NOTES OF GUIDANCE ON THE COMPLETION OF THE NOTICE OF SUBMISSION FORM

You should complete both sides of the form in typescript or black ink and use block capitals. The completed form must be handed to your Head/Dean of School (or his/her nominee) with two copies of your dissertation and (where appropriate) the examination fee.

Each copy of the dissertation shall contain:

1. a summary not exceeding three hundred words;
2. a statement signed by the candidate showing the extent to which the work submitted is the result of the candidate's own investigation, and an explicit acknowledgement (with references) of any other sources used;
3. a full bibliography;
4. a signed declaration to certify that the work submitted has not been accepted in substance for any degree or award, and is not being submitted concurrently in candidature for any degree or other award;
5. a signed statement regarding availability of the Dissertation;
6. a completed 'Notice of Submission' form.

The declaration and statements referred to in 2 to 5 above should be incorporated at the beginning of the dissertation, as shown in Appendix 1.

Where this is in accordance with the School's policy on the submission of Taught Master's Dissertations:

- candidates may submit their work for examination in temporary binding;
- candidates may be instructed by the Schools to submit one of the two required copies in an approved electronic format.

APPENDIX 1: Specimen Layout for Declaration/Statements page to be included in Taught Master's Degree Dissertations

CANDIDATE'S ID NUMBER	C1837071
CANDIDATE'S SURNAME	**Please circle appropriate value** **Mr / Miss / Ms / Mrs / Rev / Dr / Other please specify** YE
CANDIDATE'S FULL FORENAMES	YUJIAO

DECLARATION

This work has not previously been accepted in substance for any degree and is not concurrently submitted in candidature for any degree.

Signed ..... (candidate) Date 20/09/2019

STATEMENT 1

This dissertation is being submitted in partial fulfillment of the requirements for the degree of MSc (insert MA, MSc, MBA, MScD, LLM etc, as appropriate)

Signed ..... (candidate) Date 20/09/2019

STATEMENT 2

This dissertation is the result of my own independent work/investigation, except where otherwise stated.
Other sources are acknowledged by footnotes giving explicit references. A Bibliography is appended.

Signed ..... (candidate) Date 20/09/2019

STATEMENT 3 – TO BE COMPLETED WHERE THE SECOND COPY OF THE DISSERTATION IS SUBMITTED IN AN APPROVED ELECTRONIC FORMAT

I confirm that the electronic copy is identical to the bound copy of the dissertation

Signed ..... (candidate) Date 20/09/2019

STATEMENT 4

I hereby give consent for my dissertation, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed ..... (candidate) Date 20/09/2019

STATEMENT 5 - BAR ON ACCESS APPROVED

I hereby give consent for my dissertation, if accepted, to be available for photocopying and for inter-library loans **after expiry of a bar on access approved by the Graduate Development Committee.**

Signed ..... (candidate) Date 20/09/2019

NODIADAU CYFARWYDDYD YNGHYLCH LLENWI FFURFLEN HYSBYSIAD O GYFLWYNO

Dylech lenwi dwy ochr y ffurflen mewn teip neu inc du a defnyddio priflythrennau. Ar ôl i chi lenwi'r ffurflen, rhaid ei chyflwyno i Bennaeth/Deon eich Ysgol (neu ei (h)enwebai) gyda dau gopi o'ch traethawd hir ac (os yw'n briodol) y ffi arholi.

Rhaid i bob copi o'r traethawd hir gynnwys:

1. crynodeb nad yw'n fwy na thri chant o eiriau;
2. gosodiad sydd wedi'i lofnodi gan yr ymgeisydd ac sy'n dangos i ba raddau y mae'r gwaith a gyflwynir yn ffrwyth ymchwiliad yr ymgeisydd ei hun, a chydnabyddiaeth benodol (gyda chyfeiriadau) o unrhyw ffynonellau eraill sydd wedi'u defnyddio;
3. llyfryddiaeth lawn;
4. datganiad sydd wedi'i lofnodi i dystio nad yw sylwedd y gwaith a gyflwynir wedi'i dderbyn ar gyfer unrhyw radd neu wobr arall, ac na chyflwynir ef yr un pryd mewn ymgeisyddiaeth ar gyfer unrhyw radd neu wobr arall;
5. gosodiad, sydd wedi'i lofnodi, ynghylch argaeledd y Traethawd Hir;
6. ffurflen 'Hysbysiad o Gyflwyno', a honno wedi'i llenwi.

Dylai'r datganiad a'r gosodiadau y cyfeiriwyd atynt yn 2-5 uchod gael eu hymgorffori ar ddechrau'r traethawd hir, fel y dangosir yn Atodiad 1.

Os yw hynny'n unol â pholisi'r Ysgol ar gyflwyno Traethodau Hir am Radd Athro a Addysgir:

- caiff ymgeiswyr gyflwyno'u gwaith i'w arholi mewn rhwymiad dros dro;
- gall ymgeiswyr gael eu cyfarwyddo gan yr Ysgolion i gyflwyno un o'r ddau gopi gofynnol mewn fformat electronig a gymeradwywyd.

ATODIAD 1: Patrwm Enghreifftiol ar gyfer tudalen y Datganiad/Gosodiadau sydd i'w chynnwys mewn Traethodau Hir am Radd Athro a Addysgir

RHIF ADNABOD YR YMGEISYDD	
CYFENW'R YMGEISYDD	**Rhowch gylch o gwmpas y teitl priodol** **Mr / Miss / Ms / Mrs / Y Parch / Dr / Arall – enwch ef ……..**
ENWAU BLAEN LLAWN YR YMGEISYDD	

DATGANIAD

Nid yw sylwedd y gwaith hwn wedi'i dderbyn o'r blaen ar gyfer unrhyw radd ac ni chyflwynir mohono mewn ymgeisyddiaeth ar gyfer unrhyw radd ar hyn o bryd.

Llofnodwyd ……(ymgeisydd) Dyddiad ……

GOSODIAD 1

Cyflwynir y traethawd hir hwn gan rannol gyflawni gofynion gradd ……(rhowch MA, MSc, MBA, MScD, LLM ac ati, fel y bo'n briodol)

Llofnodwyd ……(ymgeisydd) Dyddiad ……

GOSODIAD 2

Ffrwyth fy ngwaith/ymchwilio annibynnol fy hun yw'r traethawd hir hwn, ac eithrio lle y nodir fel arall.
Cydnabyddir ffynonellau eraill mewn troednodiadau sy'n rhoi cyfeiriadau manwl. Atodir Llyfryddiaeth.

Llofnodwyd ……(ymgeisydd) Dyddiad ……

GOSODIAD 3 – I'W LENWI OS CYFLWYNIR AIL GOPI O'R TRAETHAWD HIR MEWN FFORMAT ELECTRONIG A GYMERADWYWYD

Yr wyf yn cadarnhau bod y copi electronig yn union yr un fath â'r copi o'r traethawd hir sydd wedi'i rwymo.

Llofnodwyd ……(ymgeisydd) Dyddiad ……

GOSODIAD 4

Rhoddaf ganiatâd drwy hyn i'm traethawd hir, os derbynnir ef, fod ar gael i'w lungopïo ac i'w fenthyca rhwng llyfrgelloedd, ac i'r teitl a'r crynodeb fod ar gael i sefydliadau allanol.

Llofnodwyd ……(ymgeisydd) Dyddiad ……

GOSODIAD 5 - CYMERADWYWYD Y GWAHARDDIAD AR FYNEDIAD

Rhoddaf ganiatâd drwy hyn i'm traethawd hir, os derbynnir ef, fod ar gael i'w lungopïo ac i'w fenthyca rhwng llyfrgelloedd **ar ôl i'r gwaharddiad ar fynediad iddo, a gymeradwywyd gan y Pwyllgor Datblygu Graddedigion, ddod i ben.**

Llofnodwyd ……(ymgeisydd) Dyddiad ……

Gamifying Agile Software Development

Teamworking, Contribution Analysis

MSc Computing and IT Management

School of Computer Science and Informatics, Cardiff University

1819Place-CMT400

60 credits

Author: Yujiao Ye

Student Number: C1837071

Supervisor: James Osborne

Moderator: Carl Jones

1st September 2019

Abstract

Working with staff at the National Software Academy, you will come up with a way to ensure that students work effectively on a range of team projects. It is a well-known problem that some students refuse to pull their weight when working as part of a team. Such students have a drag effect on those students who are keen to achieve the best possible results in their assessments. Those students who work hard often do not see a significant difference in their final results compared to those students who do not work hard and are able to blag their way past the assessment criteria. You will be responsible for coming up with a way to gamify the agile software development process so that students who work hard and those who do not are rewarded and penalised appropriately.

Acknowledgements

I would like to acknowledgement Dr James Osborne for excellent teaching in the field of agile and scrum team framework and provide a lot of information that I might need for this project.

TABLE OF CONTENT

1. Introduction

This project aims to design and build a website peer assessment survey which captures the data and stored in the database but does not process the data. Accordingly, they will get rewarded and punished grades from the peer assessment scores. The project will be tested by complete the peer assessment with individual accounts and a table output which contains students that are entered in advance and their scores for the client (Dr James Osborne).

In peoples' work and studies, there are not only individual assessments but also teamwork. For teamwork, the common problem is the uneven weight of work. An essential method to avoid the problem is finding a way to evaluate students by themselves and get a reward and punish grades.

This project is designed for the scrum teams. The scrum team is a kind of agile process, which has a master in every group. Also, groups have the planning, daily, client and retrospective meeting in every sprint. A sprint is a cycle of the amount of time (e.g. every week). Every team member has individual works in every race, and the backlog is a table of the tasks for every sprint, which contains the functions and the planned time for the assignment (Resnick 2011).

Peer assessment is a practical approach to encourage students for better grades. According to a study, the 233 undergraduate students in 2007 had more significant improvement compared to the students who have no peer assessment (Xiao andLucking 2008). Additionally, peer assessment within the teams is more direct and efficient than from the teachers (Topping 2009). Students will learn their strengths and weakness and then make alternate actions. Every student in a team should evaluate others except themselves. According to a journal named Psychological reports, the self-evaluation is biased compared to the peer evaluation (Tsai et al. 2014). Therefore, students will not peer-assessed themselves in this project.

The login and register system are designed for students (users) and teachers (administer). The students need to register with their student number, password, first name, last name, group number before they log in and do their peer assessment. As the same as teachers, except they need to register with their stuff number. The login system aims to ensure the data of their accounts is safe and not clashed to each other. They have a unique student or staff number, and hashed password stored in the database. For the teacher's account, the Excel CSV file, which contains student numbers and scores of each question, need to be downloaded from the database as a request. In the case, they need to process the data and not only see the scores. Numbers are different because of the count of numbers (students: 8 and teachers:6) to distinguish the role of users.

There are eight questions for each peer assessment, which four questions are designed for the scrum team meeting, two for team contribution and two for team communication. Every question has five options which present scores from 1 to 5. Students need to evaluate their group members through these questions and then submit the answers to the database. Once they have submitted, the success message is shown to the screens of students. For the teacher's account, the timestamp is displayed because the peer assessment is designed for every sprint.

Therefore, the project goal is capturing data from the peer assessment website, which include questions and information for the student, such as student number. Then output the data as an Excel file. As a result, the teacher could deal with the data to give a reasonable grade to each student.

2. Background

1) Online survey websites

For this project, the main design is a peer assessment of students, which is similar to a survey. There are many survey websites which could be used for this project, such as Survey Monkey (https://www.surveymonkey.com), Smart Survey (https://app.smartsurvey.co.uk), Survey Planet (https://app.surveyplanet.com), and so on.

In this section, the three sites will be described as primary and basic research. The login is required if the user wants to create a questionnaire, which is easy for users to find their surveys. However, for this project, only one questionnaire should be released for all students in different sprints. Also, themes and formats are pre-designed, which users could choose. All three website has charging plans (25 pounds to 75 pounds) for more themes and functions users might need, such as image or file questions and result analyse.

Also, themes and styles are pre-designed, which users could choose. All three website has charging plans (25 pounds to 75 pounds) for more themes and functions users might need, such as more responses and result analyse. Formats of questions are similar to each other, for example, checkboxes, radios, matrix (checkboxes and radios), ranking, dropdown menus, text area and date.

For outgiving questionnaires to responses, users need to copy the URL or send an email to them, and it is same for all three sites. The peer assessments should be released to students after every sprint. The process is complex and cumbersome because teachers should send the link to every student at the end of every sprint.

2) Website login

Logging-in system is a set of data to authenticate a user. Usually, the data contains a username, password. Sometimes, users could access the websites without logging in, but others not. As mentioned before, users could not use Survey Monkey, Smart Survey and Survey Planet if they do not log in. However, some shopping websites can access the website if they do not log in, such as when people view the items on Amazon, Tesco.

However, the logging-in system is necessary if people want to buy the items. The database stores the information of every order, such as order time, shipping information. As well as the three survey sites, all questions, answers and results are stored in the database.

3) Tools

Several tools were used in the project at designing process, database, coding and local server setting.

Balsamiq Mockups is computer software for wireframe website, application on computers and phones designing. It has simulated low-fidelity wireframe and a built-in component library to design a screen view rapidly. Adobe Photoshop is another popular wireframe design software, which is more professional than Balsamiq. However, Balsamiq is more straightforward and faster compared with Adobe Photoshop, which exactly satisfies the demands of this project.

For the database, several bench works could be used, such as MySQL, Oracle and PhpMyAdmin. MySQL and Oracle are all the products of Oracle company, but there are differences between them. The first difference is the price. MySQL is free for download and install, but Oracle Express Edition has limited functions compared to MySQL if users do not buy the Standard Edition. The second difference is that Oracle

supports four styles of character data (CHAR, VARCHAR2, NCHAR, NVARCHAR2) compared to MySQL has two (CHAR, VARCHAR). Also, Oracle has a particularly extensive storage features such as tablespace, whereas MySQL has not. The last but most important is that the Oracle can be used for enterprise level, but MySQL cannot. PhpMyAdmin is different from others, and it does not need to download and could be used at a browser as soon as the user has connected to the database. Also, PhpMyAdmin has bundled with several development packages such as MAMP and WAMP.

WAMP, MAMP and LAMP are three development bundles. They are almost same, but the operating system is different. WAMP is the abbreviation of 'Windows, Apache, MySQL and PHP', MAMP is 'Mac, Apache, MySQL and PHP', and LAMP is 'Linux, Apache, MySQL and PHP' (Prettyman 2016).

3. Products

1) Requirement

For this project, there are five requirements from the client. First of all, the registration system for students. The students should register their student number, first name, last name, group number, and password before they log in to the system. The second requirement is student need to be able to log in and log out with their student number and password. Next, the survey system is to grab the data from the website. Every student needs to value the other members of the same group. After catching the data, the client required to output the data for CSV file from the database. At last, a database and a local server need to be set up.

2) Design

Designing is significant because the requirements need to satisfy. There are five parts

in this section. These are questions for peer assessment, answers or options correspond to specific questions, concept wireframe designs, and colour consideration for the final design. This project is similar to an agile project, which means that the project needs to be reviewed to the client and then alter anything that the client wants.

The peer assessment questions can be divided into three aspects for the client requirement: the scrum team evaluation, team communication and team contribution. To evaluate the Scrum team, students should have a donation to every type of meeting. The meetings include the planning, daily scrum, client and retrospective meeting. For team communication, students should finish their work and merged it into the team master frequently. Also, every scrum team has their team agreement, and the teammates should be attached to it. The last aspect is the students' contribution to the group. The students should provide a positive contribution to coding and non-coding activities. Therefore, there are eight questions in total for the peer assessment, which four questions for the Scrum team evaluation, two questions for team contribution and communication. The final questions are:

1. Did the student contribute positively during the planning meeting?
2. Did the student contribute positively during the daily scrum meeting?
3. Did the student contribute positively during the client meeting?
4. Did the student contribute positively during the retrospective meeting?
5. Did the student make a positive contribution to coding activities?
6. Did the student make a positive contribution to non-coding activities?
7. Has the student merged work to master frequently?
8. Has the student stuck to the working agreement?

The other major feature of a peer assessment questionnaire is the answers. Through the request of the client, options should be scored from one to five for every question, which is easy for him to process the raw output data. Thus, each question has five

answers with a specific score. From question one to six and question eight in the list above, the answers will be the frequency words (never, rarely, sometimes, often, frequently). And the options of question seven is the frequency with exact time length, which is 1/week, 2/week, 3/week, 1/day and >1/day (Figure 3.2.2-12). At the concept designs, the options for some questions are the percentage from 0 to 100 per cent. The reason that the percentage type of answers was abandoned is unclear. The students may think the 0 per cent is 0 score in ordinary cognition, whereas the score of this option is one score. Besides, this type of answer does not meet client needs.

The concept designs for peer assessment are created through Balsamiq wireframe software. There are two formats of the survey. The first design (Appendices 1) is a long page with all eight questions, which has a field for three question types. The 'Submit' button is at the bottom of the page. And another design (Appendices 2) is a question for one page. Every page has a 'Previous' and 'Next' button, which except the last page has a 'Submit' button instead of 'Next'. For the ultimate plan, the second design is accepted because every question is demonstrated in a separate page, whereas the first design is a lengthy page with all the questions. The format of questions is different within two designs. The first design list eight questions once and a radio button table for the options. Each row in a table presents a student on the team. Another design is repeating the problems several times, except the name of students are different. For the final design, the first question format is chosen because the other design is wordy and a waste of time for reading them. The style of the answers is presented in the second design: radio buttons, checkboxes and a button bar for every answer. The radio buttons are standard for one choice questions, but the choice may be unclear. For example, if students want to choose the 'Sometimes' option, and the radio button of 'Often' is near to it, they may click 'Often' instead of 'Sometimes'. The second style is the checkboxes. For normal cognition, checkboxes are presented a multiple-answers question, which could confuse the students. The button bar could solve the problem entirely. Students can easily distinguish a chosen option by highlighting it, and when

they change their options, the highlight will return to the original statement.

After the wireframe design, the selection of colour was considered. The black, white and grey was one option for the website at first, but the problem is dull and monotonous concerning colours. Therefore, the colours of green and naked pink were chosen from the website 'Color Hunt' (https://colorhunt.co/). Also, the client was satisfied when the design performed to him.

1. Register and login

The registration interface is required for further functions. The students need to fill in the necessary information. On the left of both register and login page header bar, the status of the student is presented (logged out). As a result, the scored can be stored in the database in their account. Figure 3.2.1-1 and Figure 3.2.1-2 are the signup interface with and without inputs. After the register, the website will automatically jump to the login page. Figure 3.2.1-3 is the interface which shows the massage.

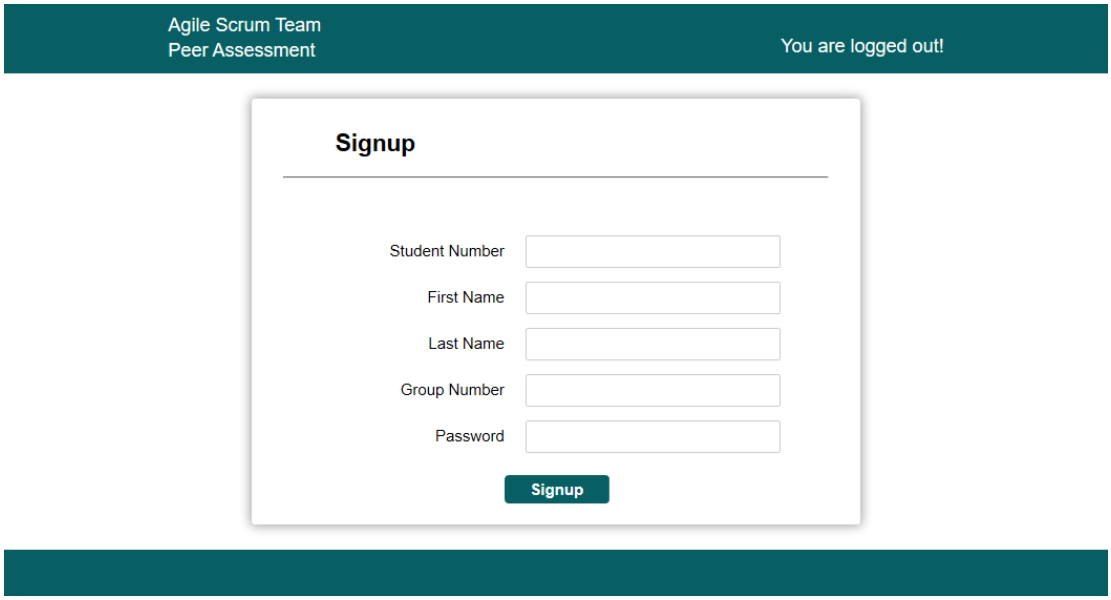


Figure 3.2.1-1

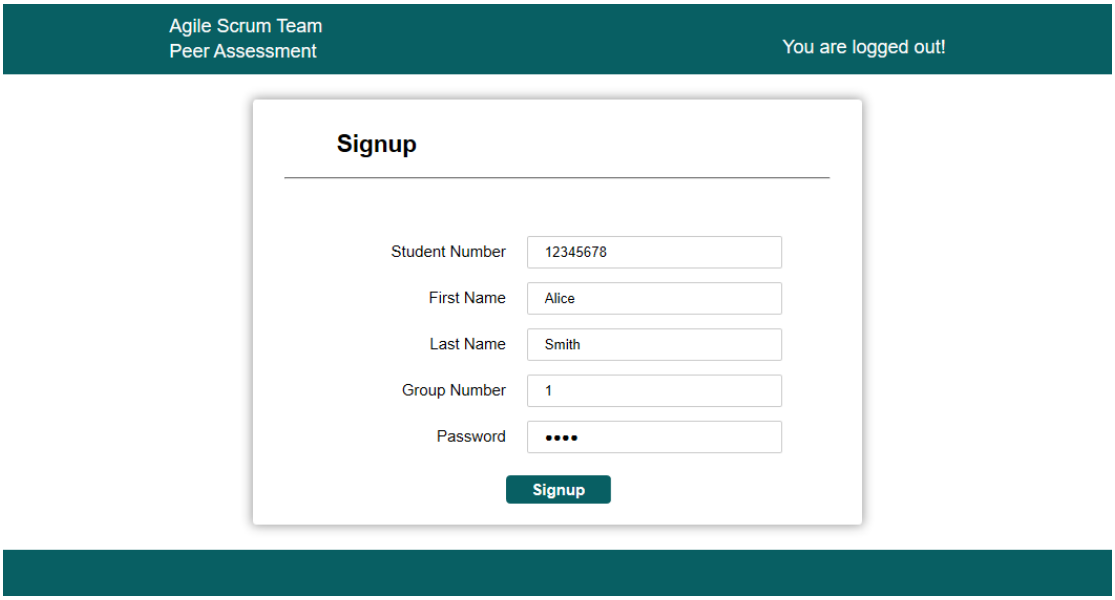


Figure 3.2.1-2

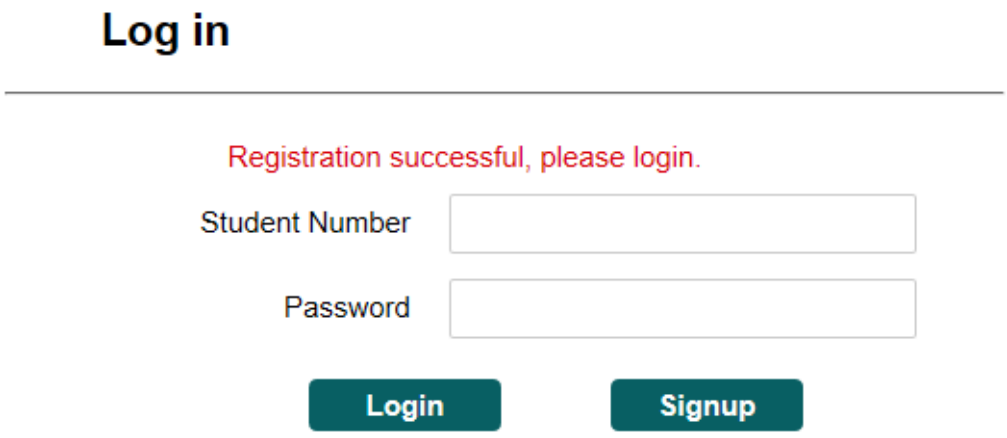


Figure 3.2.1-3

Figure 3.2.1-4 is the two errors during the registration. When the errors appeared, there will be a red text at the top of the input field. On the right is empty fields, which will perform when students miss one or some of the inputs and click the 'Signup' button. Also, the massage of 'Invalid student number' means that the student number has been taken when they register the account.

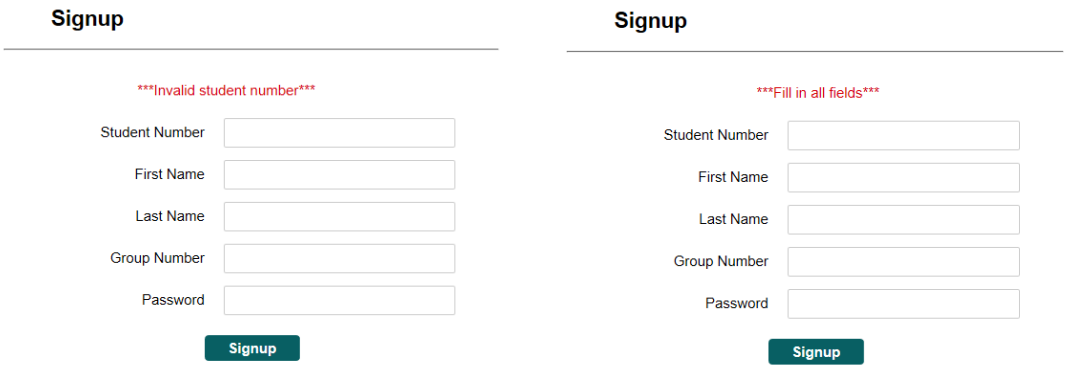


Figure 3.2.1-4

The login interface is similar to the registration, expect there just two inputs. Figure 3.2.1-5 and Figure 3.2.1-6 are the login interface, where the password cannot be seen after typing. Students could log in with their unique student number and password. Also, there is a button which links to the registration site if the student has no account. After typing the inputs, the student can log in by clicking the 'Login' button or press Enter the keyboard. If the student successfully logged into the site, the page will again automatically jump to the peer assessment site (Figure 3.2.1-7).

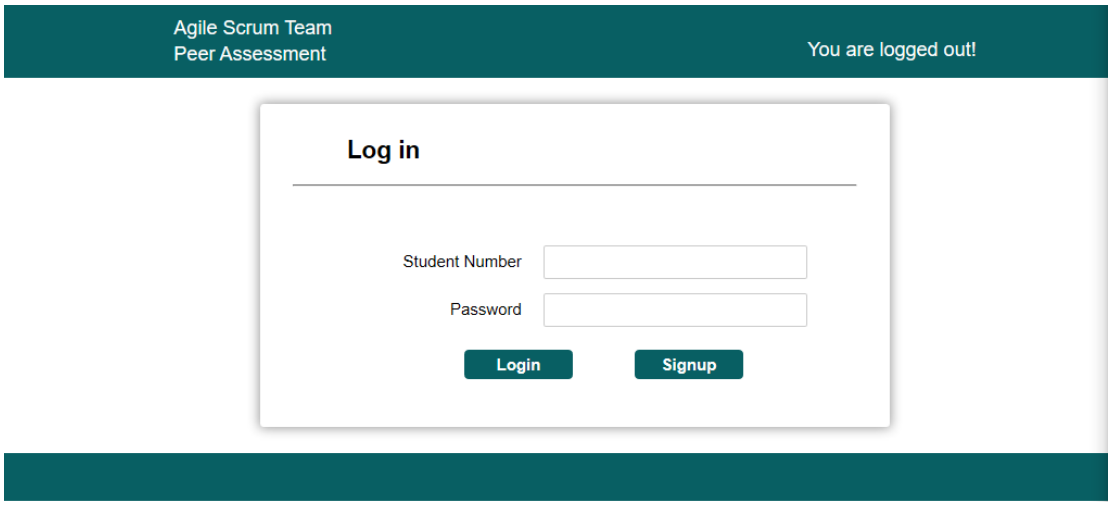


Figure 3.2.1-5

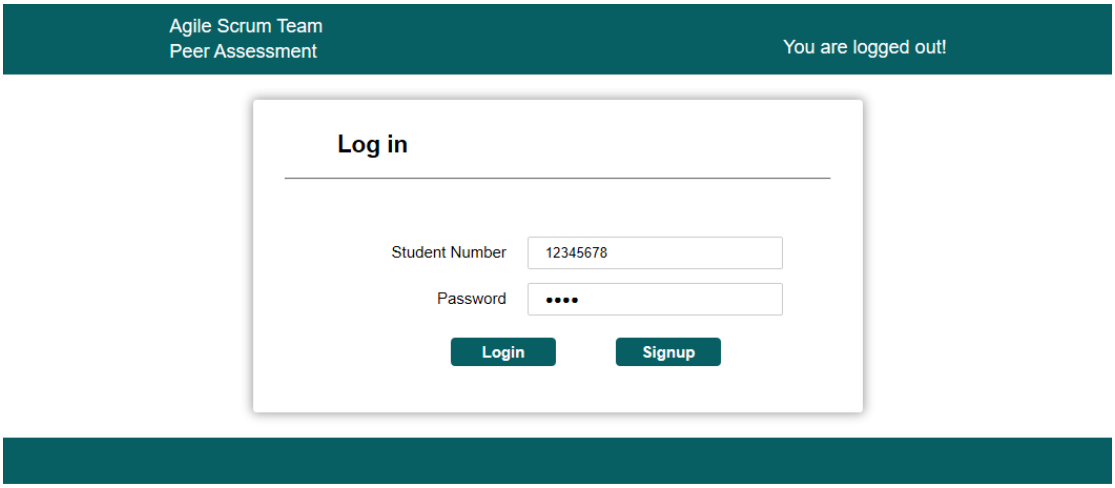


Figure 3.2.1-6

The two errors pages are the same as the register page. On the left side of Figure 3.2.1-7 is for the empty fields error, and when the password is different from the stored password, the right side of the figure will appear.

Figure 3.2.1-7

2. Peer assessment survey

After a student logged in, the peer assessment will be displayed (Figure 3.2.2-8). However, if there is no account logged in, the page will automatically jump to the login page. The system is differentiated students and teachers by the length of the numbers. Students have eight-digit codes, whereas teachers get six.

A title of the website (Agile Scrum Team Peer Assessment) is at the left of the header bar, which is the same as register and login pages. On the right, the login status and students first and last name are presented after the login state. The 'Logout' button is

in the header bar under the text, and when a student clicks the button, he or she will be logged out and jump to the login page.

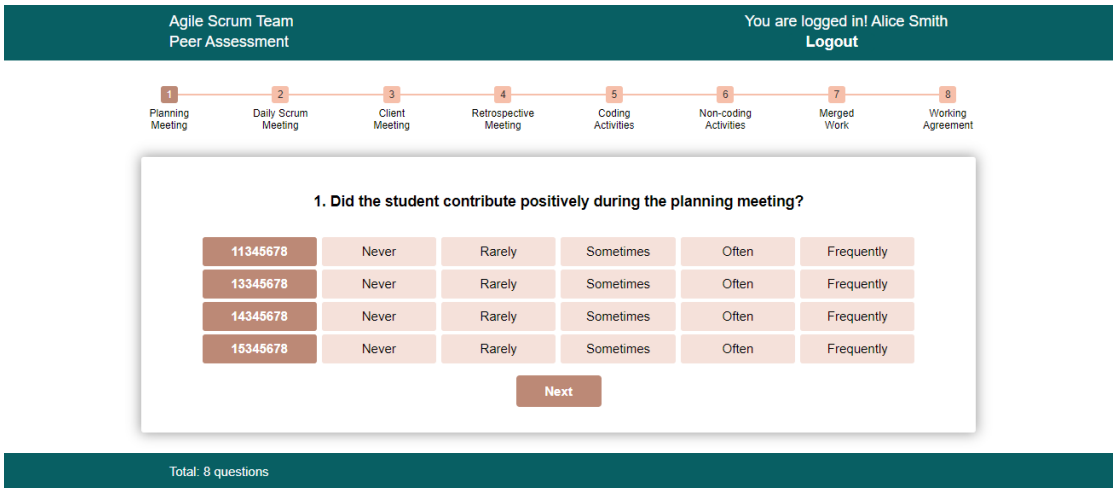


Figure 3.2.2-8

A progress bar, which contains the question number and the keywords for eight questions, is displayed at the top of this webpage. An active state is the darker colour for the icons, which means the student has finished or is doing the question. The last active state icon is the question that the student stays temporarily. Thus, the first question is active state all the time, and once the student clicks the 'Next' button, the next stage change to a darker colour.

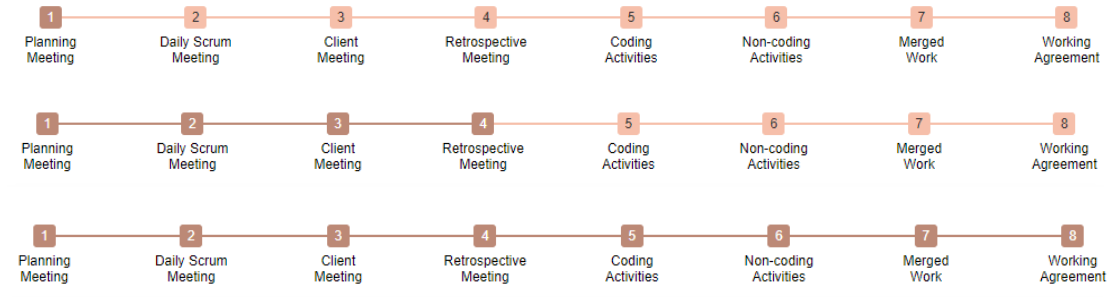


Figure 3.2.2-9

The field box under the process bar is a table of peer assessment question, answers and the corresponding student needed to be valued. The peer assessment question is above the table, and a button is below for the next issue. The first column with a darker

colour background is the student number for each teammate. The reason for not displaying student names is that there may be duplicate names in one group. Also, the length of every student names is different, and it may cause disorderly for the table. The answers are corresponding to the student number in the first column. The answers could be placed at the top of the column and change the text area to radio buttons. However, they are displayed in every cell because students could choose cells more accurately when they determine the answers.

The animation of answer buttons is different from the function buttons (such as 'Next', 'Previous', and 'Submit'). When the cursor hovers over the answers, the colour of the text remains black, and the background will change slightly darker compared to the original, such as the second line of Figure 3.2.2-10. Once a student chose the answer 'Frequently' (first line in Figure 3.2.2-10), the text change to white, and get the darkest background with an outer circle. The hover and active state of the function buttons are the same. In Figure 3.2.2-11, the 'Next' button is focused and just get an outer circle compared to the answer buttons.

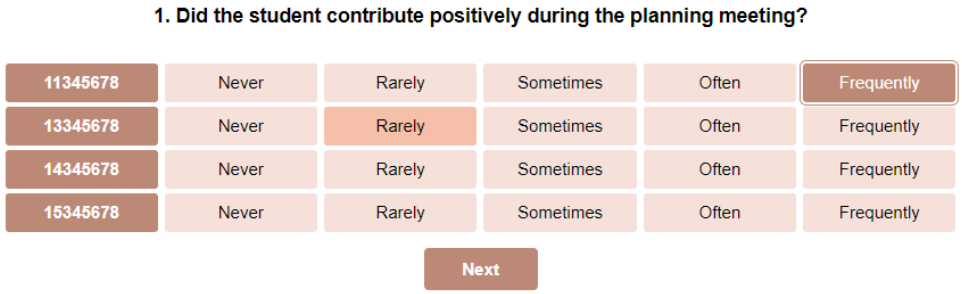


Figure 3.2.2-10

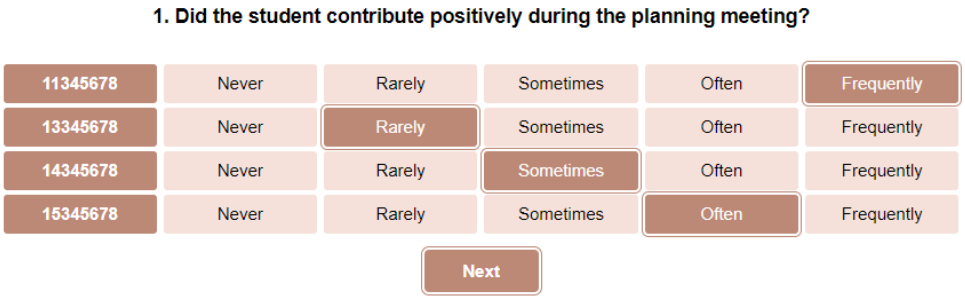


Figure 3.2.2-11

Once the student completes the first question and clicks the 'Next' button, the page will jump to the next question. Being an example that the student just finished the sixth question, the seventh is displayed in the filed box, as showed in Figure 3.2.2-12. The function button becomes two of them: 'Previous' and 'Next'. The previous answer remains the same if the student clicks the 'Previous' button. As mentioned before, the answers to the seventh question are different from others - Figure 3.2.2-12. The 'Next' button changes to 'Submit' the whole peer assessment at the last question (Figure 3.2.2-13).

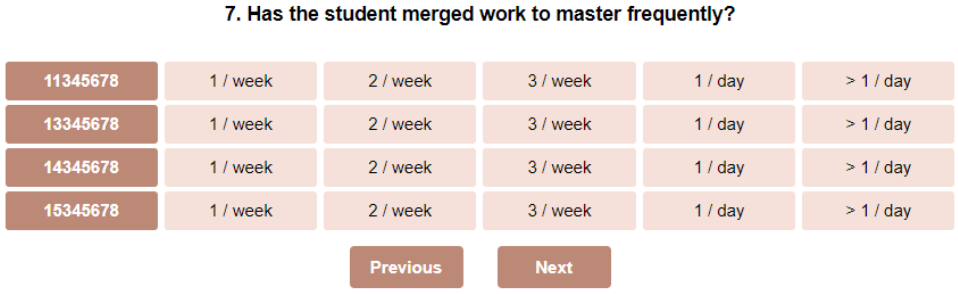


Figure 3.2.2-12

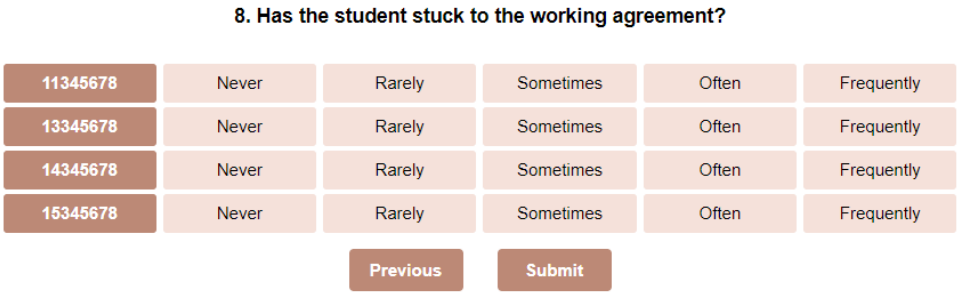


Figure 3.2.2-13

After students submit the peer assessment, the massage of successful submit (Figure 3.2.2-14) will appear at the page. Also, a 'Logout' button is under the massage, and the button will lead the student back to the login page.

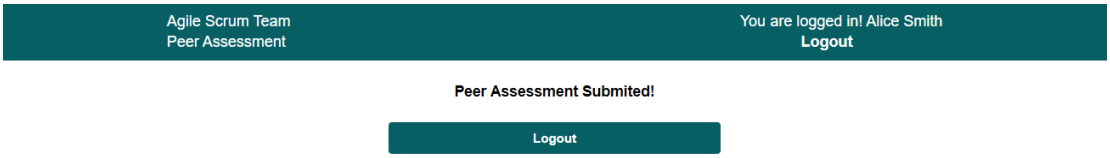


Figure 3.2.2-14

There is a situation that the 'Empty answer(s)' error message is performed. When the student does not choose all the answers, the error message will be displayed before the table. Additionally, the page will not remember the answers, and the student needs to choose the answers again.

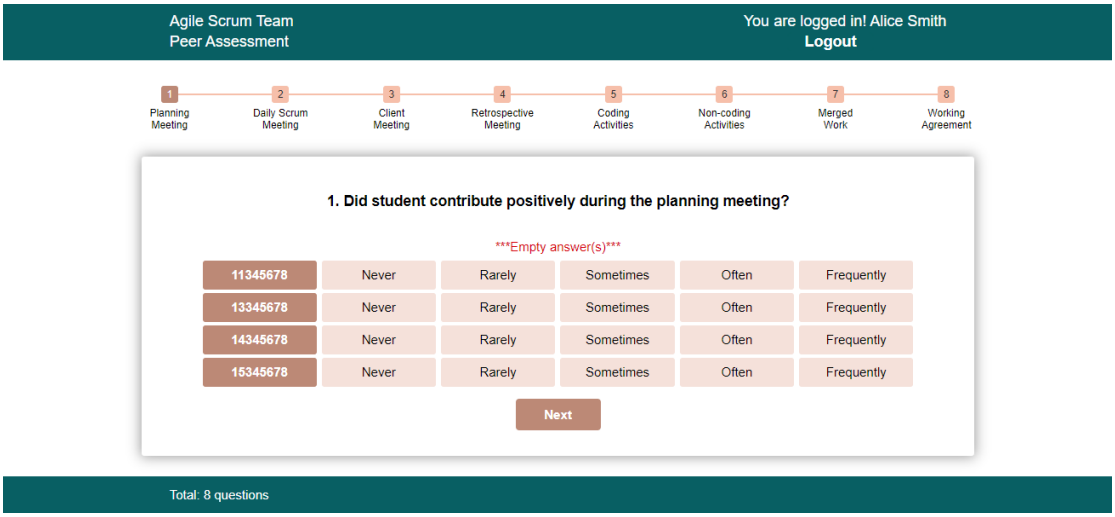
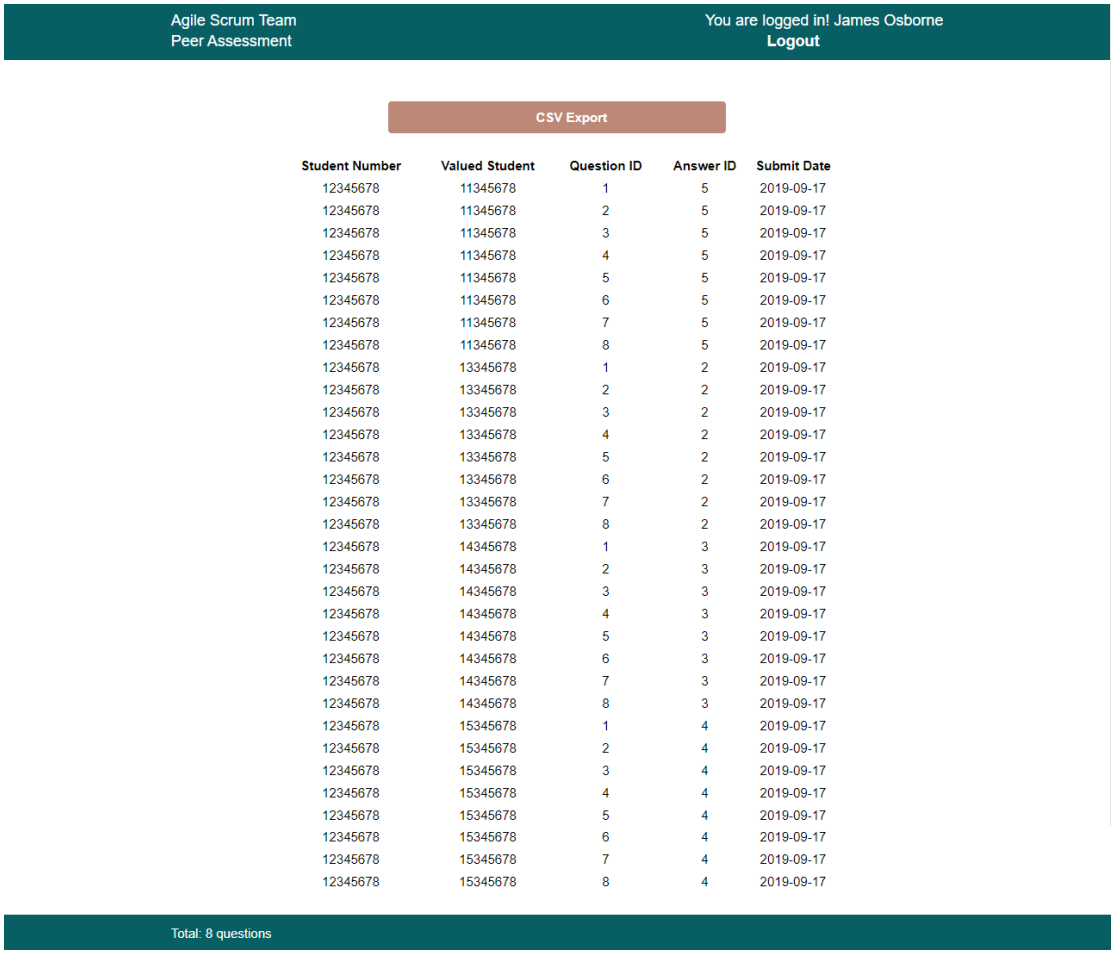


Figure 3.2.2-15

3. Data preview and export

In this case, after a client (teacher) logs into the system by the six-digit number with a password, the preview of the student peer assessment data (Figure 3.2.3-16) will be displayed on the screen.

The position of the column orders the table, which is student number, valued student, question ID, answer ID and submission date. The first column is the student who logged in the system, and the valued student column is the student's teammates who should be peer-assessed.



Student Number	Valued Student	Question ID	Answer ID	Submit Date
12345678	11345678	1	5	2019-09-17
12345678	11345678	2	5	2019-09-17
12345678	11345678	3	5	2019-09-17
12345678	11345678	4	5	2019-09-17
12345678	11345678	5	5	2019-09-17
12345678	11345678	6	5	2019-09-17
12345678	11345678	7	5	2019-09-17
12345678	11345678	8	5	2019-09-17
12345678	13345678	1	2	2019-09-17
12345678	13345678	2	2	2019-09-17
12345678	13345678	3	2	2019-09-17
12345678	13345678	4	2	2019-09-17
12345678	13345678	5	2	2019-09-17
12345678	13345678	6	2	2019-09-17
12345678	13345678	7	2	2019-09-17
12345678	13345678	8	2	2019-09-17
12345678	14345678	1	3	2019-09-17
12345678	14345678	2	3	2019-09-17
12345678	14345678	3	3	2019-09-17
12345678	14345678	4	3	2019-09-17
12345678	14345678	5	3	2019-09-17
12345678	14345678	6	3	2019-09-17
12345678	14345678	7	3	2019-09-17
12345678	14345678	8	3	2019-09-17
12345678	15345678	1	4	2019-09-17
12345678	15345678	2	4	2019-09-17
12345678	15345678	3	4	2019-09-17
12345678	15345678	4	4	2019-09-17
12345678	15345678	5	4	2019-09-17
12345678	15345678	6	4	2019-09-17
12345678	15345678	7	4	2019-09-17
12345678	15345678	8	4	2019-09-17

Figure 3.2.3-16

The animation of the 'CSV Export' button is the same as the function buttons in the

peer assessment survey page. The focused and hover-over state are the same, which is showed in Figure 3.2.3-17.

Figure 3.2.3-17

3) Implementation

1. Database structure

The database is created into five tables (Figure 3.3.1-18): students, questions, answers, question-answer and data. The software of the database is PhpMyAdmin, which is bundles to MAMP. All the five tables are stored in the local database 'pa', as the Figure showed.

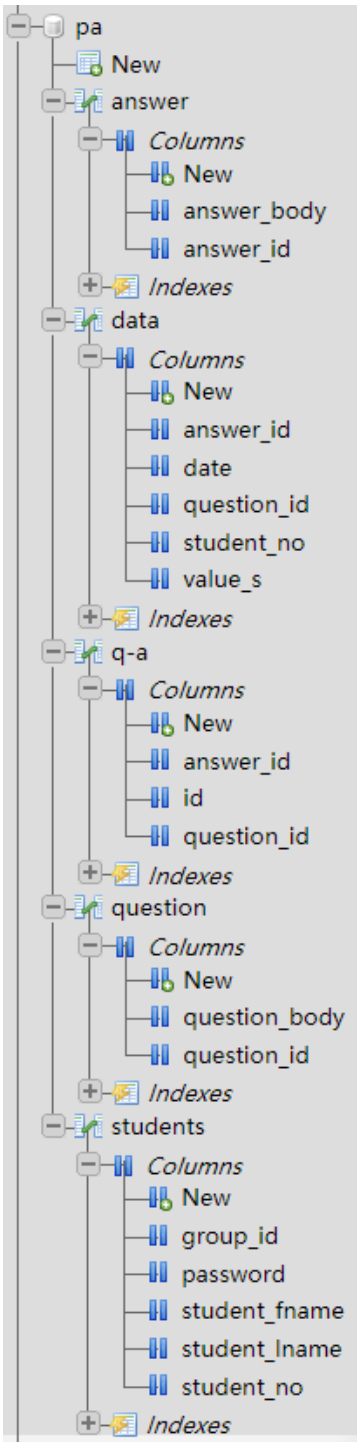


Figure 3.3.1-18

One of the tables is a one-to-many database structure: students. There are five attributes in the table, and the primary key is student number because every student has a unique student number. The information of students is stored in the database through the registration website. The other attributes are the first and last name, group number and hashed password.

Questions and answers are designed to a many-to-many database structure. This table is created for the peer assessment question display. According to the tips of the book (Teorey 2011), the ER (Entity Relation) model needs to be simple for not wasting time and hard communication. Therefore, the table of question-answer is designed to two one-to-many database relationships. The questions and answers table are similar to each other. Question/answer ID and question/answer body are the attributes, and the primary key is question/answer ID. The question-answer table is to record the questions with the corresponding answer,

where the foreign key is the question/answer ID in questions/answers table. The details of the database tables will appear in the appendix.

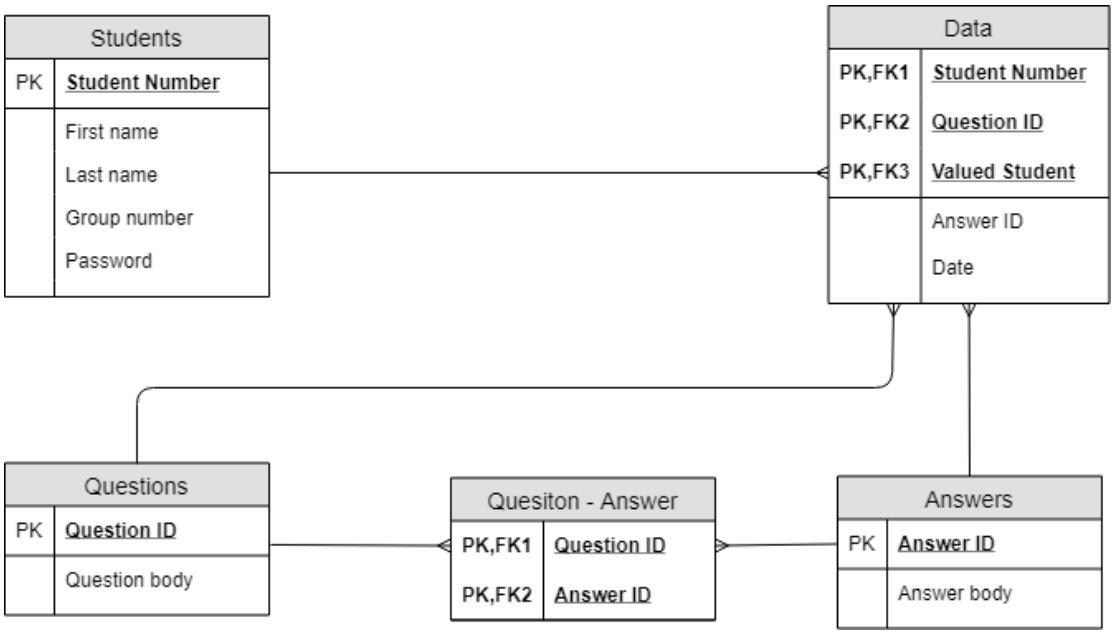


Figure 3.3.1-19

Last but not least, the data table records the students' peer assessment scores for every teammates and question. There are three foreign keys for this table, where the question ID is the primary keys from the questions table. The student number is the student number of the logged-in user, and the valued student is the group member at the same group. The date attribute is the current timestamp when the user submits the peer assessment. For example, a student logged in with his number (12345678), and student number of her four teammates are 11345678, 13345678, 14345678 and 15345678. Then the preview (Figure 3.2.3-16) and export CSV file (Table 3.3.1-1) data table should be like the table below:

Table 3.3.1-1

Student Number	Valued Student	Question ID	Answer ID	Submit Date
12345678	11345678	1	5	2019/9/17
12345678	11345678	2	5	2019/9/17
12345678	11345678	3	5	2019/9/17
12345678	11345678	4	5	2019/9/17
12345678	11345678	5	5	2019/9/17

12345678	11345678	6	5	2019/9/17
12345678	11345678	7	5	2019/9/17
12345678	11345678	8	5	2019/9/17
12345678	13345678	1	2	2019/9/17
12345678	13345678	2	2	2019/9/17
12345678	13345678	3	2	2019/9/17
12345678	13345678	4	2	2019/9/17
12345678	13345678	5	2	2019/9/17
12345678	13345678	6	2	2019/9/17
12345678	13345678	7	2	2019/9/17
12345678	13345678	8	2	2019/9/17
12345678	14345678	1	3	2019/9/17
12345678	14345678	2	3	2019/9/17
12345678	14345678	3	3	2019/9/17
12345678	14345678	4	3	2019/9/17
12345678	14345678	5	3	2019/9/17
12345678	14345678	6	3	2019/9/17
12345678	14345678	7	3	2019/9/17
12345678	14345678	8	3	2019/9/17
12345678	15345678	1	4	2019/9/17
12345678	15345678	2	4	2019/9/17
12345678	15345678	3	4	2019/9/17
12345678	15345678	4	4	2019/9/17
12345678	15345678	5	4	2019/9/17
12345678	15345678	6	4	2019/9/17
12345678	15345678	7	4	2019/9/17
12345678	15345678	8	4	2019/9/17

2. Path

Figure 3.3.2-20 shows the path of the website folder. As mentioned before, the language of static webpages is HTML and CSS, and then they have been changed to dynamic websites using PHP language.

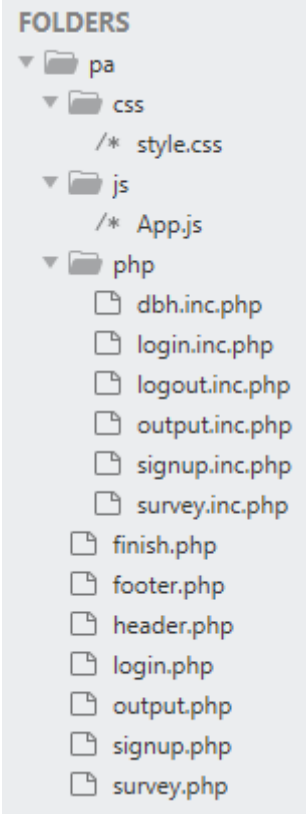


Figure 3.3.2-20

Under the root folder (pa), there are seven frontends for the site. The 'header.php' and 'footer.php' are used in every page for a header and footer. Other files in the 'pa' folder are the main content, as shown before in the Figures of the Design section.

The 'css' folder is all the styles used for the webpage. Also, the 'APP.js' in 'js' folder is the animation that the webpage needs, such as when students click the 'Next' button, the question will jump to the next. The 'style.css' file has linked to the header page, as the same as the JavaScript file.

The 'php' folder is the backend for every webpage. The 'dbh.inc.php' file is the setting of server and database connection. To running the script, the local server and database should be connected. The later than 5.6 PHP connection code is given in the MAMP default website, and the database of the project has been changed to ‘pa’ (Figure). Additionally, the other PHP files under this folder are using for processing the data from the database, such as select and insert.

```php
$user = 'root';
$password = 'root';
$db = 'pa';
$host = 'localhost';
$port = 3307;

$link = mysqli_init();
$success = mysqli_real_connect(
   $link,
   $host,
   $user,
   $password,
   $db,
   $port
);
```

Figure 3.3.2-21

3. Header and footer

The header and footer for all pages are the same, but with different login states. The login state is confirmed by the Session code which is in the 'login.inc.php' file and will be described below in the Login and logout section. The 'session_start' is at the start of the coding, which determines the 'session' function is beginning to use. Otherwise, the function will not work.

There are three JavaScript libraries used in the header and footer coding, which is displayed in Figure 3.3.3-22. Also, a style sheet linked to the 'header.php' file and a JavaScript file linked to the 'footer.php' file.

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/jquery.min.js">
</script>
<script src="https://code.jquery.com/jquery-1.11.0.min.js" integrity="
sha256-spTpc4lvj4dOkKjrGokIrHkJgNA0xMS98Pw9N7ir9oI=" crossorigin="anonymous">
</script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/
jquery-easing/1.4.1/jquery.easing.min.js"></script>
```

Figure 3.3.3-22

After linking libraries and the style sheets, there are two and three situations in 'header.php' and 'footer.php' respectively.

The first situation in 'header.php' is that the student has logged in, which is a login status, the name of the student or teacher and a logout button will appear in the top-left corner. If the state is not logged in, the message of 'You are logged out!' will appear. The 'if-else' code and an 'isset' code has been used in these situations (Figure 3.3.3-23).

```php
<?php
if (isset($_SESSION['student_no'])) {
    echo '<p class="status">You are logged in! ' . $_SESSION['
        student_fname'] . ' ' . $_SESSION['student_lname'] . '</p>';
    echo '<form action="php/logout.inc.php" method="post"><button
        type="submit" name="logout_submit">Logout</button></form>';
}
else {
    echo '<p class="status" style="margin-top: 20px;">You are logged
        out!</p>';
}
?>
```

Figure 3.3.3-23

The three situations in 'footer.php' are student logged in, teacher logged in and logged out. The code is similar to the header file, but with an 'if' statement for the length of the student or teacher number (strlen($_SESSION['student_no']) inside the 'isset' statement (Figure 3.3.3-24).

```php
<?php
if (isset($_SESSION['student_no'])) {
    if (strlen($_SESSION['student_no']) == 8) {
        echo '<footer><div class="pull-left">Total: 8 questions</div></footer>';
    } else {
        echo '<footer><div class="pull-left"> </div></footer>';
    }
} else {
    echo '<footer><div class="pull-left"> </div></footer>';
}
?>
```

Figure 3.3.3-24

4. Registration

The first main section is the register system, which is an input form to record the information of students. The form using 'POST' method to grab data from the input area in the 'signup.php' file. The password input cannot be seen after the student typed in the text. The 'Signup' is a function button, which pushes the data into the backend file to process the next steps.

The 'signup.inc.php' is the backend file for this page. At first, the database connection is required before process the data. Then, the system grabs the input text that students already filled in the form using '$_POST[]' function. The password also cannot be seen in the database with the 'password_hash()' function.

```
$student = $_POST['student_no'];
$fname = $_POST['student_fname'];
$lname = $_POST['student_lname'];
$group = $_POST['group_id'];
$password = $_POST['pwd'];
$hashpwd = password_hash($password, PASSWORD_DEFAULT);
```

Figure 3.3.4-25

The next step is to verify if there is an empty input. To achieve it, the 'if' function is used, as the Figure 3.3.4-26. The next level is to verify if there is student number has been registered or not. If there is one error appeared, the error message will show on the page, as the Figure 3.2.1-4 in the Design section. The '$GET[]' function is used for the delivering of the error messages in the 'signup.php' file.

```
<?php
if (isset($_GET['error'])) {
    if ($_GET['error'] == emptyfields) {
        echo '***Fill in all fields***';
    }
    else if ($_GET['error'] == wrongnumber) {
        echo '***Invalid student number***';
    }
}
?>
```

Figure 3.3.4-26

If neither of the error occurs, the information from the input will record into the database. The code is presented in Figure 3.3.4-27. Also, the system is to test the connection to the database first. After all the process finished, the page will jump to the login system for further steps.

```php
if (empty($student) || empty($fname) || empty($lname)  || empty($group) ||
    empty($password)){
    header("Location: ../signup.php?error=emptyfields");
    exit();
}
else{
    $sql = mysqli_query($link, "SELECT * FROM `students` WHERE student_no = "
        . $student .";");
    if (mysqli_num_rows($sql) > 0) {
        header("Location: ../signup.php?error=wrongnumber");
        exit();
    }
    else {
        $sql = "INSERT INTO `students` (student_no, student_fname,
            student_lname, group_id, password)
                VALUES ('". $student . "', '" . $fname. "', '" . $lname . "
                    ', '" . $group . "', '" . $hashpwd . "')";
        $stmt = mysqli_stmt_init($link);
        if (!mysqli_stmt_prepare($stmt, $sql)) {
            header("Location: ../signup.php?error=sqlerror");
            exit();
        } else {
            $result = $link->query($sql);
            header("Location: ../login.php?seccess");
            exit();
        }
    }
}
```

Figure 3.3.4-27

5. Login and logout

The login page is similar to the register website, but with two function buttons: login and sign up. The method of the form in 'login.php' is 'POST' and 'isset' function in the 'login.inc.php' backend file to receive and process the data.

The first two levels of coding in 'login.inc.php' file are to verify the empty field error and invalid password error. To check the password, the 'password_verify()' function is used to compare the hashed password stored in the database.

If the password is correct, the 'SESSION' function starts ('session_start()'). The information about students' number, names, and group id are copied into the 'SESSION' variable ($_SESSION[]). As a result, the session function can be used in the other files without pull the data out again.

After that, the system will check the number of student or teacher by checking the length of the numbers. As mentioned before, the student number is eight-digit, and teachers get six. If the number is eight-digit, the page will jump to the 'survey.php' file, otherwise, the 'output.php' file. The full code will be found in Figure 3.3.5-28 below.

```php
<?php
if (isset($_POST['login_submit'])){

    require "dbh.inc.php";

    $student = $_POST['student_no'];
    $password = $_POST['pwd'];

    if (empty($student) || empty($password)){
        header("Location: ../login.php?error=emptyfields&student_no=".$student);
        exit();
    }
    else {
        $sql = mysqli_query($link, "SELECT * FROM `students` WHERE student_no =
            ". $student);
        if ($row = mysqli_fetch_assoc($sql)) {
            $check = password_verify($password, $row['password']);
            if ($check == false) {
                header("Location: ../login.php?error=wrongpwd");
                exit();
            } elseif ($check == true) {
                session_start();
                $_SESSION['student_no'] = $row['student_no'];
                $_SESSION['student_fname'] = $row['student_fname'];
                $_SESSION['student_lname'] = $row['student_lname'];
                $_SESSION['group_id'] = $row['group_id'];
                if (strlen($row['student_no']) == 6) {
                    header("Location: ../output.php");
                    exit();
                } else {
                    header("Location: ../survey.php");
                    exit();
                }
            }
        }
    }
}
else{
    header("Location ../login.php");
    exit();
}
```

Figure 3.3.5-28

The logout is simpler than the register and login system. The 'Logout' button is inside of a ‘form’ tag. The 'session_unset()' and 'session_destroy()' functions are used for logout the system. Once the students or teachers click the logout button, the web page will hop back to the login page (Figure 3.3.5-29).

```php
session_start();
session_unset();
session_destroy();
header("Location: ../login.php");
```

Figure 3.3.5-29

6. Peer assessment survey

The frontend of the page (survey.php) has two main sections, the process bar and the survey question field. The header, footer and database files are required for the page. The page will automatically jump to the login page if there has a logout status.

The progress bar is a list of text (Figure). The animation that mentioned above is created by the 'style.css' and 'APP.js'. The class of activated progress bar is styled first in the CSS style sheet, and then add the class 'active' by the JavaScript file once the student clicks the 'Next' button. Also, the 'Previous' button is the same, except the activated class is removed from the list tags.

```html
<ul id="progressbar">
    <li class="active">Planning<br>Meeting</li>
    <li>Daily Scrum<br>Meeting</li>
    <li>Client<br>Meeting</li>
    <li>Retrospective<br>Meeting</li>
    <li>Coding<br>Activities</li>
    <li>Non-coding<br>Activities</li>
    <li>Merged<br>Work</li>
    <li>Working<br>Agreement</li>
</ul>
```

Figure 3.3.6-30

Every question need a while loop for all of the teammates depends on how many people in a team. To achieve that, the information stored in the 'students' table need to selected with the '$_SESSION[]' function for the same group ID. Then, the student numbers in the survey website table should not include the current login student number (Figure 3.3.6-31).

```php
$sqlg = mysqli_query($link, "SELECT * FROM `students` WHERE
    group_id = " . $_SESSION['group_id']);
while ($rowg = mysqli_fetch_assoc($sqlg)){
    echo '<div class="row">';
    if ($rowg['student_no'] !== $_SESSION['student_no']) {
    }
    echo '</div>';
}
```

Figure 3.3.6-31

The error section is for the empty answers of not finished peer assessment (Figure 3.3.6-32). The 'GET' function is similar to the error messages of the register and login page.

```
if (isset($_GET['error'])) {
    if ($_GET['error'] == emptyanswer) {
        echo '<p class="error">***Empty answer(s)***</p>';
    }
}
```

Figure 3.3.6-32

Furthermore, every valued student needs a while loop for the answers in each line depends on the question ID and answer ID. The database has a table named 'q-a' which is described in the database section. The variable '$sqla' is for calling the answer bodies in the survey website tables with the question and answer ID (Figure 3.3.6-33). Also, the variable '$i' is for the answer output. In the 'while' loop, the variable added one first compered to the default (add one at the end). The total number of answers for each question is five. Therefore, the variable '$i' is between one to five, which corresponds to the client needs: the answers data should stay between one to five. Also, the name and value of the input tag are arrays and independent for every valued student and questions.

```
$sqla = mysqli_query($link, "SELECT q.question_id AS id, a.answer_body
    AS answer FROM `q-a` AS qa, `question` AS q,`answer` AS a WHERE q.
    question_id = qa.question_id and a.answer_id = qa.answer_id");
$i = 0;
while ($rowa = mysqli_fetch_assoc($sqla)) {
    if ($rowa['id'] == $rowq["question_id"]) {
        $i++;
        $student = $_SESSION['student_no'];
        $valued = $rowg['student_no'];
        $question = $rowq["question_id"];
        echo '<label class="choice"><input id="answer" type="radio"
            name="answer[' . $student . $valued . $question . $i . ']"
            value="' . $student . ',' . $valued . ',' . $question . ','
            . $i . '"/>' . $rowa['answer'] . '<span
            class="checkmark"></span></label>';
    }
}
```

Figure 3.3.6-33

After the tables, there are three buttons which mentioned in the Design section: 'Next',

'Previous' and 'Submit'. All code of the form in the 'surver.php' file is in Figure 3.3.6-34where the table and error sections are compressed.

```
$sqlq = mysqli_query($link, "SELECT * FROM `question`");
if ($sqlq -> num_rows > 0){
    while ($rowq = mysqli_fetch_assoc($sqlq)){
        echo '<fieldset class="questionForm" data-question="' . $rowq
            ["question_id"] . '">';
        echo '<h3>' . $rowq["question_id"] . '. ' . $rowq["
            question_body"] . '</h3>';

        if (isset($_GET['error'])) {...
        }

        $sqlg = mysqli_query($link, "SELECT * FROM `students` WHERE
            group_id = " . $_SESSION['group_id']);
        while ($rowg = mysqli_fetch_assoc($sqlg)){...
        }
        if ($rowq['question_id'] !== '1') {
            echo '<input type="button" name="previous"
                class="previous action-button" value="Previous"/>';
        }
        if ($rowq['question_id'] == '8') {
            echo '<input type="submit" name="survey_submit"
                class="action-button" value="Submit" />';
        } else {
            echo '<input type="button" name="next" class="next
                action-button" value="Next" />';
        }
        echo '</fieldset>';
    }
}
```

Figure 3.3.6-34

As mentioned before, the value of the radio button input is arrays and independent. Thus, the variables of '$_POST[]' will be segmented by commas and the 'foreach' function is used for every answer. The code is in the Figure 3.3.6-35.

```
$answers = $_POST['answer'];
foreach ($answers as $key => $value) {
    $items = explode (",", $value);
    $array[] = $items;
}
```

Figure 3.3.6-35

The insert function is to insert the value by order of data type inside the array. The insert SQL query code is displayed in the Figure 3.3.6-36. The date is the current timestamp, which is set when the table created. After the data was inserted into the database, the page will jump to the successful page ('finish.php').

```php
foreach ($array as $row => $value) {
    $student = mysqli_real_escape_string($link, $value[0]);
    $valued = mysqli_real_escape_string($link, $value[1]);
    $question = mysqli_real_escape_string($link, $value[2]);
    $answer = mysqli_real_escape_string($link, $value[3]);
    $sql = "INSERT INTO `data` (student_no, value_s,
        question_id,   answer_id) VALUES ('". $student . "', '
        " . $valued . "', '  " . $question . "', '"  . $answer
          . "')";
    mysqli_query($link, $sql);
}
header("Location: ../finish.php");
exit();
```

Figure 3.3.6-36

Two errors may appear on the page. The first error is that the student may not finish the peer assessment. To avoid the error, the length of the array is constrained with an 'if' function. If the number of teammates is different from the array, the data will be denied. The second situation is that the student does not choose any of the answers. The data is not an array if there is no data transfer from the website. The full function code is in the Figure 3.3.6-37 where the insert function is compressed.

```php
if (isset($_POST['survey_submit'])){

    $answers = $_POST['answer'];
    foreach ($answers as $key => $value) {
        $items = explode (",", $value);
        $array[] = $items;
    }

    function insert($array, $link){
        if (is_array($array)) {

            $sql = mysqli_query($link, "SELECT * FROM `students` WHERE
                group_id = ". $_SESSION['group_id'] .";");
            if ( (count($array)) !== ((mysqli_num_rows($sql)-1)*8) ){
                header("Location: ../survey.php?error=emptyanswer");
                exit();
            } else {
                foreach ($array as $row => $value) { ...
                }
                header("Location: ../finish.php");
                exit();
            }
        } else {
            header("Location: ../survey.php?error=emptyanswer");
            exit();
        }
    }
    insert($array, $link);
}
```

Figure 3.3.6-37

7. Preview and Export data

The output page also required the header, footer and the database connection files. As the same as the survey page, if there is no login status, the page will jump to the login page automatically.

The 'CSV Export' button is in a form tag. The method of the form is 'POST', and the input button is a function button for exporting the results. The preview of the data has a form of the table, which has five columns for each value type, as mentioned in the database section. The cells under the column labels are calling from 'data' table in the database, which is using 'while' loop for each row. The full coding is in the Figure 3.3.7-38.

```
if (isset($_SESSION['student_no'])) {
    $sql = mysqli_query($link, "SELECT student_no, value_s, question_id,
        answer_id, date(date) AS date FROM `data` WHERE LENGTH(`student_no`)=8
        ");

    ?>
    <br>
    <div class="container">
        <br>
        <form method="post" action="php/output.inc.php">
            <input type="submit" name="survey_output" class="action-button"
            value=" CSV Export">
        </form>
        <br>
        <table style="margin: auto;">
            <tr>
                <th width="25%">Student Number</th>
                <th width="25%">Valued Student</th>
                <th width="18%">Question ID</th>
                <th width="18%">Answer ID</th>
                <th width="24%">Submit Date</th>
            </tr>
            <?php
            while ( $row = mysqli_fetch_array($sql))
            {
            ?>
            <tr>
                <td><?php echo $row["student_no"] ?></td>
                <td><?php echo $row["value_s"] ?></td>
                <td><?php echo $row["question_id"] ?></td>
                <td><?php echo $row["answer_id"] ?></td>
                <td><?php echo $row["date"] ?></td>
            </tr>

        <?php
        }
        ?>
        </table>
        <br>
    </div>
```

Figure 3.3.7-38

The backend of output page ('output.inc.php' code is in Figure 3.3.7-39) is using the 'isset' and 'POST' function for the button 'CSV Export'. Once the student clicks the button, the backend is created an attachment, which has a CSV form, a UTF-8 character encoding and a name of 'data.csv'. Then open the attachment file that just created by 'fopen()' function, and write it. The value in the CSV file is the same as the previewed webpage and using the 'fputcsv()' function to insert the values into the attachment. At the end of the backend file, the CSV attachment just written should be closed by 'fclose()' function.

```
if (isset($_POST['survey_output'])) {
    header('Content-type: text/csv; charset=utf-8');
    header('Content-Disposition: attachment; filename=data.csv');
    $output = fopen("php://output", "w");
    fputcsv($output, array('Student Number', 'Valued Student', 'Question ID',
        'Answer ID', 'Submit Date'));
    $sql = mysqli_query($link, "SELECT student_no, value_s, question_id,
        answer_id, date(date) AS date FROM `data` WHERE LENGTH(`student_no`)=8
        ");
    while ($row = mysqli_fetch_assoc($sql)) {
        fputcsv($output, $row);
    }
    fclose($output);
}
```

Figure 3.3.7-39

4) Project management

As the comparison above in the Background section, the Balsamiq is fast and straightforward. For this project, the Balsamic is enough for the website layout.

Four languages have been used in this project: HTML, CSS, PHP and JavaScript. The Dreamweaver and Sublime both been used as coding tools. Before connecting to the local server, Dreamweaver is used for the static webpages, which could present to the client for the final design. As a result, the alternative plan could be generated for the client needs when the meeting is ongoing. After the client approves of the design, the

Sublime was used for dynamic webpages. PHP and JavaScript documents were inserted into the origin code and also the server and database.

As mentioned in the Background section, the MAMP is used for setting the local server and database. This project is agile, so the client in the essential. In the future, the client can be changed to a user, so that the limit can be released.

4. Analysis

Two main changes influence the whole project. The first change is that there is no login system at the first and second meetings with the client. The client wanted a survey-formed peer assessment for output of the scores at the beginning of the project. However, after I showed the concept designs, I realised that the login system is necessary because the data need to be distinguished with other group members. The other reason is that the client account is different from the students. The client requires the output of the data, which cannot be seen by students. If the students know the teammate who valued an unexpected score, the student may evaluate the teammate unfair. The client account is different from the students.

The other change of the project is the decision for the tool of database workbench. The MySQL Workbench is used at the beginning of the project. The reason is that I have used the software, and is familiar to process data in the University account. The University database could be used in different computers (library and home). However, I changed it to the PhpMyAdmin. The main reason is that MAMP supports phpMyAdmin, whereas not for MySQL. MySQL is complicated than phpMyAdmin, which means that the VPN of University is required, and then open the Git bash for the server setting. For phpMyAdmin, once the MAMP software is initiated, the phpMyAdmin can be opened in the default MAMP website.

5. Conclusion and Future

The primary purpose of this project is to build a website for the Scrum team, which collects and stores data into the database and can be export to a CSV file. The answer to the peer assessment is different depends on every student and question. The goal of the project is archived, but it needs to be consummated.

In the future, the project could improve in several aspects, for example, the design and structure, more functions for students or teachers, and the user test instead of the client.

The first aspect that can improve is to add media queries for different media devices. In the case that students may not be able to visit the website by computers, the interface of tablets and phones can be added into the CSS style. The breakpoints of the devices are usually maximum 640px for phone, 748px to 1024px for tablets, 1224px for desktop and 1824px for large screen. The other development of design and structure is the change the colour. According to the research from Danial, there are about 6% of the male population has red and green colour blindness. And there may no different with the colours chosen. Therefore, the colour selection can be changed for the colour blindness students or teachers.

The second aspect that can be improved is the functions. At present, there is no attempt for the peer assessments in every sprint period. Students cannot choose which sprint that they want to evaluate and see the status of every peer assessment (submit or not). Also, for teachers, they cannot manipulate the data in the website, but just preview the raw data at the site. In the future, the edit data should be provided on the website.

Furthermore, user testing should be instead of the client. Users should access the website in their devices and do the questionnaire or download the output data.

6. Reflection

For this project, my research skills are improved. I have got the help from Cathy Parker for the software that is easy to use. Also, she has taught me about the difference between several online libraries. For example, the Cardiff University library database is different from google scholar, and they do not coincide to each other.

For this technique, I have learnt the PHP language and phpMyAdmin for the database. I am skilled in PHP at present by using the 'isset', 'POST', and linking to the database. Also, I have understood more the logic and syntax of the PHP language. For the phpMyAdmin, the understanding of structures between the table is more precise than before. For example, the ER diagram has been modified several times, and the relationships between tables may conflict with each other.

However, the management of project and time need to be improved. A lot of time wasted at the beginning of the project for struggling with the concept designs and websites.

References

Prettyman, S. 2016. *Learn PHP 7: Object-Oriented Modular Programming using HTML5, CSS3, JavaScript, XML, JSON, and MySQL*. Berkeley, CA: Berkeley, CA: Apress. doi: 10.1007/978-1-4842-1730-6.

Resnick, S. 2011. *Professional Scrum with team foundation server 2010*. Bjork, A. andDela Maza, M. eds. Indianapolis, Ind.: Indianapolis, Ind. : Wiley.

Teorey, T.J. 2011. *Database modeling and design logical design*. 5th ed. Amsterdam : Amsterdam .

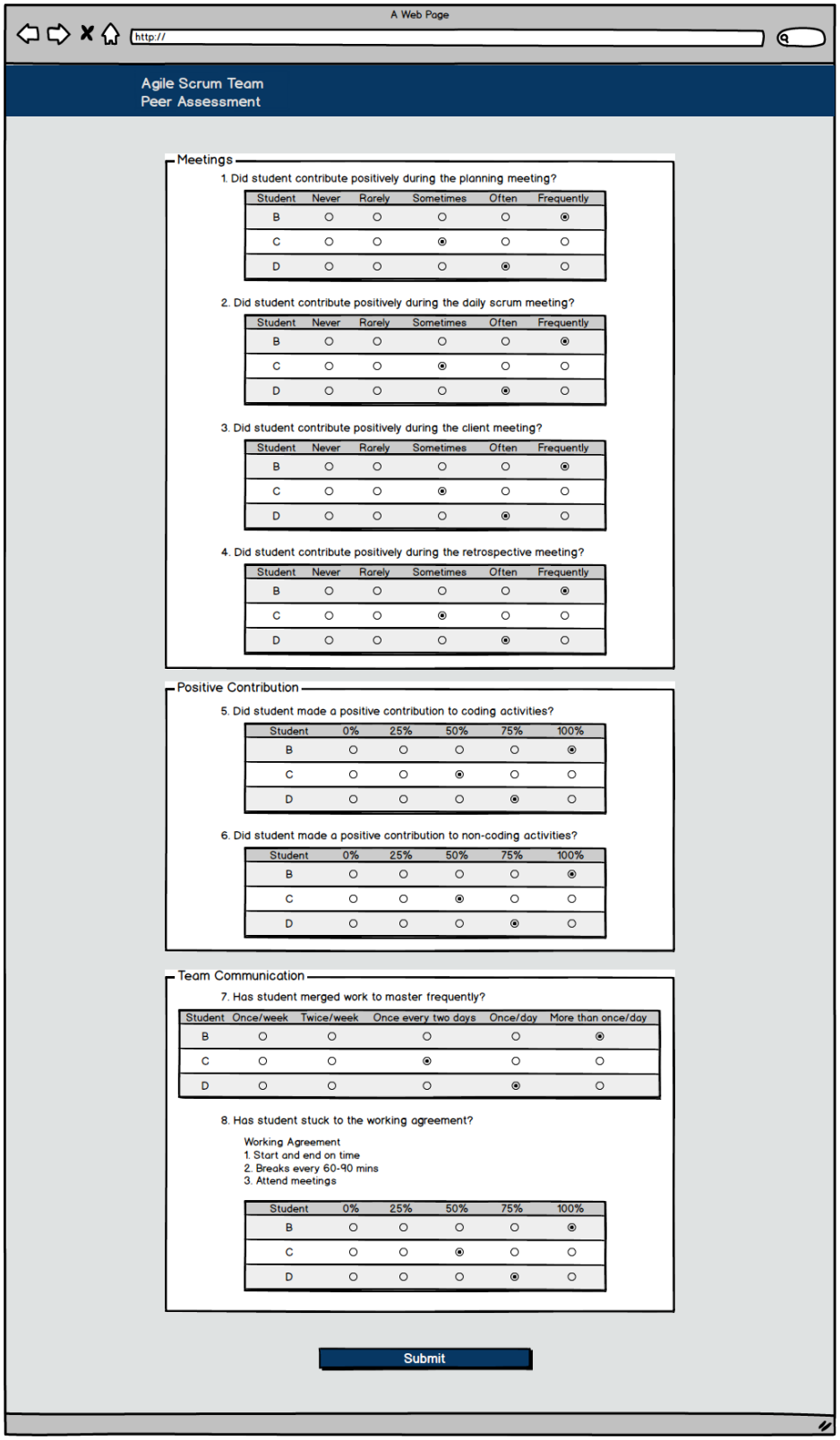
Topping, K.J. 2009. Peer Assessment. *Theory Into Practice* 48(1), pp. 20–27. doi: 10.1080/00405840802577569.

Tsai, C.-C. et al. 2014. The effect of social comparison with peers on self-evaluation. *Psychological reports* 115(2), p. 526. doi: 10.2466/07.PR0.115c20z7.

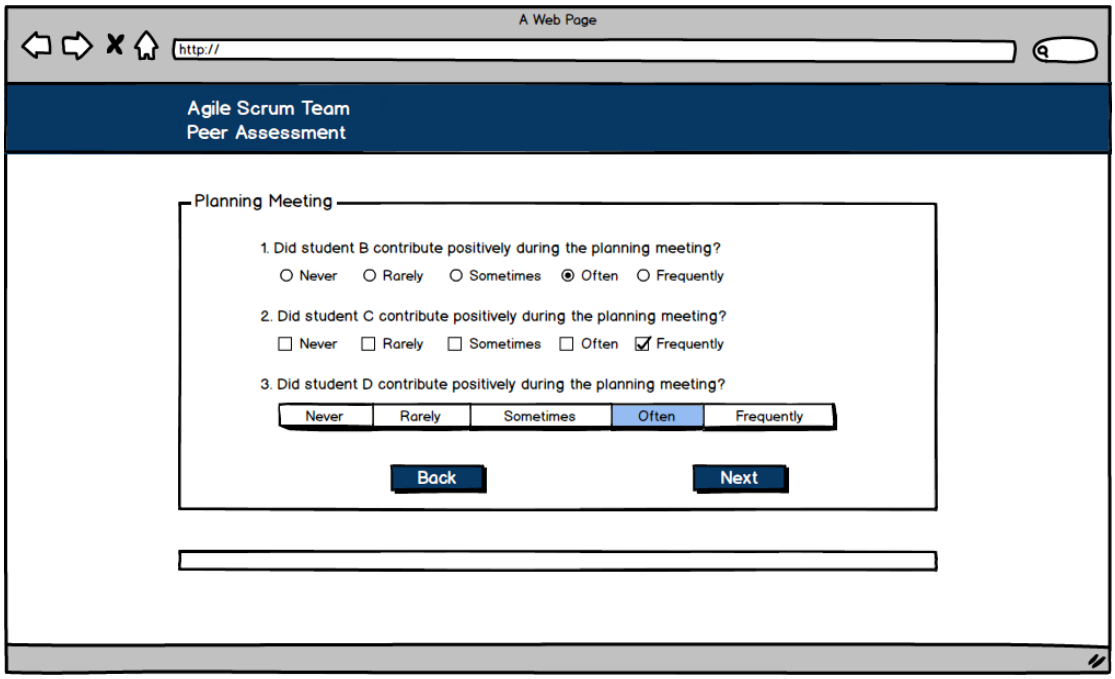
Xiao, Y. andLucking, R. 2008. The impact of two types of peer assessment on students' performance and satisfaction within a Wiki environment. *The Internet and Higher Education* 11(3–4), pp. 186–193. doi: 10.1016/j.iheduc.2008.06.005.

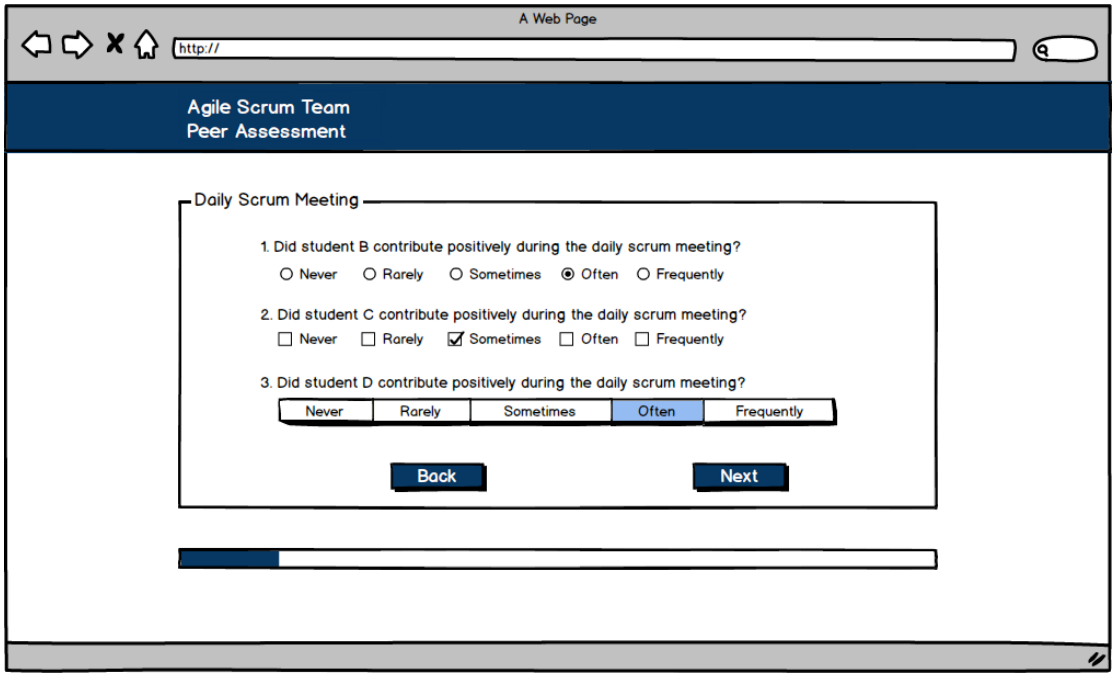
Daniel. 2009. *Deuteranopia – Red-Green Color Blindness* Available at: https://www.color-blindness.com/deuteranopia-red-green-color-blindness/ [Accessed: 17 September 2019].

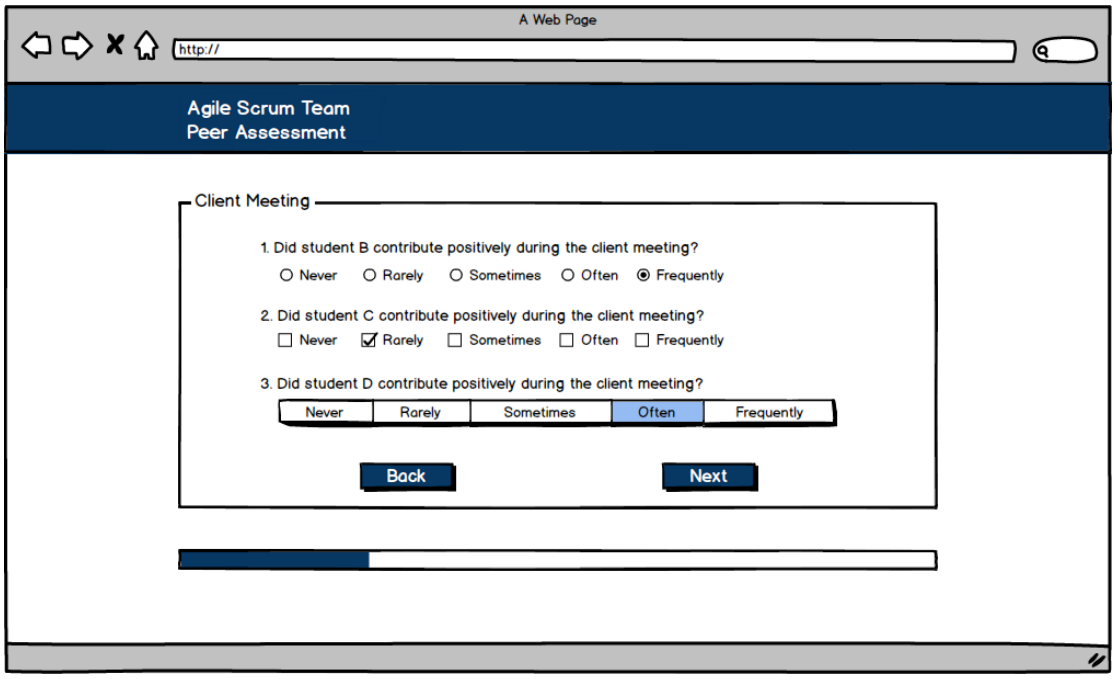
Appendices 1

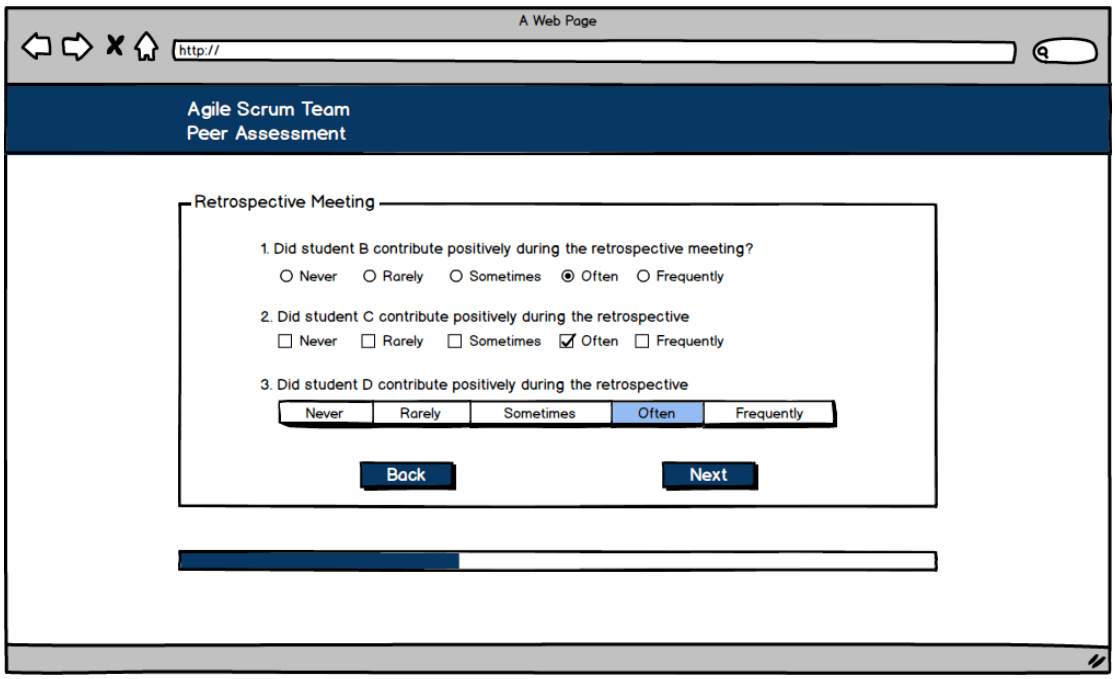


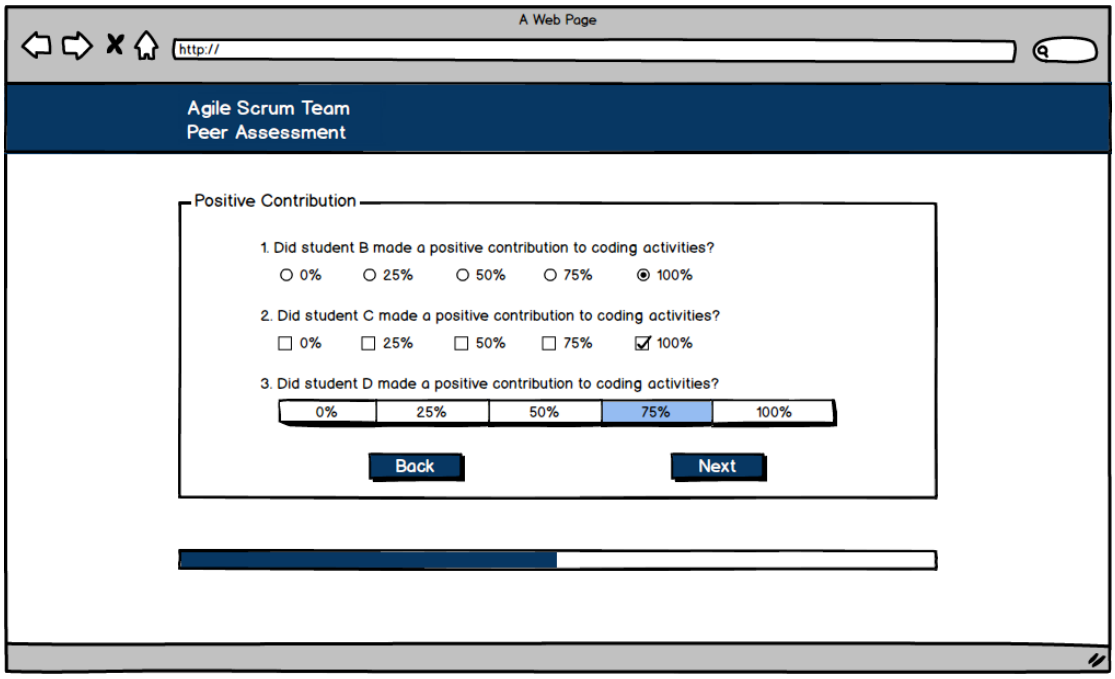
Appendices 2

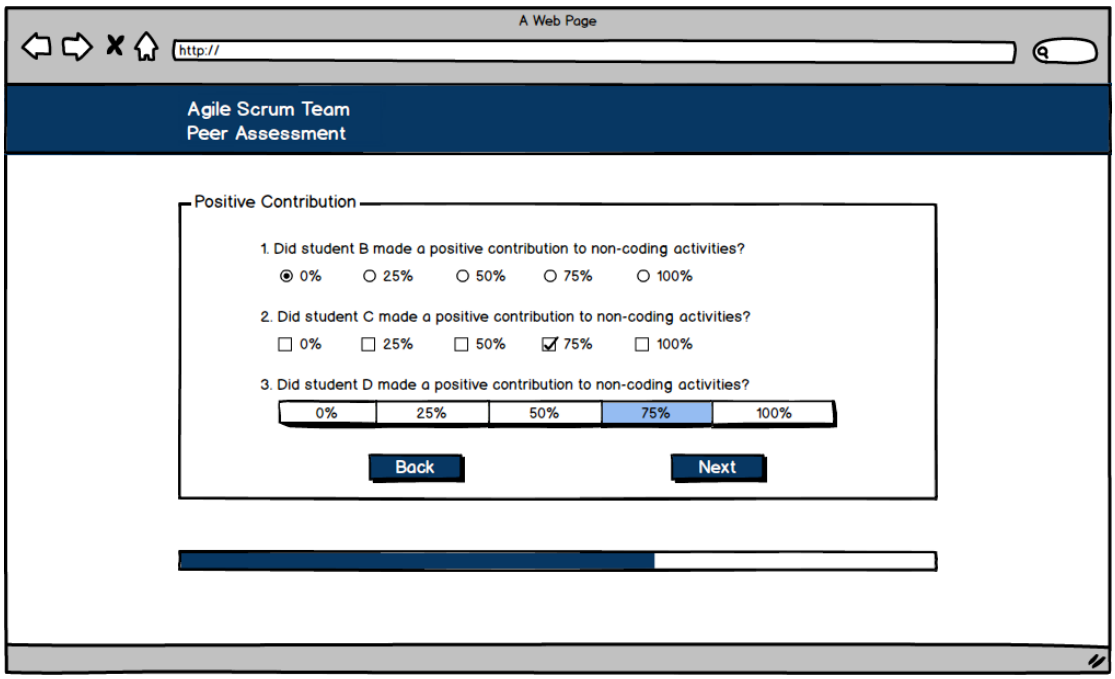


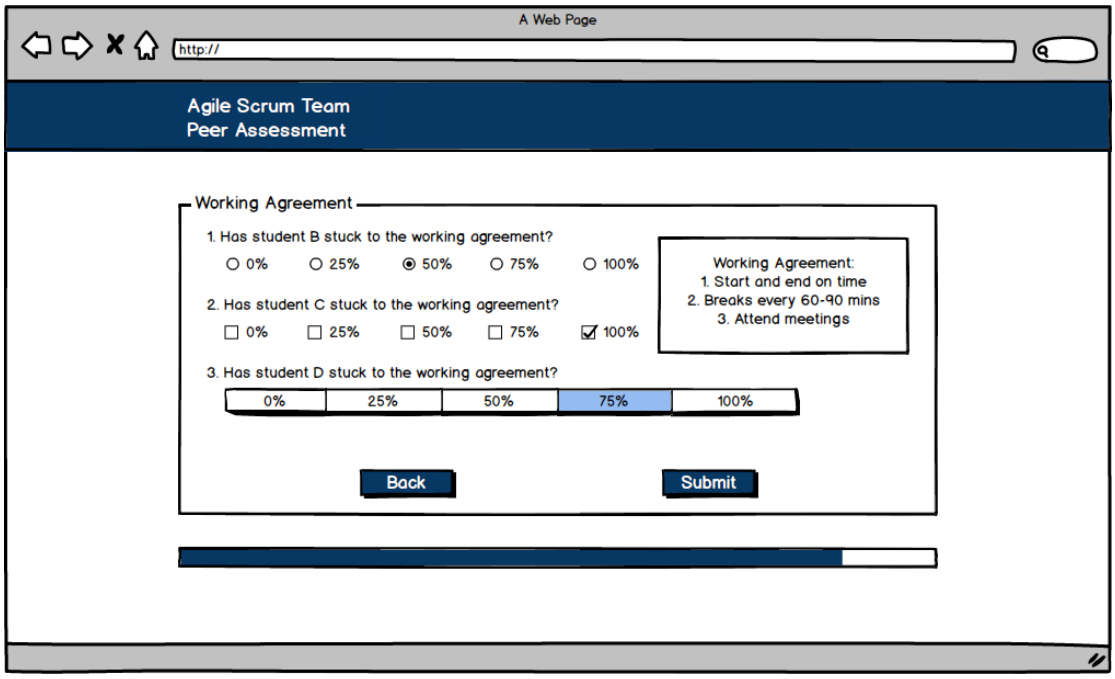



A Web Page
http://
Agile Scrum Team
Peer Assessment
Client Meeting
1. Did student B contribute positively during the client meeting?
Never Rarely Sometimes Often Frequently
2. Did student C contribute positively during the client meeting?
Never Rarely Sometimes Often Frequently
3. Did student D contribute positively during the client meeting?
Never Rarely Sometimes Often Frequently
Back
Next


A Web Page
http://
Agile Scrum Team
Peer Assessment
Retrospective Meeting
1. Did student B contribute positively during the retrospective meeting?
Never Rarely Sometimes Often Frequently
2. Did student C contribute positively during the retrospective
Never Rarely Sometimes Often Frequently
3. Did student D contribute positively during the retrospective
Never Rarely Sometimes Often Frequently
Back
Next


A Web Page
http://
Agile Scrum Team
Peer Assessment
Positive Contribution
1. Did student B made a positive contribution to coding activities?
0% 25% 50% 75% 100%
2. Did student C made a positive contribution to coding activities?
0% 25% 50% 75% 100%
3. Did student D made a positive contribution to coding activities?
0% 25% 50% 75% 100%
Back
Next


A Web Page
http://
Agile Scrum Team
Peer Assessment
Positive Contribution
1. Did student B made a positive contribution to non-coding activities?
0% 25% 50% 75% 100%
2. Did student C made a positive contribution to non-coding activities?
0% 25% 50% 75% 100%
3. Did student D made a positive contribution to non-coding activities?
0% 25% 50% 75% 100%
Back
Next



Appendices 3

There are the all the table and values inserted into the database.

answer	
answer_id	answer_body
1	Never
2	Rarely
3	Sometimes
4	Often
5	Frequently
6	1 / week
7	2 / week
8	3 / week
9	1 / day
10	> 1 / day

question	
question_id	question_body
1	Did the student contribute positively during the planning meeting?
2	Did the student contribute positively during the daily scrum meeting?
3	Did the student contribute positively during the client meeting?

4	Did the student contribute positively during the retrospective meeting?
5	Did the student make a positive contribution to coding activities?
6	Did the student make a positive contribution to non-coding activities?
7	Has the student merged work to master frequently?
8	Has the student stuck to the working agreement?

q-a		
id	question_id	answer_id
1	1	1
2	1	2
3	1	3
4	1	4
5	1	5
6	2	1
7	2	2
8	2	3
9	2	4
10	2	5
11	3	1
12	3	2
13	3	3
14	3	4
15	3	5
16	4	1
17	4	2
18	4	3
19	4	4
20	4	5
21	5	1
22	5	2
23	5	3
24	5	4
25	5	5
26	6	1
27	6	2
28	6	3
29	6	4
30	6	5
31	7	6
32	7	7

33	7	8
34	7	9
35	7	10
36	8	1
37	8	2
38	8	3
39	8	4
40	8	5

data				
student_no	value_s	question_id	answer_id	date
12345678	11345678	1	5	2019/9/17 15:18
12345678	11345678	2	5	2019/9/17 15:18
12345678	11345678	3	5	2019/9/17 15:18
12345678	11345678	4	5	2019/9/17 15:18
12345678	11345678	5	5	2019/9/17 15:18
12345678	11345678	6	5	2019/9/17 15:18
12345678	11345678	7	5	2019/9/17 15:18
12345678	11345678	8	5	2019/9/17 15:18
12345678	13345678	1	2	2019/9/17 15:18
12345678	13345678	2	2	2019/9/17 15:18
12345678	13345678	3	2	2019/9/17 15:18
12345678	13345678	4	2	2019/9/17 15:18
12345678	13345678	5	2	2019/9/17 15:18
12345678	13345678	6	2	2019/9/17 15:18
12345678	13345678	7	2	2019/9/17 15:18
12345678	13345678	8	2	2019/9/17 15:18
12345678	14345678	1	3	2019/9/17 15:18
12345678	14345678	2	3	2019/9/17 15:18
12345678	14345678	3	3	2019/9/17 15:18
12345678	14345678	4	3	2019/9/17 15:18
12345678	14345678	5	3	2019/9/17 15:18
12345678	14345678	6	3	2019/9/17 15:18
12345678	14345678	7	3	2019/9/17 15:18
12345678	14345678	8	3	2019/9/17 15:18
12345678	15345678	1	4	2019/9/17 15:18
12345678	15345678	2	4	2019/9/17 15:18
12345678	15345678	3	4	2019/9/17 15:18
12345678	15345678	4	4	2019/9/17 15:18
12345678	15345678	5	4	2019/9/17 15:18

12345678	15345678	6	4	2019/9/17 15:18
12345678	15345678	7	4	2019/9/17 15:18
12345678	15345678	8	4	2019/9/17 15:18

students				
student_no	student_fname	student_lname	group _id	password
123456	James	Osborne	0	$2y$10$K3JMbQQKhI1w7Rw1uwK9c.Drbz.LkAJ3jHqvOcSjFd/7PjnviDWBO
11111111	123	123	5	$2y$10$x8fFfr9TwexAV9QM/4L57uqmTTpQwFP5xTIyMgkLIk8Zph7FH0tYW
11345678	Bob	Jones	1	$2y$10$K3JMbQQKhI1w7Rw1uwK9c.Drbz.LkAJ3jHqvOcSjFd/7PjnviDWBO
12345678	Alice	Smith	1	$2y$10$K3JMbQQKhI1w7Rw1uwK9c.Drbz.LkAJ3jHqvOcSjFd/7PjnviDWBO
13345678	Charlie	Morgan	1	$2y$10$K3JMbQQKhI1w7Rw1uwK9c.Drbz.LkAJ3jHqvOcSjFd/7PjnviDWBO
14345678	Dianne	WIliams	1	$2y$10$K3JMbQQKhI1w7Rw1uwK9c.Drbz.LkAJ3jHqvOcSjFd/7PjnviDWBO
15345678	Brandi	Tyler	1	$2y$10$K3JMbQQKhI1w7Rw1uwK9c.Drbz.LkAJ3jHqvOcSjFd/7PjnviDWBO
16345678	Jill	Anderson	2	$2y$10$K3JMbQQKhI1w7Rw1uwK9c.Drbz.LkAJ3jHqvOcSjFd/7PjnviDWBO
17345678	Jack	Rowland	2	$2y$10$K3JMbQQKhI1w7Rw1uwK9c.Drbz.LkAJ3jHqvOcSjFd/7PjnviDWBO
18345678	Clark	Andrews	2	$2y$10$K3JMbQQKhI1w7Rw1uwK9c.Drbz.LkAJ3jHqvOcSjFd/7PjnviDWBO
19345678	Kevin	Bailey	2	$2y$10$K3JMbQQKhI1w7Rw1uwK9c.Drbz.LkAJ3jHqvOcSjFd/7PjnviDWBO
22222222	123	123	123	$2y$10$obvcn2XOvwALo0on0t7DUuliZ1FO02RJjkpt.ONVv5/CjsslnxKVC