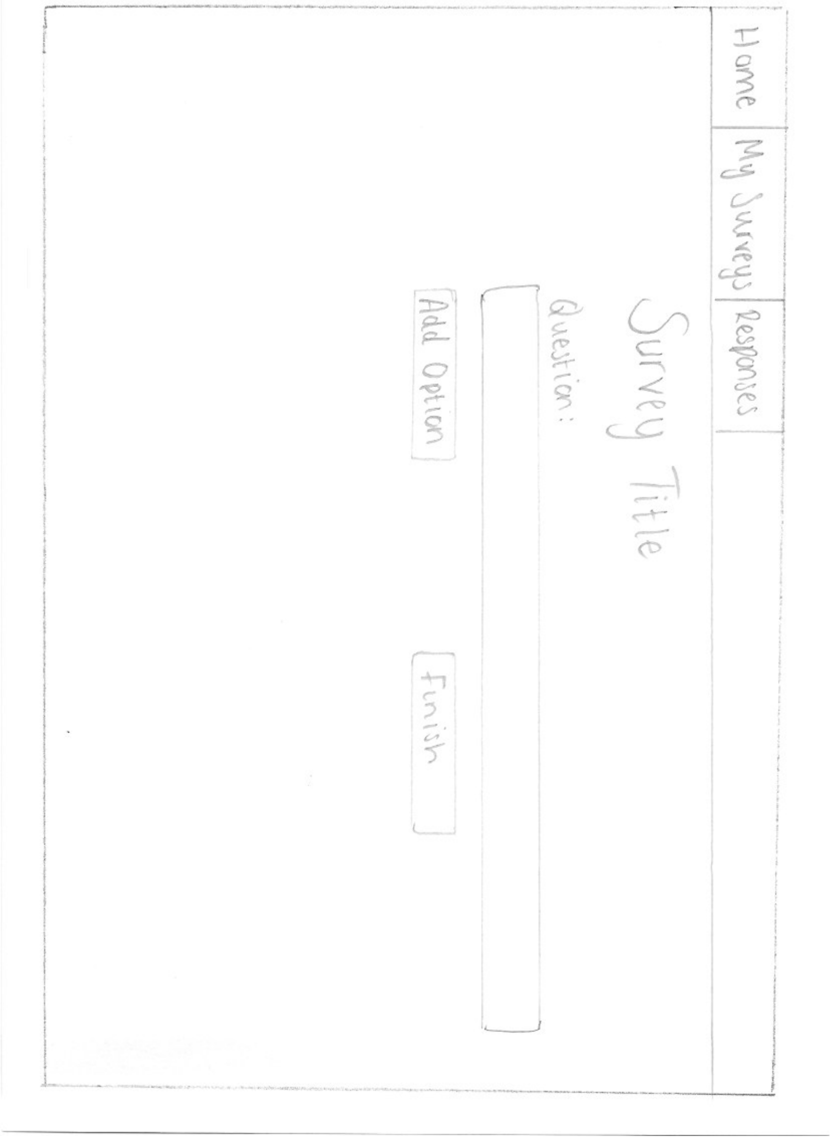


Figure 16: This design is for the adding of questions to the survey.

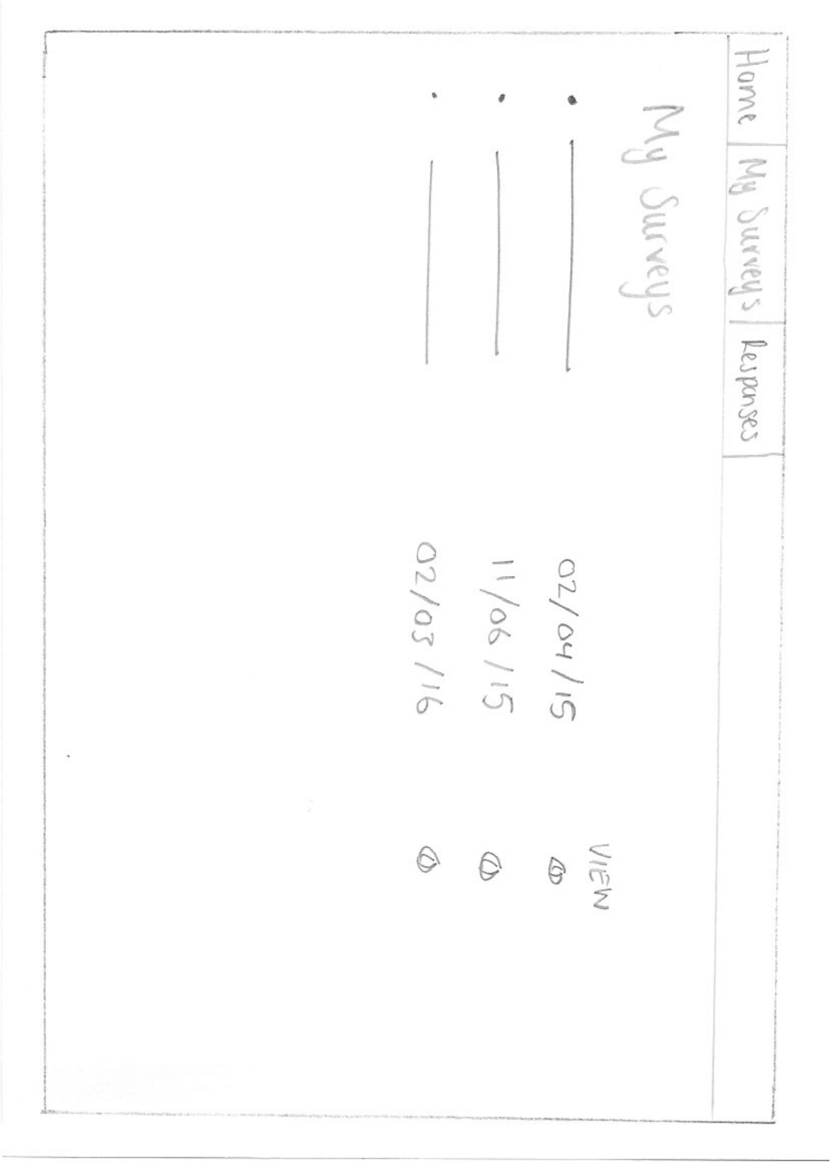


Figure 17: This is a list of all the user's surveys they have created.

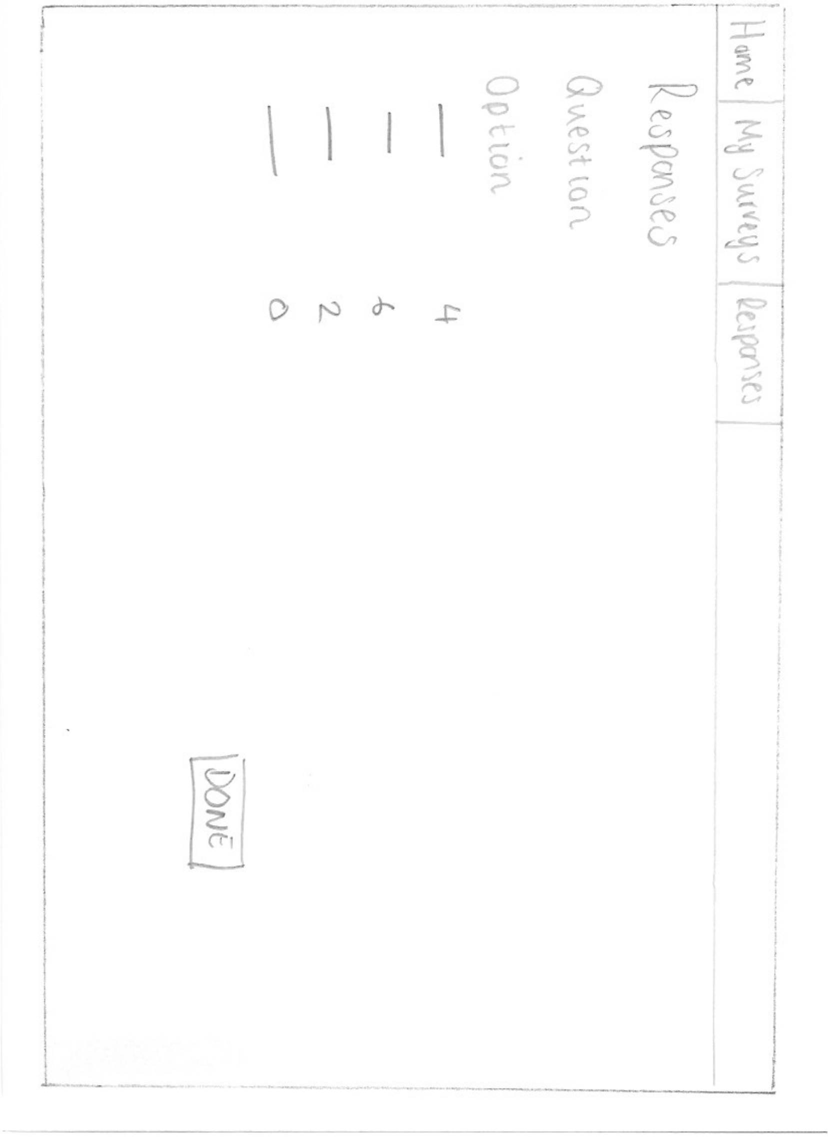


Figure 18: This shows the responses gathered from the respondents.

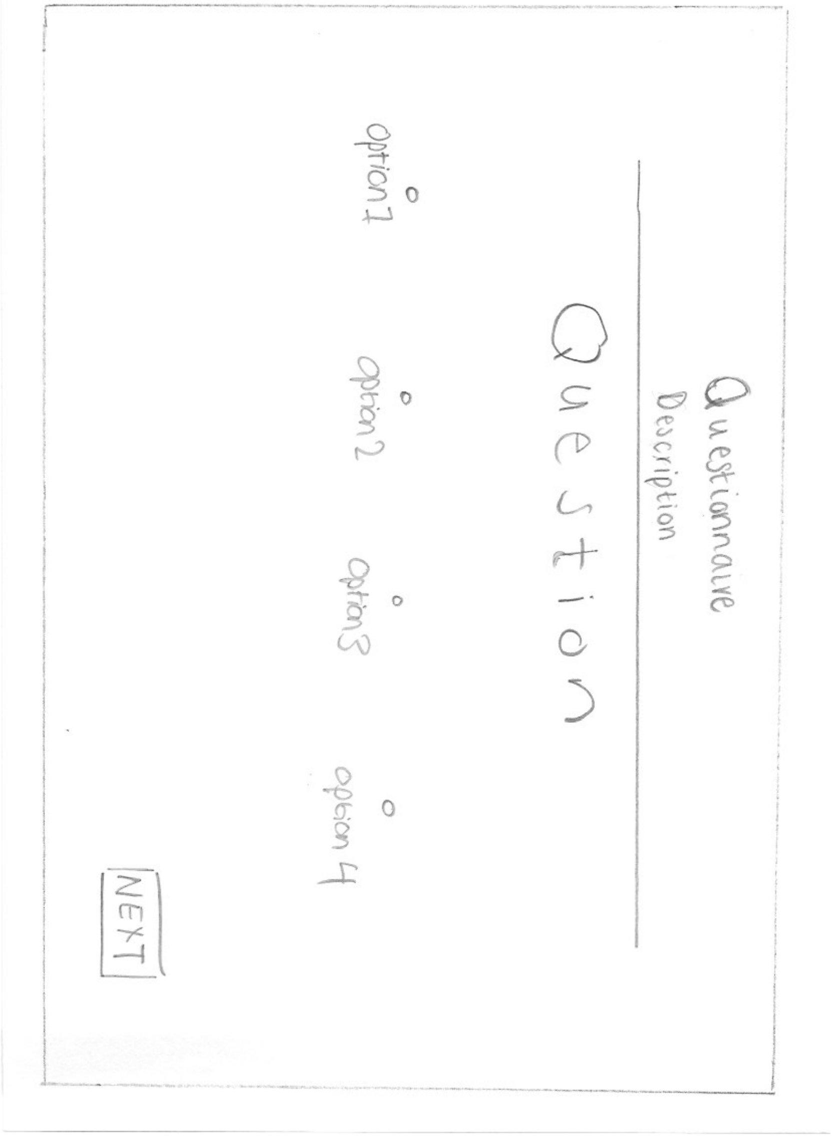


Figure 19: Here is the front-end of the survey that the respondents will access via the URL web link.

8.3 Appendix C - Low Fidelity Prototype 2

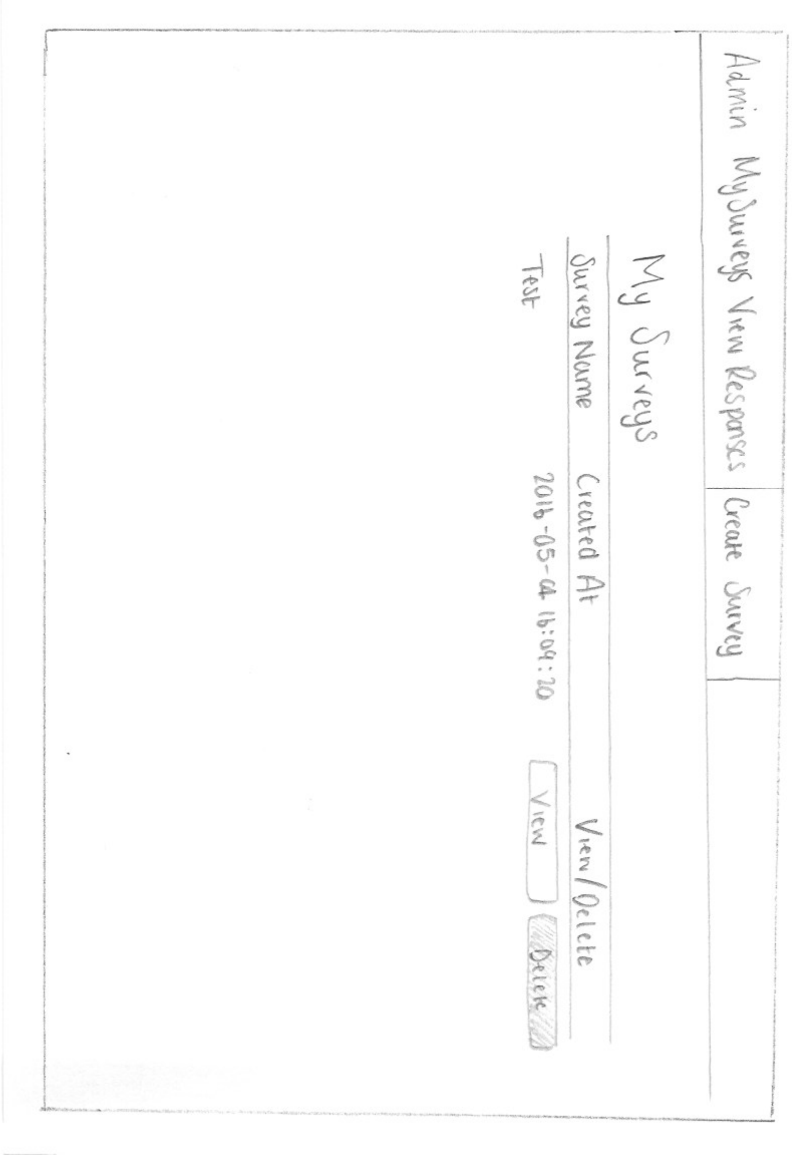


Figure 20: This entity relationship diagram focuses on the relationships throughout the database including the primary and foreign keys.

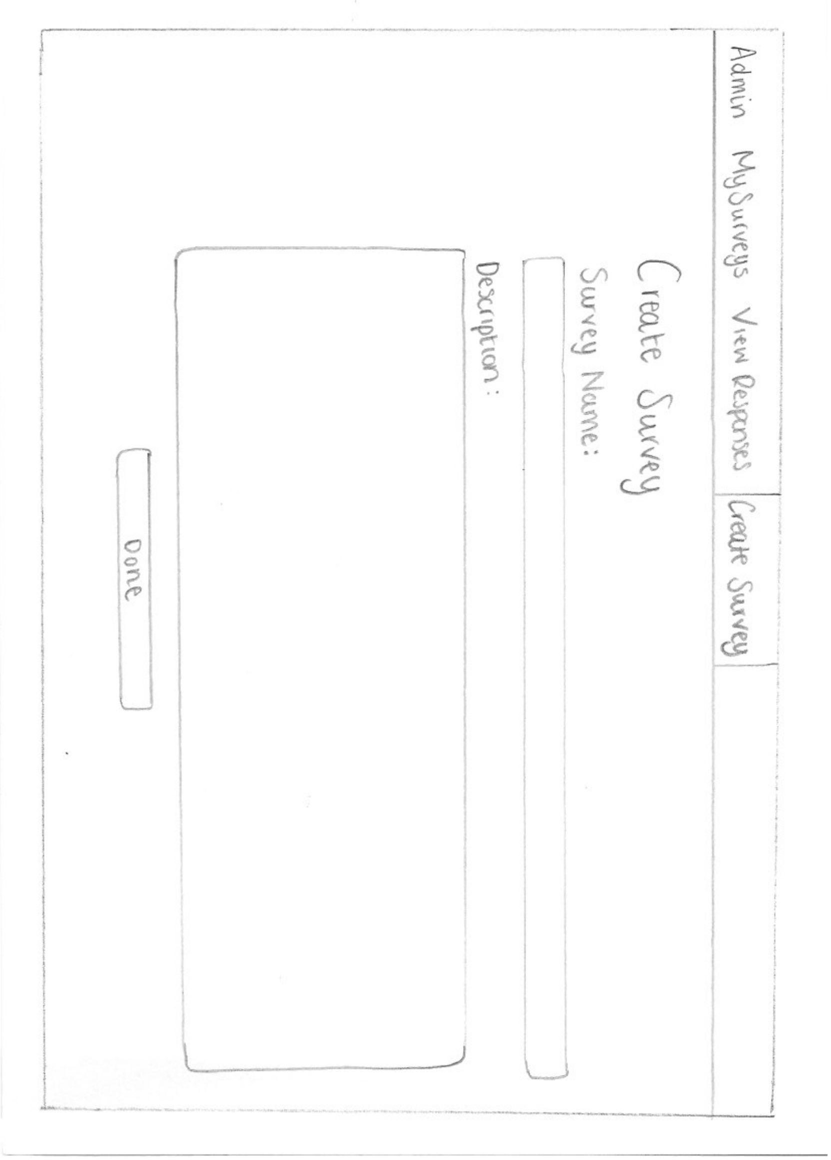


Figure 21: This is the finalised design of the My Surveys page for the Admin user to create, view and see responses for all their surveys.

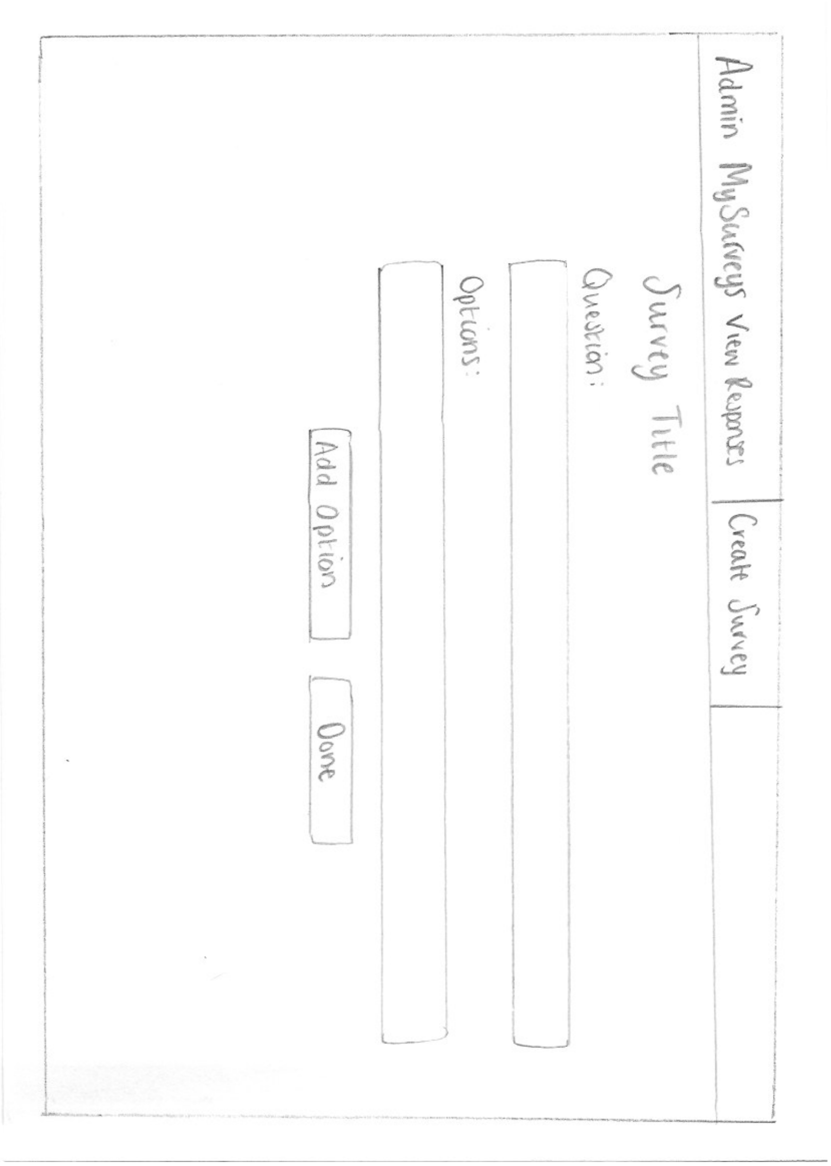


Figure 22: This is the create survey/questionnaire form the user will complete.

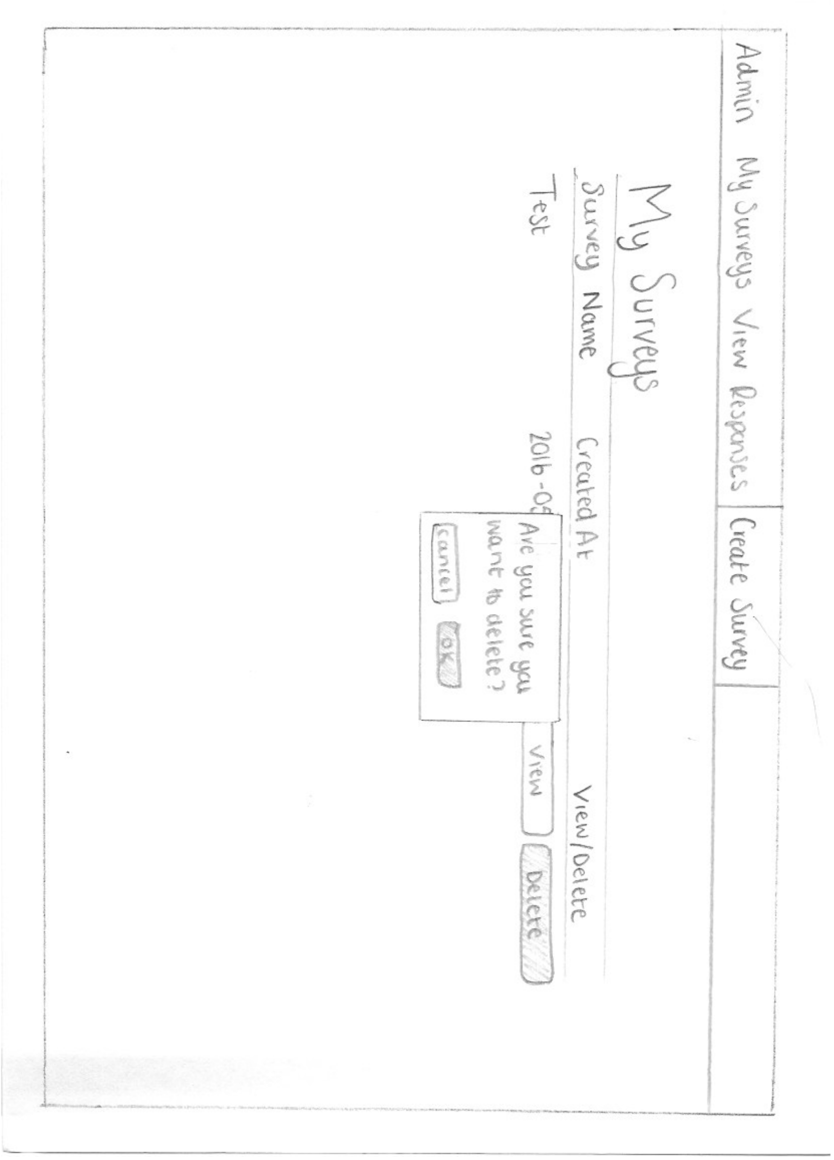
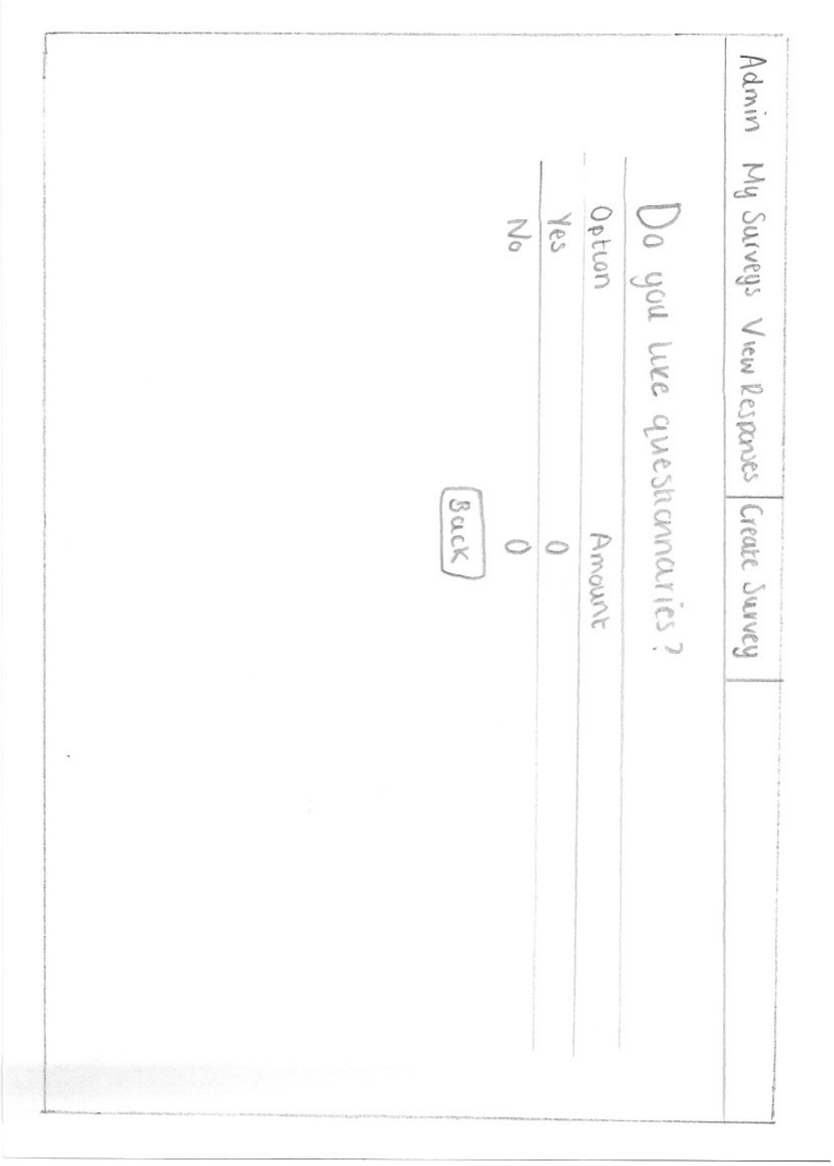


Figure 23: Here is the dialogue confirmation box the Admin user will be faced with when they are trying to delete a survey.



Do you like questionnaries ?

Option	Amount
Yes	0
No	0

Figure 24: This image displays how the Admin user will see the responses initiated from the respondents responses to the questions.

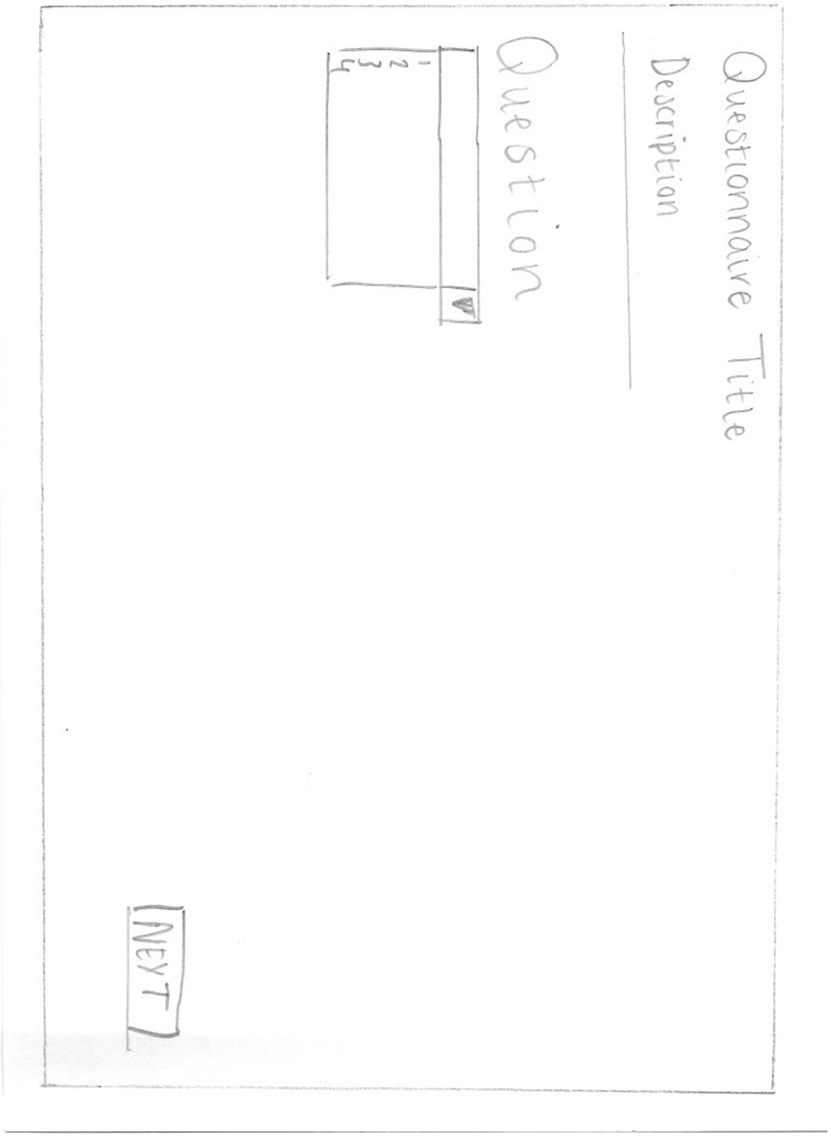


Figure 25: This is final design of the front-end view for the respondents to answer the question.