**How to use this template:**

1. If you haven’t done already, create a google drive folder named, “TeamName”, and   
   share the folder with your team members and me [inssong@gmail.com](mailto:inssong@gmail.com)
2. Make a copy of this template into your team folder: Goto File menu -> click Make a copy
3. Move the copy to your team folder: File -> Move-To-Folder, select the shared folder.
4. Please read [the referencing instruction](https://docs.google.com/document/d/1COZjLFySsWrIy71uPOpEXcAOH4-R2hdEgS8uii1RXWg/edit?usp=sharing) to use the Google research tool for writing reports.

[Make sure to work on a copy of this file in your shared folder “TeamName” with your team]

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_[marking notes]**

**CP3046** & **CP5046** ASSESSMENT TASK 3: **Project Audit**

This assessment task has been prepared by Dr. Dmitry Konovalov for James Cook University. Updated 18 February 2015.

**© Copyright 2015**

This publication is copyright. Apart from any fair dealing for the purpose of private study, research, criticism, or review as permitted under the Copyright Act, no part may be reproduced by any process or placed in computer memory without written permission.

|  |  |
| --- | --- |
| Aligned BIT course & CP3046 subject learning outcomes | * SLO - Subject Learning Outcome: SLO-1, SLO-2, SLO-3, SLO-4, SLO-5; * BLO - BIT Course Learning Outcome: BLO-K1, BLO-K2, BLO-K3, BLO-S2, BLO-S5, BLO-A1, BLO-A3. |
| **Group or individual** | *Individual.* ***Students from the same team may be awarded different marks for this assessment item*** |
| **Weighting / Length** | *65%* |
| **Due date** | *before the last workshop* |
| **Aligned MIT course & CP5046 subject learning outcomes** | * **SLO - Subject Learning Outcome: SLO-1, SLO-2, SLO-3, SLO-5;** * **MLO - MIT Course Learning Outcome : MLO-K1, MLO-K2, MLO-S4, MLO-S5, MLO-A1, MLO-A3.** |
| **Group or individual** | *Individual.* ***Students from the same team may be awarded different marks for this assessment item*** |
| **Weighting / Length** | *50%* |
| **Due date** | *before the last workshop* |

## ASSESSMENT TASK DESCRIPTION

This task is the Audit of: (i) your group ICT solution delivered to the client, and (ii) ***individual contributions*** to the overall team effort and results. In terms of the Agile Software development, this is the ***iteration-2***. It delivers the ***alpha release*** to the client, and makes planning for ***iteration-3*** (***beta release***), which is due middle of CP3047/CP5047.

## [Total awarded \_\_\_\_\_ out of 100 marks] ASSESSMENT TASK CRITERIA SHEET

***NOTE! Students from the same team may be awarded different marks for this assessment item. To arrive at the individual student marks, your lecturer may consider all or some of the following contributing factors: your team overall marks; team feedback forms; personal observations of the student project contribution; student workshop attendance and subject participation; student competency during workshops, this audit and the project presentation.***

The following is the list of items which are required to be completed in this assignment. Maximum possible marks are given in brackets at the beginning of each item. Where relevant, the SLOs and BLO/MLOs from the preceding table are specified.

**[\_\_\_\_/Prerequisite for marking]** Assignment is completed using electronic copy of ***this*** document and submitted to LearnJCU electronically.One submission per team.

**[\_\_\_\_\_/20 marks, BLO-S5] *Individual* Team Feedback form is submitted to LearnJCU electronically.** One submission per student. Assignment is done in a group with 2-4 students.

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name** | **Skills** | **Project roles** | **Contribution to the Product** |
| Lim Hui Ching | * HTML/CSS * Bootstrap * JavaScript * Trello | * HTML/CSS Developer * Designer | * Home Page * Student Page |
| Park Ji Hyun | * HTML/CSS * Bootstrap * JavaScript * Trello | * HTML/CSS Developer * Designer | * Teacher Page |
| Nguyen Kim Anh | * PHP * MySQL * Trello | * PHP Developer * Database Architect | * Connection to the Database in PHP |
| Paul Wenzel | * PHP * HTML/CSS * Bootstrap * JavaScript * SQL * Trello * Visio * UMLet | * Project Manager * PHP Developer | * PHP Model-View-Controller Framework * Backend Logic * Database Architecture |

**[\_\_\_\_\_/40 marks, SLO-1, SLO-2, BLO-S3] Report and demonstrate the *ACTUALLY* delivered alpha-release (see your user stories in iteration-1).** Any deviations from the alpha-release-iteration-1 must be documented and briefly explained. Screen-shots (or illustrations) of running alpha-release with comments or explanations.Write here: Minimum **TWO** pages, maximum **TEN** pages.

**Vision Statement**

“Lecture Poll Web App” is a web-based application for teachers who are concerned about if their students are able to follow their lecture and students who want to give their teacher anonymous feedback.

Unlike available ICT solutions “Lecture Poll Web App” is an easy to use web application that is specially optimized for mobile devices with small screens. Since it is designed for only one purpose, it has a really intuitive user interface and does not need any customization. This is why users who use “Lecture Poll Web App” will be more efficient and faster compared to users who are using other available solutions for the same purpose.

The application enables students to vote anonymously either “I got it!” or “I am lost!” in order to tell the teacher if they are able to follow the lecture. Additionally, the teacher sees through the application how the students are doing in real-time.

“Lecture Poll Web App” brings benefits for both, student and teacher. The teacher can react to the students in real-time by optimizing the speed of the lecture based on the students voting. Furthermore, the teacher can analyze the result of a lecture poll after the end of the lecture to see which parts of the lecture were easy and which parts were more difficult for the students to understand. Based on this information the teacher can adjust the lecture for future classes. For students, the application makes it easy to give honest feedback about the lecture since it is anonymous. As a result, students will receive a better learning experience.

**Lecture Poll Web Application Overview**

Our application “Lecture Poll Web App” consist of three main pages: home page, student page and teacher page.

On the home page there is one section called “Login” where both students and teachers can login using a student or teacher poll code. Students will be redirected to the student page and teachers will be directed to teacher page when they log-in. There is another section called “Create Poll”, which can be used by a teacher to create a new poll with the desired name.

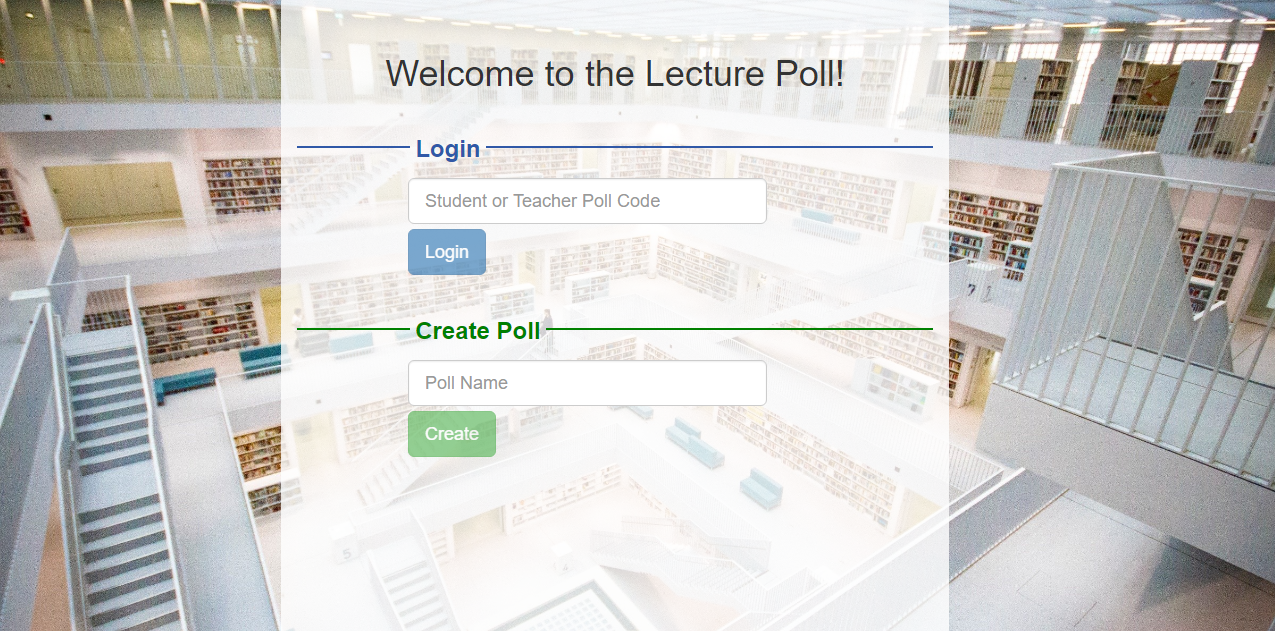


Figure 1: Home page of the Lecture Poll App

Once students logged in using a student poll code, they will be redirected to the Student page where they can express their status during the lecture. At the top of the student page, there is a smiley that shows the current rating of the student. It is either a happy smiley face for expressing “I got it!” or sad smiley face meaning “I am lost!”. At the bottom of the page are two buttons, “I got it!” and “I am lost” which the student can click to express their status.

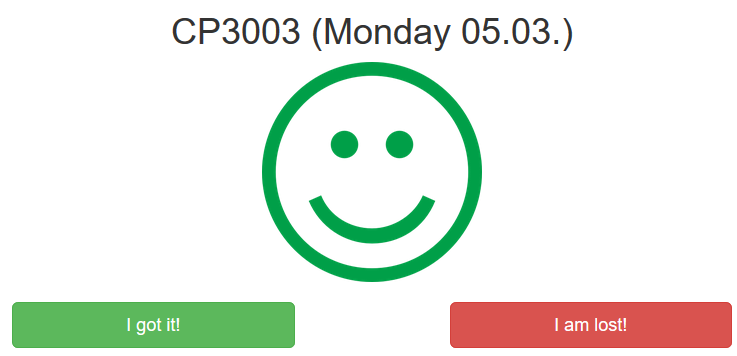


Figure 2: Student page

When teachers log in using a teacher poll code or create a new poll, they get redirected to the teacher page where they can see the mood of their students. At the top of page, it shows the name of the poll assigned by the teacher during the creation of the poll. Below the poll name, a smiley face represents the overall status of the students in the class. If more than 80% of students can follow the lecture a green smiley face will be shown. A yellow neutral face will appear when more that 50% but less than 80% of students can follow the lecture. If 50% ore less of the students are able to follow the lecture red sad face will be shown. Right under the smiley a graph gives a more detailed view about how many students are able to follow the lecture or are lost. Right below the graph, there is poll code button to show both student and teacher poll code for this poll. The button “Download Data” triggers a download of the poll data. The data gets represented in a .csv-File and not only includes the current status but all ratings given by the students during the particular poll. The last button closes the poll, so no student will be able to give up a rating anymore.

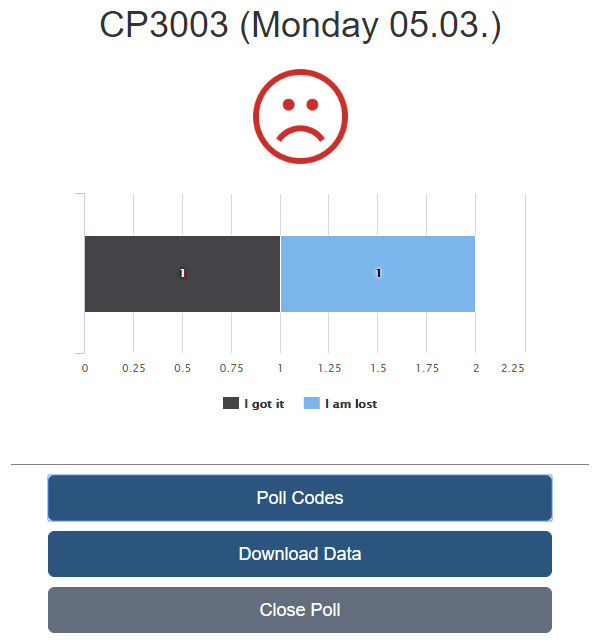


Figure 3: Teacher page

**Alpha Release Development**

The alpha release of “Lecture Poll Web Application” contains 2 milestones with 5 user stories allocated to each milestone.

The user stories from the first milestone focused on receiving a running product. This is why during milestone one the most important features of the application got implemented. In order to implement all of the basic features in such a short time, a really simple user interface was chosen for the first milestone.

The second milestone was all about optimizing the user interface to provide users with a unique user experience. Moreover, additional features like downloading the poll data as a .csv-file and the option for closing a poll were added.

**Milestone 1**

In milestone one, our team delivered the fundamental functionalities of the application. Our client, Dr. Nicolas Greliche was extremely satisfied with our delivery in iteration one, with acceptance level of excellence in most of the criterions. He was impressed that we managed it to build a running product after the first iteration. The actual functionality implemented in milestone one is going to be documented as follows.

*User Story 1: Teacher can create a poll.*

On the home page the teacher is able to create a poll. The home page consists of two sections. One section for the teacher to create a poll and another section for teachers and students to login to a poll.

If the teacher wants to create a poll, he enters a name for the poll first. The name for the poll should be meaningful and an identifier for the teacher and the students to know for which lecture the poll got created. After the teacher has entered a poll name, he clicks the button “Start New Poll…” to create a new poll. Afterwards, the teacher will be automatically redirected to the teacher page of the newly created poll.

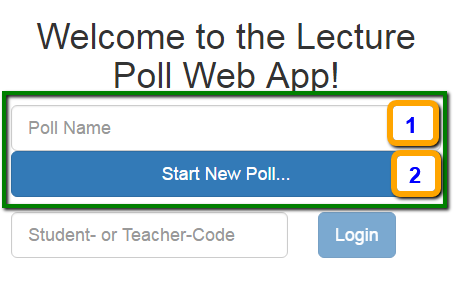
****

Figure 4: Create a poll on the home page

*User story 2: Teacher can look up the teacher and student poll code.*

Each poll gets automatically assigned a students and teacher poll code as soon as the poll gets created. Those codes can be used by teachers and students to login to the poll. On the teacher page, the teacher can lookup both poll codes by clicking on the button “Poll Codes”. This opens up a pop-up dialog showing the poll codes.

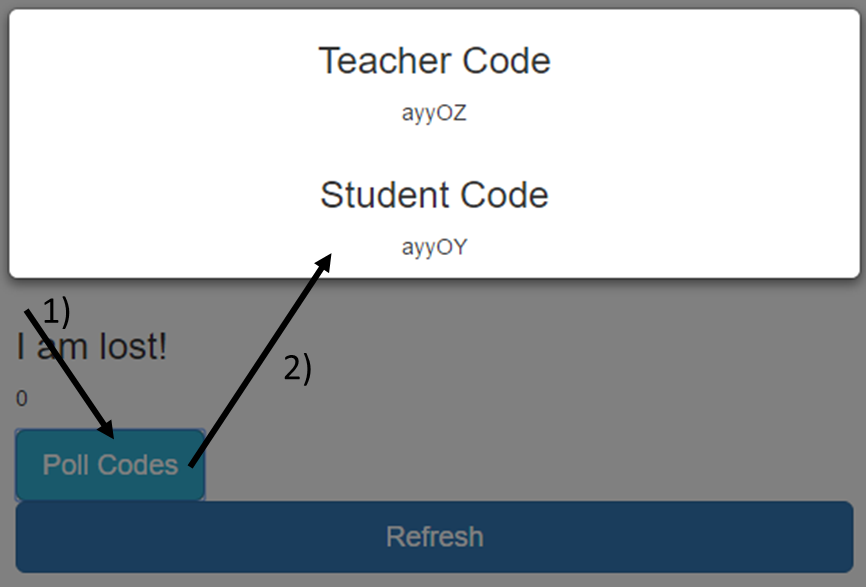


Figure 5: Teacher looks up teacher and student poll codes on the teacher page

*User story 3: Teacher/Student can login to a poll via teacher/student poll code.*

To login to a poll both teacher and student first go to the home page of Lecture Poll Web App.

Teacher and students can then enter their poll code into the “Student- or Teacher-Code” input field. Afterwards they click on the “Login” button to login to the poll. If a teacher poll code got entered, the user gets redirected to the teacher page. If a student poll code got entered, the user gets redirected to the student page.

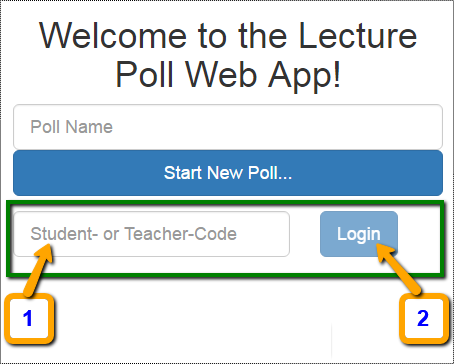


Figure 6: Teacher/Student login in Iteration 1

*User story 4: Student is able to rate the lecture*

After the students logged in using a student poll code they will be on the student page. At the bottom of the student page the students can see the details about the poll that they entered.

On the student page the students got two options. They can either vote “I got it!” by clicking on the green button or they can vote “I am lost!” by clicking on the red button. Depending on which button they click, they either express that they are able to follow the lecture or not.

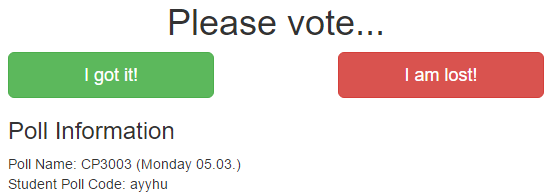


Figure 7: Student page

*User story 5: Teach can see the status of his students in real time.*

After teachers logged in using a teacher poll code they will be on the teacher page. On the teacher page, the teachers are able to see the current status of their students. The status gets expressed by two numbers. The first number represents the number of students who got it. The second number represents the number of students who are lost. The teachers have to click the “Refresh” button in order to refresh the page and view the most current status of their students.

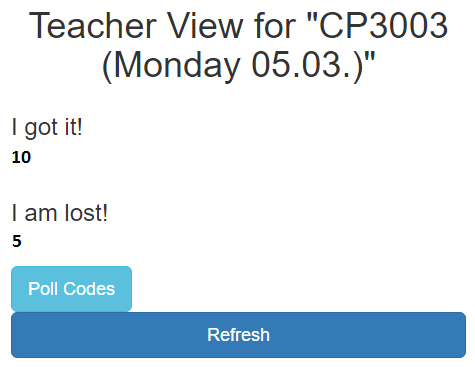


Figure 8: Teacher page showing the current status of the students

**Milestone 2**

In second iteration, we improved the user interface to optimize the user experience. In addition, we added further features. Our client praised how we adopted to his feedback of iteration one and loves the new features that we added during iteration two. All in all, he is really satisfied with the alpha release version and wants to use Lecture Poll Web App in his lectures as soon as possible.

*Improvements made based on the client’s feedback*

The home page has been completely restructured and redesigned during iteration two. We did this in order to implement the client’s feedback and to improve the user experience. The key improvement is the clear separation between the login and create poll forms on the home page.

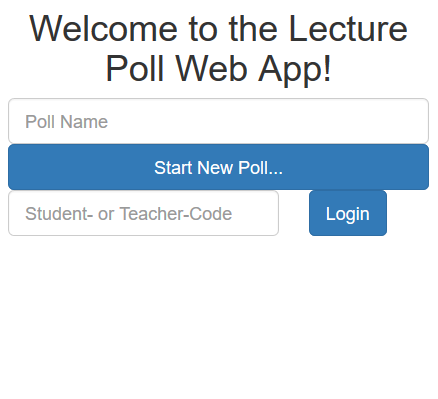


Figure 9: Home page after iteration 1

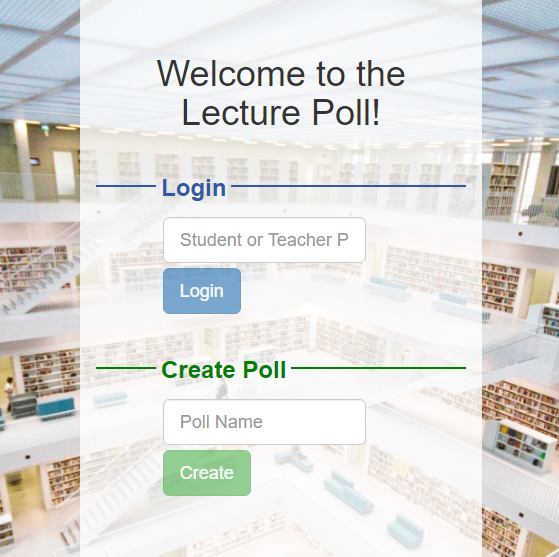


Figure 10: Redesigned home page after iteration 2

Furthermore, the student page has been improved by making the current status of the student visible to the student by adding a smiley face that represents his current status.

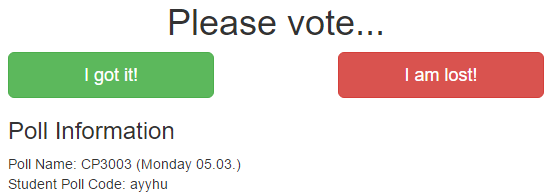


Figure 11: Student page after iteration 1



Figure 12: Redesigned student page - green smiley face shown after student clicked on "I got it!"



Figure 13: Redesigned student page - green smiley face shown after student clicked on "I am lost!”

*User story 6: Teachers and students get visual feedback if they enter a wrong poll code.*

On the home page a visual feedback has been added to show teachers and students if they entered a valid poll code. If an invalid poll code gets entered, the text field gets highlighted in red. If a valid poll code gets entered the text field gets highlighted in green.

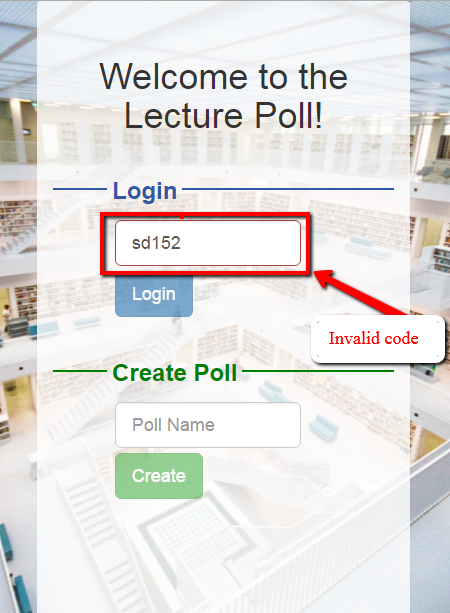


Figure 14: Warning with red text field when entering invalid poll code

*User story 7: Teacher is able to end a poll.*

After the lecture is over, the teacher can end the poll in order to prevent students to keep on rating for the finished lecture. To end a poll, the teacher clicks on the “Close Poll” button at the bottom of the teacher page.

After the teacher ended a poll, students cannot rate or login to the poll anymore.

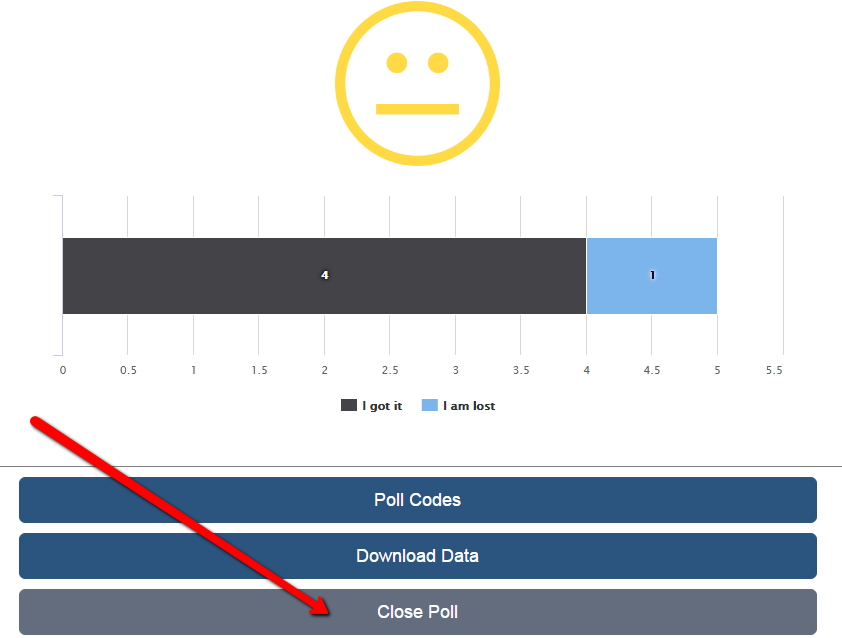


Figure 15: Teacher can close a poll

*User story 8: Teacher can see the mood of his students in a faster and easier way.*

The teacher page is supposed to give the teacher a quick overview about his students’ current status. Therefore, we added a view that summarizes the status of the students in a single smiley. If more than 80% of students can follow the lecture a green smiley face will be shown. A yellow neutral face will appear when more that 50% but less than 80% of students can follow the lecture. If 50% or less of the students are able to follow the lecture a red sad face will be shown.

In addition to that, we added a bar chart which shows in detail how many students are able to follow the lecture and how many are lost. The bar chart replaces the numeric representation developed in iteration one.

The smiley view and the bar chart make it really easy and fast for the teachers to lookup the current status of his students.

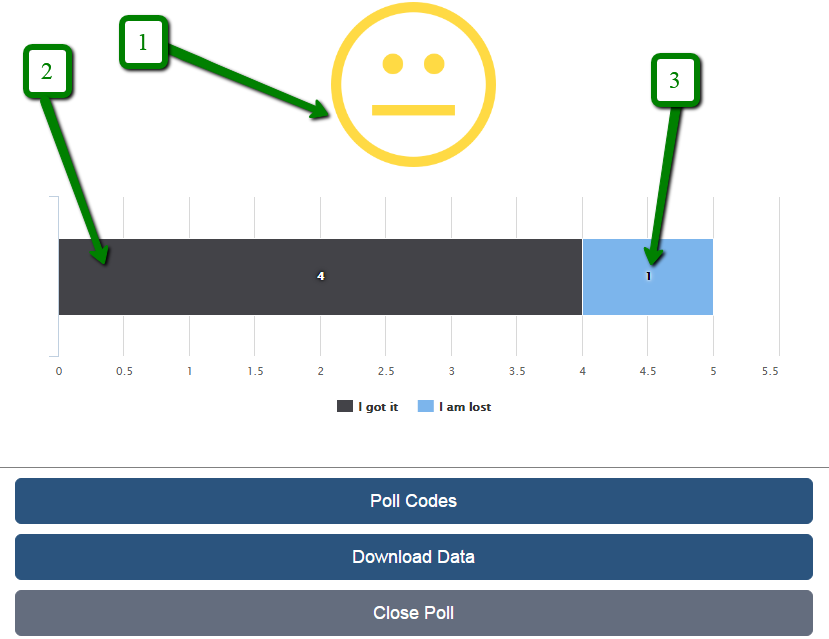


Figure 16: Teacher watching students mood during the lecture time

*User story 9: The teacher page refreshes automatically.*

In order to improve the user experience for the teacher even further, the teacher page does not have to be refreshed manually. Instead the page refreshes automatically, so that the most current status of the student gets presented on the teacher page at any point in time. This means that the teacher can just leave the teacher page opened on his device and can lookup the most current data at any time without having to refresh the web page manually by clicking any button.

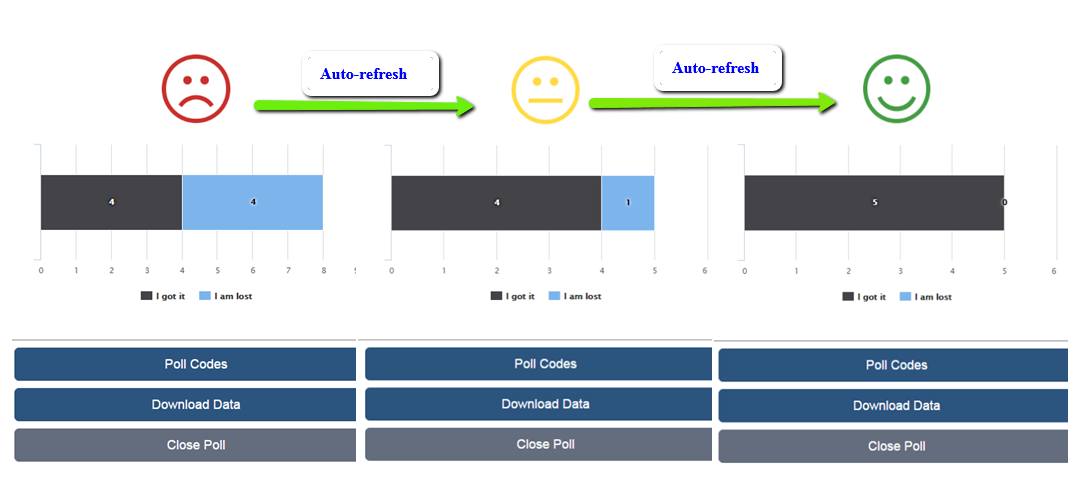


Figure 17: Automatically update Teacher's view

*User story 10: Teacher can export the poll result as a “.csv”-file.*

The teacher is able to download the poll data as a .csv-file by clicking on the “Download Data” button at the bottom of the teacher page. The file lists all ratings given by the students for the specific poll. The teacher can use this data to generate a graph or perform analysis in various ways. The teacher could for example analyse at what time the students were not able to follow the lecture anymore.

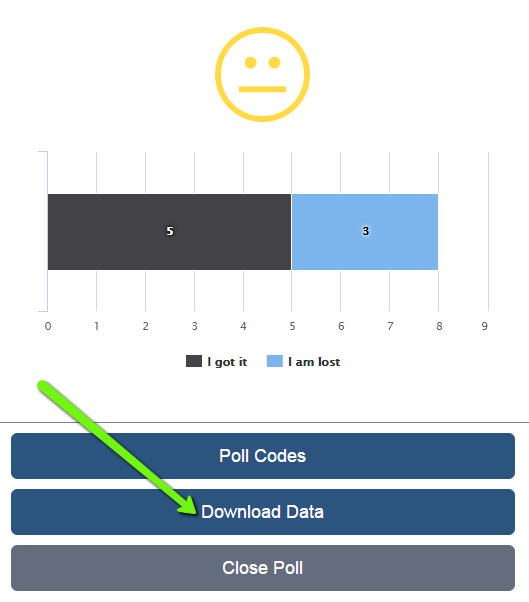


Figure 18: "Download Data" button to download the poll data as a .csv-file

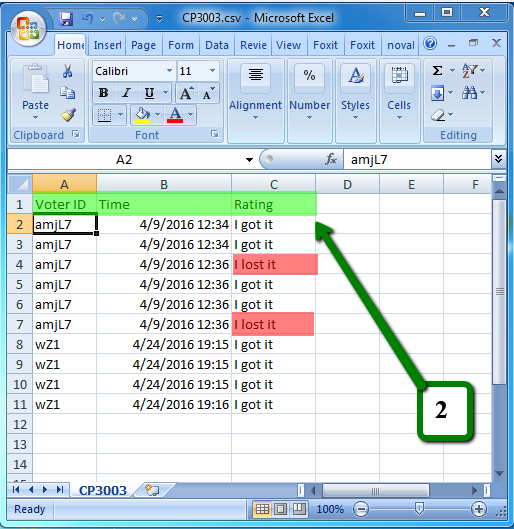


Figure 19: Exported .csv-file opened in Excel

**[\_\_\_\_\_/20 marks, BLO-K1, BLO-K3, BLO-S5, MLO-K1, MLO-S5] Client signed acceptance of the alpha-release, and the proposed beta- and final-releases**. Any changes from iteration-1 are approved by the client. Write here: Minimum **TWO** pages, maximum **TEN** pages.

* [\_\_\_\_/5 marks] Provide the burn-down and velocity charts/values for iteration-2, and how they are used to plan beta- and final-releases.

**Alpha, Beta and Final Release Scope**

**Client Scope Agreement**

**“Lecture Poll Web App”**

**Project Description:**

“Lecture Poll Web App” is a web-based application that assists teachers to know whether students understand their lecture or if they got lost. The application can be accessed from any device that provides a web browser and an internet connection. Furthermore, the application is optimized for mobile devices with small screens.

The application enables students to vote anonymously either “I got it!” or “I am lost!” in order to tell the teacher if they are able to follow the lecture. Additionally, the teacher sees through the application how the students are doing in real-time.

The application brings benefits for both, student and teacher. The teacher can react to the students in real-time by optimizing the speed of the lecture based on the students voting. Furthermore, the teacher can analyze the result of a lecture after the end of the lecture to see which parts of the lecture were easy and which parts were more difficult to understand for the students. Based on this information the teacher can adjust the lecture for future classes. For students the application makes it easy to give honest feedback about the lecture since it is anonymous. As a result, students will receive a better learning experience.

**Scope for the Alpha Release ICT 1:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Story Title** | **Description** | **Requirements** | **Priority** | **PED**  **(person equivalent days)** |
| **Milestone 1** | **21.03.2016 – 01.04.2016** |  | **Total PEDs** | **43** |
| **Teacher can create a poll.** | Teacher can create a new poll and can give it a name. | “Create Poll” button | H | 3 |
| “Poll Name” text field | H | 2 |
| **Teacher/Student can login to the poll via teacher/student poll code.** | The Teacher/Student can login (on the homepage) to the poll via their poll code. After the login they get redirected to the teacher/student page of the poll. | “Login” text field | H | 1 |
| “Login” button | H | 5 |
| Processing icon | L | 2 |
| Student poll page | H | 3 |
| Teacher poll page | H | 3 |
| **Teacher can look up the teacher and student poll code.** | Teacher can look up the teacher-/student-poll codes. | “Poll Codes” button | H | 3 |
| Popup Dialog (showing teacher and student poll codes) | H | 5 |
| **Student can rank the lecture.** | * Students can choose between two simple status: “I got it!” and “I’m lost!“. * Students are able to click on these buttons at any time during the lecture. * Students can see the name of the poll to clarify if they are filling out the right poll. | “I got it!” button | H | 2 |
| “I’m lost!” button | H | 2 |
| “Current-Status” view | H | 2 |
| “Poll Information” view | H | 4 |
| **Teacher can see the mood of his students in real time.** | * Two numbers are representing the mood of the students. * The first number counts the number of students with the status “I got it!”. * The second number counts the number of students with the status “I am lost!”. | Number of students with the status “I got it!” | H | 3 |
| Number of students with the status “I am lost!” | H | 3 |
| **Milestone 2** | **11.04.2016 – 30.04.2016** |  | **Total PEDs** | **42** |
| **Teachers and students get visual feedback if they entered a wrong poll code.** | If a teacher or a student enters a wrong poll code the poll code text field turns red. | “Login” text field with visual feedback | M | 3 |
| **Teacher is able to end a poll.** | * The teacher can end a poll, so the students cannot rate anymore. * Students get informed that they can not rate anymore. | “End Poll” button | M | 4 |
| “Poll Ended” view | M | 3 |
| **Teacher can see the mood of his students in real time in a faster way.** | * Teacher can see the overall mood of his students in one view. (i. e. happy face = all good (>85% got it), neutral face = moderate (> 50% got it), sad face = warning (<= 50 % got it)) * Teacher can see in a “loading bar” how many students are able to follow the lecture and how many are lost. | Overall status view | M | 4 |
| Bar chart | M | 4 |
| Bar chart description | M | 3 |
| **Teacher does not have to refresh his view in order to view the current mood of his students.** | The view of the teacher refreshes automatically as soon as a student rates his lecture. | Automated refresh | M | 4 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Teacher can export the poll result as a “.csv”-file.** | Teacher can export the poll data as a “.csv” file and either download it or send it via mail.  The exported “.csv” file can be used to create a graph.  Structure of the data: | “.csv” file | M | 5 |
| “Download” button | M | 5 |
| “Send E-Mail” button | L | 5 |
| “E-Mail” text field | L | 2 |
| **Alpha Release ICT 1** |  |  | **Total PEDs** | **85** |

**Scope for the Beta Release ICT 2:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Story Title** | **Description** | **Requirements** | **Priority** | **PED**  **(person equivalent days)** |
| **Teacher can review the history of the students’ mood during the poll.** | Teacher can review the history of the students’ status right in the teacher poll page. The history gets represented by a graph with two axis: the horizontal axis represents the time and the vertical axis represents the number of students who either got lost or can follow the lecture.  The history gets divided into 10 minutes periods. The teacher can open the bar chart for each period to get more details about the student’s mood during a specific period. | Student status graph | H | 5 |
| Graph divided into 10 minute periods | H | 3 |
| Individual bar chart for every period | H | 5 |
| Switch between history view and current view | M | 4 |
| **Teacher can record his lecture.** | Teacher can record his lecture and use this recording to see what he was talking about when his students voted “I got lost!”.  The Teacher can also download the recording as an audio file. | “Record” button | H | 5 |
| Audio timeline | H | 5 |
| “Play/Pause” button | H | 4 |
| “Download Audio File” button | M | 5 |
| Playback position pointer | L | 3 |
| **Beta Release ICT 2** |  |  | **Total PEDs** | **39** |

**Scope for the Final Release ICT 2:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Story Title** | **Description** | **Requirements** | **Priority** | **PED**  **(person equivalent days)** |
| **Teacher and Students can read a manual to learn how to use the application.** | Manual is provided for both teacher and student on the homepage. | Teacher manual | H | 5 |
| Student manual | H | 5 |
| **Teacher can add custom questions to the poll.** | Teacher can add custom questions to a poll.  This also includes that students are able to answer more than one question and the teacher can see results in his teacher page too more than one question. | “Add Question” button | H | 3 |
| “Customize Question” dialog | H | 5 |
| Multiple results on the teacher poll page | H | 3 |
| Multiple questions on the student poll page | H | 3 |
| **Students can use a native iOS App.** | Students can use a native iOS app on their iPhone or iPad to rate lectures. | Login screen | L | 5 |
| Student poll screen | L | 4 |
| “I am lost!” button | L | 3 |
| “I got it!” button | L | 3 |
| **Final Release ICT 2** |  |  | **Total PEDs** | **39** |

**Assumptions:**

1. This web application is believed to be a tool for students to express anonymously if they are able to follow the lecture or not.
2. Our client, Dr. Nicolas Greliche will offer us help by testing the web-application during a real lecture and gives us feedback.
3. In order to develop this project successfully each member will contribute to the project.
4. Each team member will complete their tasks as promised.
5. The MIS office of JCU Singapore will provide us with a PHP webserver and MySQL database.
6. At least one team member as strong PHP, HTML/CSS, JavaScript and SQL development experience.

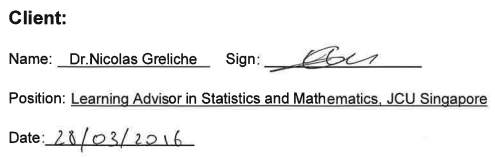
**Constraints:**

1. Our group has a limited time of ten weeks to implement requirements and features for the implementation of the web application.
2. The team only consists of four people, some with limited experiences in the development of web applications.
3. All group members have different schedules, so the team might not always be able to attend to meetings as a whole.

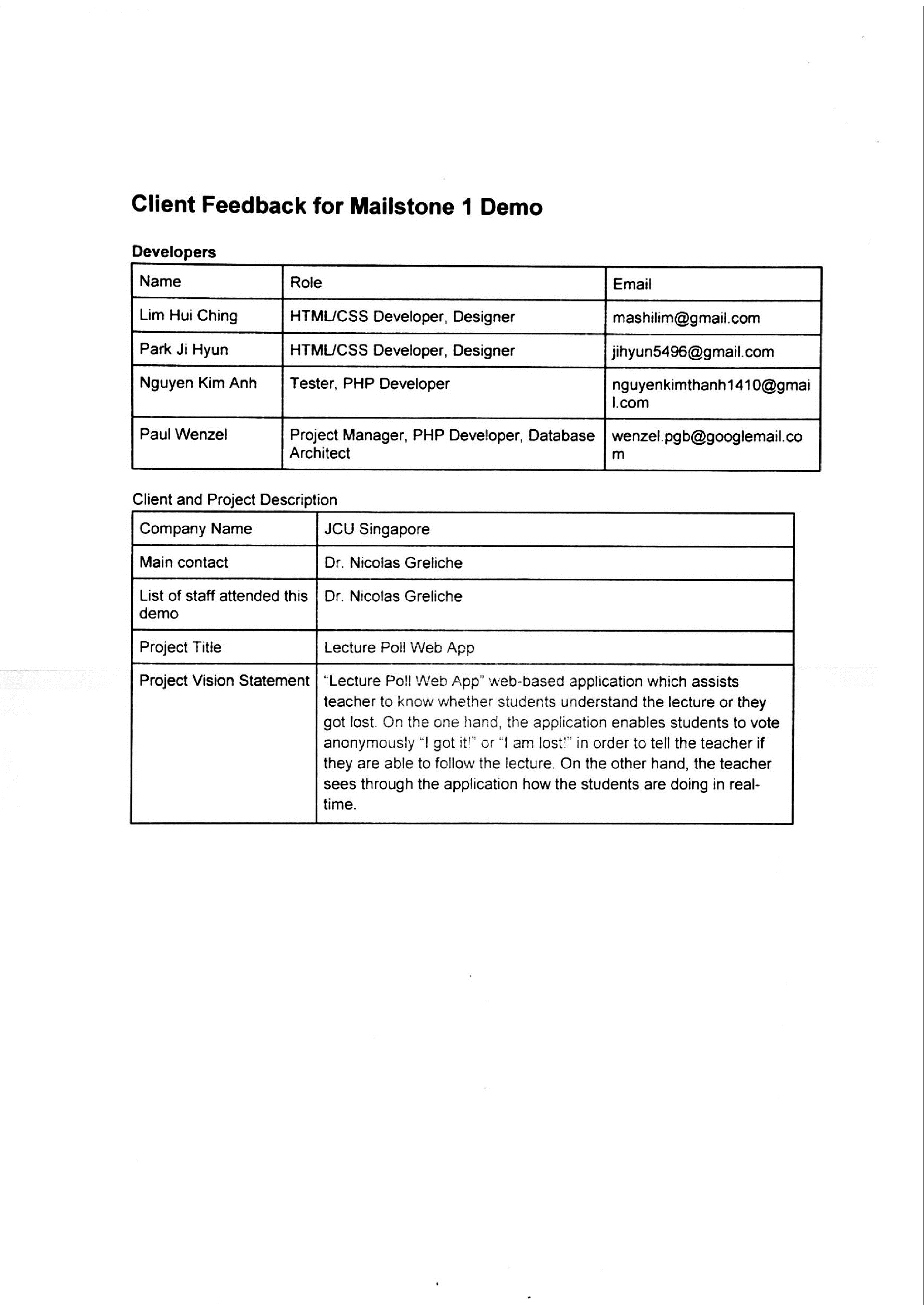
**Risks**

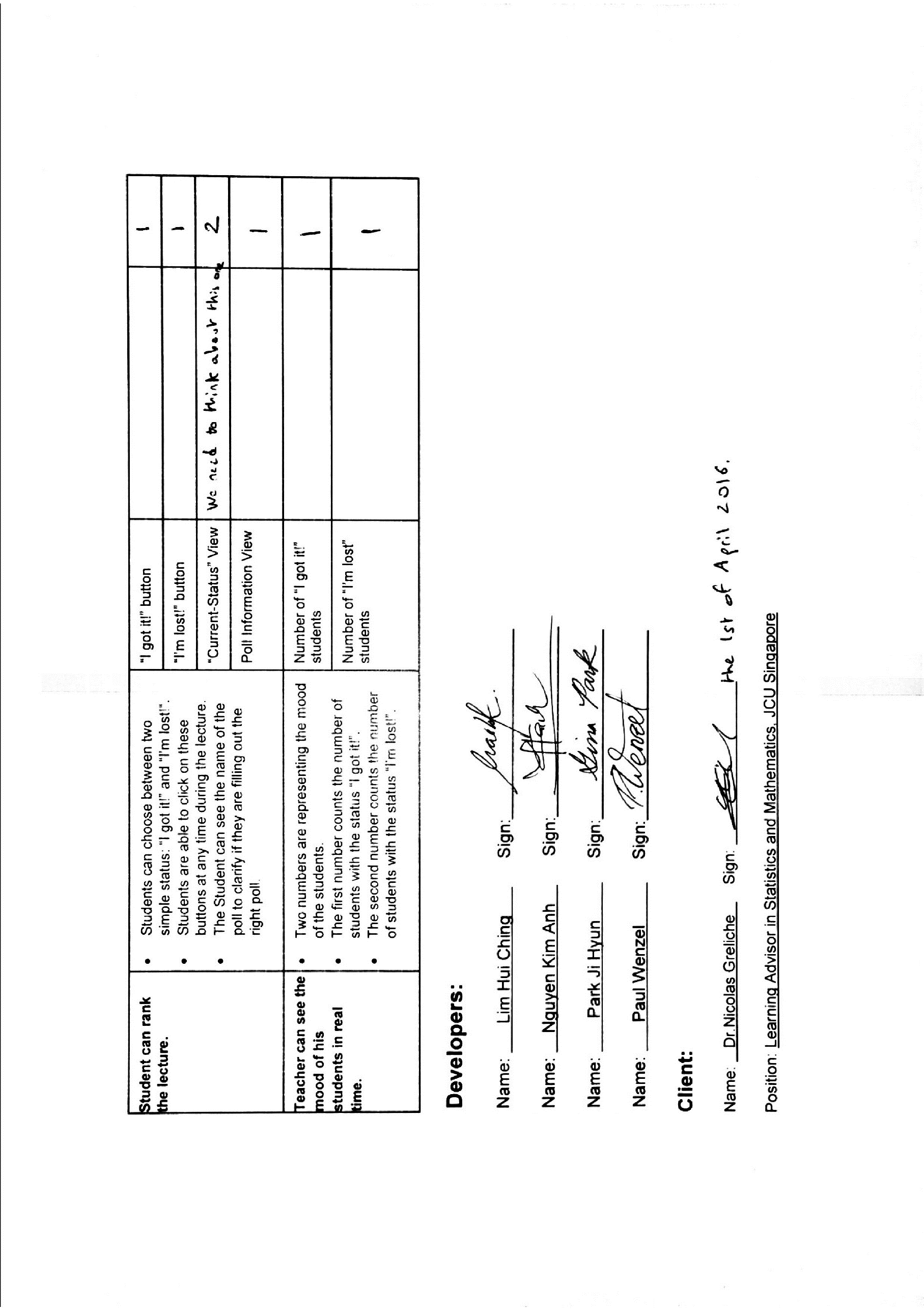
1. One or more of the assumptions listed above may not occur.
2. Not all of the predefined user stories might get implemented due to a lack of time.



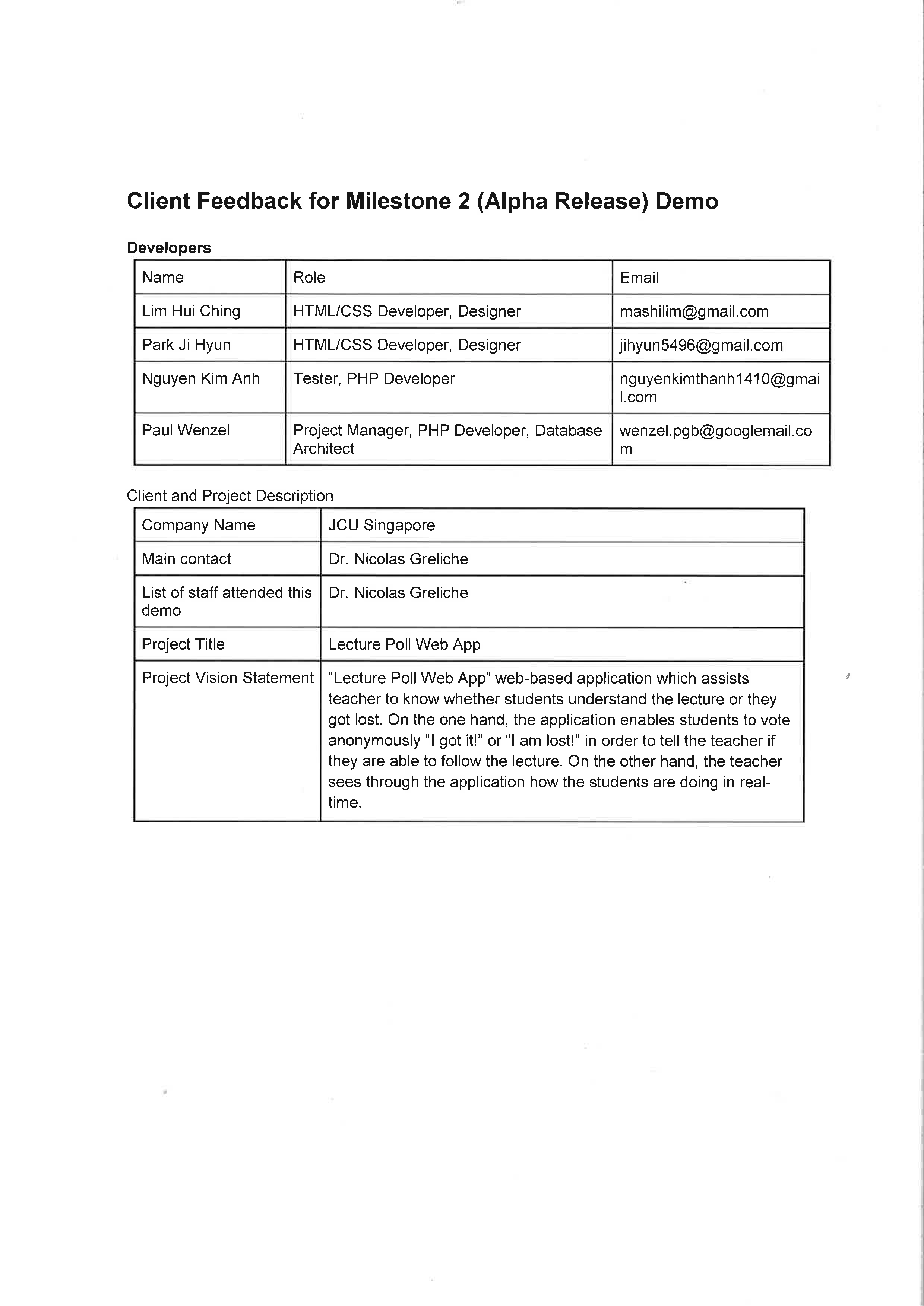


**Client Feedback Iteration 1**

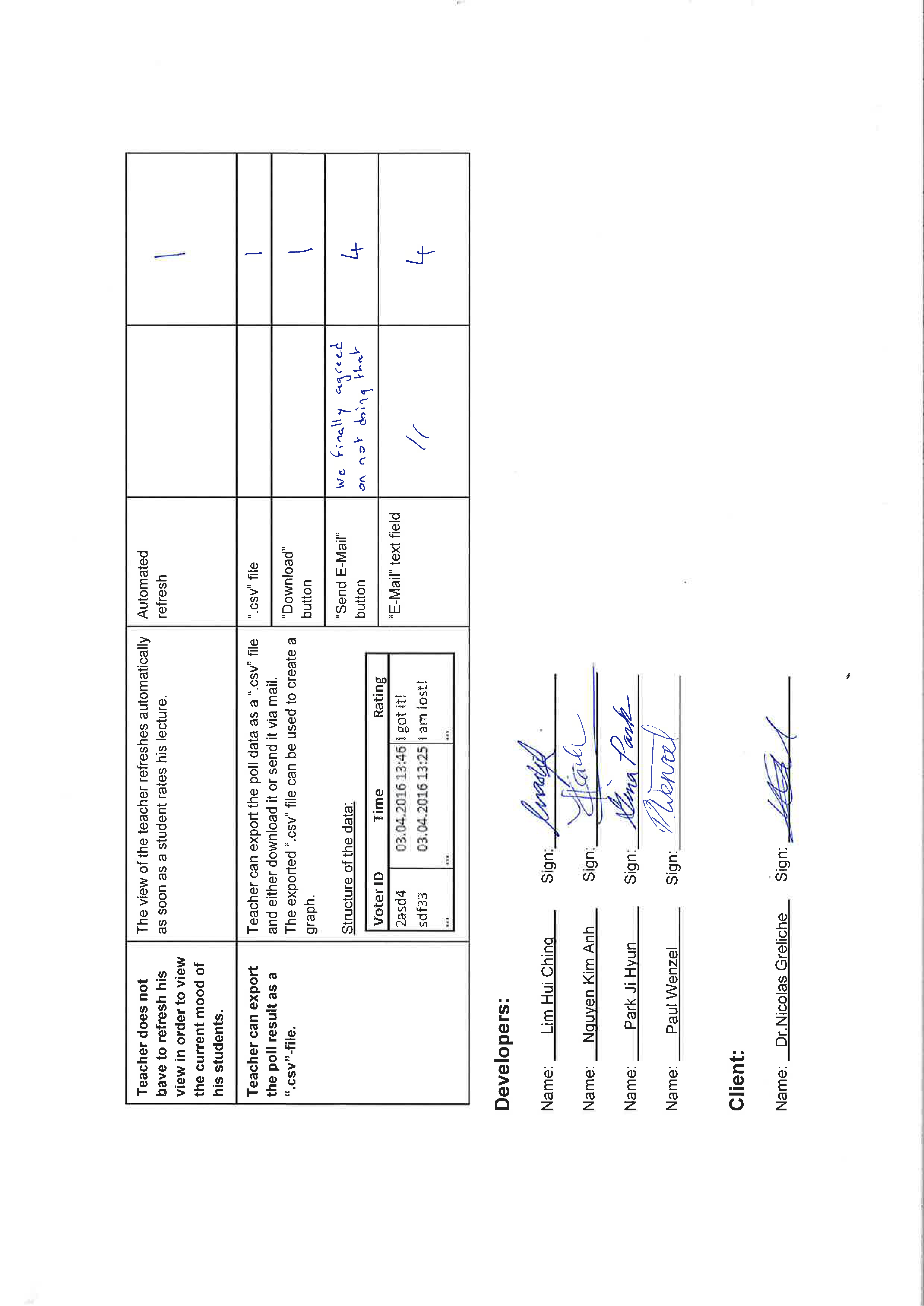




**Client Feedback for Milestone 2 (Alpha Release)**







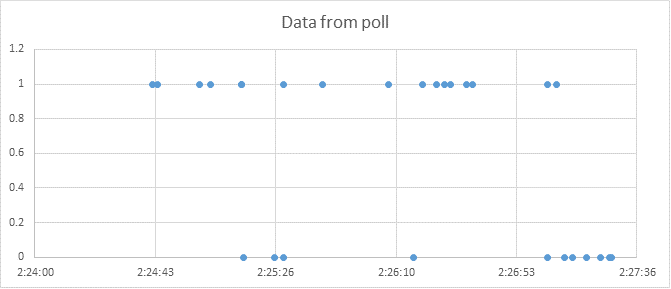
**Client Feedback after our client tested Lecture Poll Web App on his own**

**From:** Greliche, Nicolas [mailto:nicolas.greliche@jcu.edu.au]   
**Sent:** Samstag, 7. Mai 2016 14:49  
**To:** Paul Guenter Bruno Wenzel <paulguenterbruno.wenzel@my.jcu.edu.au>  
**Subject:** Lecture Poll - some very quick feedback

Hi Paul,

I just tried the app with 8 connections using a few different browsers. It worked well and smoothly.

I downloaded the data and made this veeeery quick plot below (1 being someone who got it, 0, someone who clicked they didn’t).



We can see well when people got it or not. At the end, people were a bit lost for example.

So my feedback is very positive.

Good job ;)

Cheers,

**Nicolas Greliche, PhD**

**Learning Advisor in Statistics and Mathematics,**JCU Singapore

T +65 6709 3791

**How we reacted to Client Feedback of iteration one**

Overall our client was really satisfied with our work in iteration one. He only had two points he wanted us to improve during the second iteration.

First, he wanted us to improve the user experience on the home page by clearly separating the login form from the create new poll form. The following screenshots show how we adopted to this feedback and improved the home page.

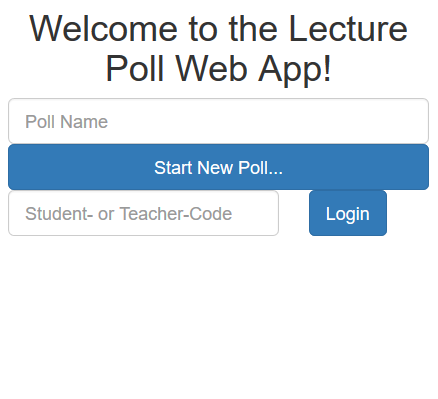


Figure 20: Home page after iteration 1

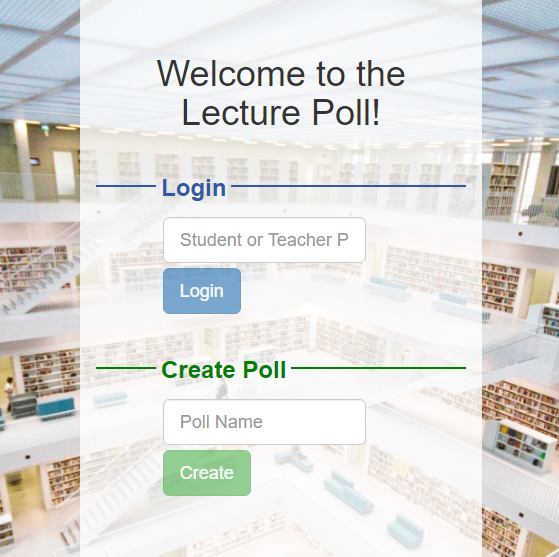


Figure 21: Redesigned home page after iteration 2

Our client really appreciated those changes and is happy how the home page looks in the alpha release.

Second, our client asked us to show the current status of the student on the student page so that the student always knows what he lastly voted for. During iteration two we also took care of this feature and followed his feedback by implementing a smiley face on the student page that represents the last given rating by the student.



Figure 22: Redesigned student page - green smiley after student clicked on "I got it!"



Figure 23: Redesigned student page - red smiley after student clicked on "I am lost!”

During iteration two our client told us that he is not interested anymore in the option of sending the poll data via e-mail since being able to download the data is already enough in his opinion. Instead he wanted us to refactor the code to a really high degree, so that he is easily able to develop the application on his own. Since our team agreed that there is no need for a second way to retrieve the poll data and we are also interested that the Lecture Poll Web App gets used in the future, we agreed with the client and intensively refactored the code.

**Burn-Down-Chart**

The burn-down chart is a chart that helps development teams to have a visual representation of the status of the project. On the one hand a burn-down-chart shows how much work is left before the project is completed. On the other hand, it shows if the team is within the planned schedule. Burn-down-charts are a common tracking tool in agile practices.

Our team used the burn-down-chart to track our work in order to make sure that we stay in time with the implementation. In addition to that we used it for showing the current status of the project to our client.

As shown in the burn-down chart above, our team has successfully implemented all user stories in iteration one. Our team had over-performance and finished the entire milestone earlier than expected. Most of the user stories in iteration one have a high priority which means they are really important to our project. That is also why we focused hard on making sure that we get them all implemented.

In iteration two, all user stories got implemented as well. We only did not implement two requirements of the last user story with a low priority. The two requirements that got not implemented are ‘send E-mail button’ and ‘E-mail’ text field. We did not implement those because our client wanted us to refactor the code on a high level instead. This is why we stopped developing before the due date of iteration two and instead focused the rest of the time on refactoring the code.

**Velocity Chart**

A velocity chart fulfils two functions. First it tracks how much work a team completed during past iterations. Second, it helps a team to estimate the amount of work the team can get done in future iterations. In summary, a velocity chart makes it possible to estimate the duration a team will need to implement specific requirements.

We used the velocity chart to track our work and estimate if we are able to implement all requirements defined by the scope. Moreover, we used it for planning a realistic beta and final release scope.

The velocity chart above shows how we performed during each week of iteration one and two. As you can see, and already shown by the burn down chart as well, we managed it to complete all requirements as committed. Like mentioned before we did not implement two requirements of the last user story from iteration two because the client wanted us to focus on refactoring instead. Those unimplemented requirements are represented in the velocity chart by the grey are of the second blue bar of iteration two. Since the client asked us not to implement them we had not committed to implement those requirements at the end of iteration two. This is why this area is not shown in blue but instead shown in grey.

**Undelivered User Stories**

All user stories got delivered. We only did not implement two requirements of the last user story with a low priority. The two requirements that got not implemented are ‘send E-mail button’ and ‘E-mail’ text field. We did not implement those because our client wanted us to refactor the code on a high level instead. This is why we stopped developing before the due date of iteration two and instead focused the rest of the time on refactoring the code.

Overall we delivered all user stories and only skipped two low priority requirements like demanded by the client.

**How we used the Burn-Down-Chart and Velocity Chart to plan the Beta and Final Release**

The beta and final release scope shown above are based on the velocity we were possible to deliver during iteration one and two of the alpha release.

During the alpha release our average velocity was (43 + 35) / 2 = 39. Our velocity from the first iteration (43) was slightly higher compared to our velocity from the second iteration (35). The reason for that is, that we focused on refactoring the code at the end of iteration two. We learned during the alpha release, that refactoring is really important to make further development efficiently. In addition to that, a refactored code is much easier to understand for new developers who join the team and also the client in case he is interested in the code.

We applied our knowledge earned during the alpha release, for planning a realistic beta and final release scope. We took the average velocity of each alpha release iteration as our estimated velocity for the beta and final release. Doing so leaves us some time for refactor the code after each beta and final release.

**[\_\_\_\_\_/20 marks, SLO-1, SLO-2, SLO-3, BLO-K1, BLO-S3, MLO-K1, MLO-S4, MLO-S5] Demonstrate the Project development and release ICT infrastructure.** This must include development environment, programming languages, source code repositories (Configuration Management), project collaboration tools, and development tools. Write here: minimum **TWO** pages, maximum **TEN** pages.

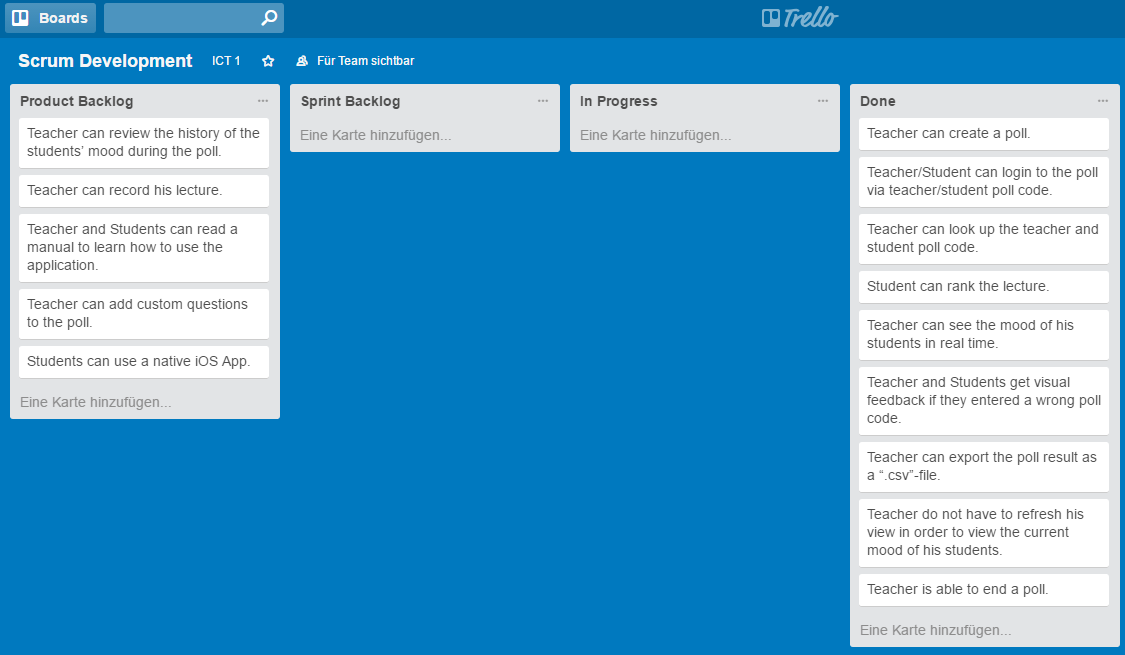
* [\_\_\_/10 marks] **Configuration Management**/version control, e.g. git, github, heroku, bitbucket; Project tools. Programming languages/IDEs. Building tools/procedures. How to set-up your development/release environment for a new team member.

**Collaboration Tools Used**

Collaboration Tools are needed to enable an easy and fast communication between all team members. Furthermore, they assist a team to organize itself and distribute tasks effectively. For ICT 1 we used Trello, WhatsApp, E-mails, Google Drive and Bitbucket as our collaboration tools.

1. *Trello*

We are organizing and allocating our tasks to each team member by using Trello. According to Scrum we used four lists representing the product backlog, sprint backlog, tasks currently in work and tasks marked as done.



1. *WhatsApp and E-mail*

For the communication between the team we mainly used WhatsApp. We created one WhatsApp group for all team members. It enabled us to share the status of tasks really easy and to plan appointments for team meetings and client meetings. The benefit of WhatsApp was that everyone in the team is using it in their free time as well and therefore checking WhatsApp messages regular. This lead to a quick response time.

For the communication with the client we used E-mails. It was a great way for sharing documents with the client and setting up appointments for meetings.

1. *Google Drive and Bitbucket*

We used two version control systems to manage all of our files. First, we used Google Drive to share all Word, Excel and PowerPoint files. Google Drive was really helpful since it keeps track of every single change made to a document. In addition to that it enabled us to edit files really quickly without even having to download the file, by using the online editors provided by Google.

Second, we used Bitbucket to host our Git repository. By using Bitbucket everyone from the team was able to retrieve and share the latest changes made to the code.

**Development Tools Used**

The following paragraphs explain what development tools we used and how we used the tools.

1. *Eclipse for PHP (PDT)*

Eclipse for PHP has been used as the integrated development environment (IDE). All the programming has been done using PDT. PDT specially supported us during programming, by providing syntax highlighting and autocomplete capabilities. One big advantage of PDT is that it supports all the different languages that we needed for developing our web application: PHP, JavaScript and HTML/CSS.

Furthermore, we used the Git plug-in for Eclipse to easily manage our Git repository by not having to use a command line but instead a rich user interface.

1. *XAMPP*

For testing purposes we needed a webserver and a database running on our local machine. Using XAMPP enabled us to set up a virtual server environment within a few minutes. We used XAMPP to set up an Apache PHP webserver and a MySQL database.

After the virtual server environment was set up, we were able to run and test our application easily.

1. *Git*

We used Git as our source code management system. It enabled us to share the written code between all team members. In addition to that, Git is merging all changes automatically. This enabled us to work on the programming code at the same time without having to worry about synchronizing all the changes manually afterwards. Moreover, Git keeps track about all the changes made to the code. Because of that, we were able to test out some ideas without having to worry about destroying the rest of the code.

1. *Microsoft Word, Power Point and Visio*

Microsoft Word, Power Point and Visio have been used to design prototypes, technology infrastructure and processes describing how the user is supposed to use the application.

The prototypes of the screens were especially helpful to come up with first ideas for the application and reflecting about it afterwards. Visualizing the user’s workflow in the application helped us to spot disadvantages in the design of the application in an early stage. Describing the technology infrastructure of the application helped to ensure that every team member understands what modules have to be developed and how they work together in the system.

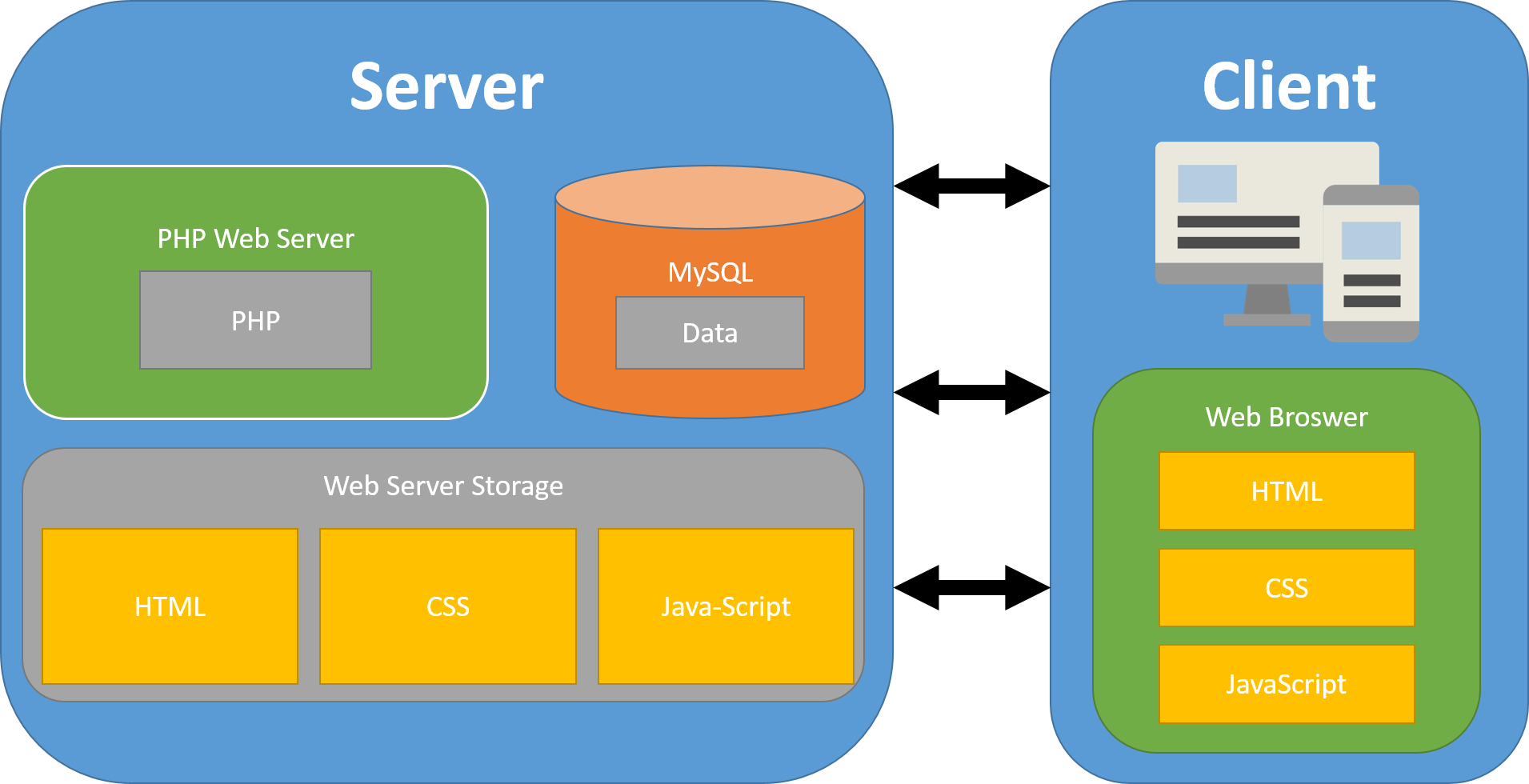
All in all, visualizing the ideas and concepts the application is based on, guaranteed that we and the client share the same vision about how the application should look like and behave. It also made it easier to reflect about the concepts of the application, before investing a lot of effort in developing the application.

**Set-up development environment for new team member**

The following illustration visualizes the steps a new team member has to accomplish in order to get fully integrated into the team and its work.

**Lecture Poll Infrastructure**

The Infrastructure of the “Lecture Poll Web App” is using client side technologies as well as server side technologies. The following paragraphs explain exactly which technologies are used and how they work together in the system.



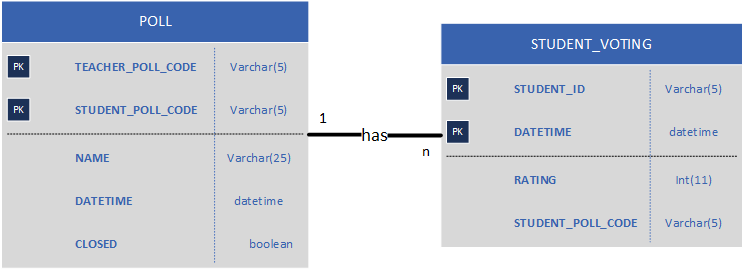
Server-Side Technologies

On the server side we are using a PHP web server and a MySQL database management system. In addition, files getting stored in the web server file directory/storage.

1. *MySQL database management system*

To store all the data required for our application we are using a MySQL database. The Database gets accessed by the PHP web server how either reads data from the database or writes data in the database by executing SQL statements.

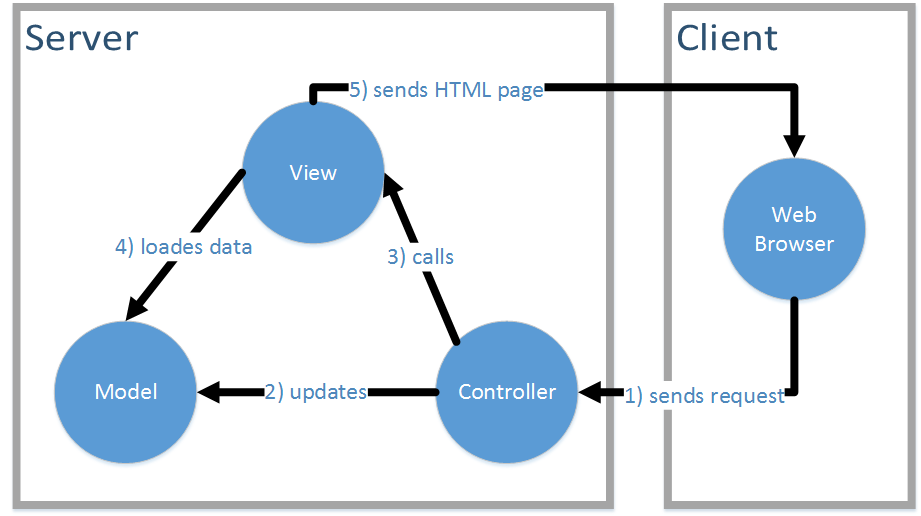
The database consists of two tables. First, the table “POLL” which stores the data about all polls. Second the table “STUDENT\_VOTING” which stores each voting of a student. The following Entity-Relationship-Diagram shows the exact structure of the database including the datatypes of each attribute.



1. *PHP Web Server*

A PHP webserver is used to generate HTML files dynamically. Therefor, PHP code is defined within the HTML pages stored in the web server storage. Before those HTML pages get send to the client they get processed by the PHP web server. Through the PHP code a database connection gets build and data gets read from the database and written in the HTML code.

In order to produce clean code that can also be reused in other applications easily, we followed the model-view-controller pattern (MVC). The model manages all the data from the application. Therefore, it is the interface to interact with the database. The view is the HTML page that gets generated and seen by the client. It loads the data it needs from the database by using the model. The controller consists of all the business logic of the application. On the one hand, the controller receives user requests and calls the requested view. On the other hand, the controller updates the data in the database and retrieves data from the database by using the model. The following figure also shows how model, view and controller are interacting with each other.



1. *Web Server Storage*

The web server storage saves all the static HTML, CSS and JavaScript files that can be requested and downloaded by the client.

Client-Side Technologies

On the client side the data send by the server gets interpreted by the web browser of the client. Three technologies are used on the client side: HTML, CSS and JavaScript.

1. *HTML (Hyper Text Markup Language)*

HTML is used to structure the data that the client is supposed to see. The client receives HTML files which get generated dynamically by the PHP web server. In the HTML files further CSS and JavaScript files get included to style the data represented by the HTML file and integrate dynamic user interface components.

1. *CSS (Cascading Style Sheets)*

CSS is used to style the HTML files. The CSS files are static files stored on the web server and can be downloaded by the client. On the one hand the CSS files set the color, margin, font style and size etc. of HTML elements. On the other hand, the CSS files are also used to define a responsive design. This means that the structure of the data changes depending on the screen size of the client.

For our application we are using the CSS framework Bootstrap (<http://getbootstrap.com/>). Bootstrap already comes with some predefined styles for most of the HTML elements. Moreover, it defines a grid layout which we used for making our application responsive.

1. *JavaScript*

For including dynamic user interface components into the HTML pages, we are using JavaScript.

Furthermore, JavaScript is used in order to refresh pages automatically without having to reload the entire page. The JavaScript files are static files stored on the web server and can be downloaded by the client.

For our application we are using the JavaScript library jQuery (<https://jquery.com/>). jQuery provides us with methods to easily navigate and manipulate a DOM document. In addition to that, there are a lot of free and open-source user interface elements around based on jQuery. In the “Lecture Poll Web App” the charts that are visible on the teacher page are based on jQuery, for example. Last but not least, jQuery makes it easy to make asynchronous xml requests also known under the “AJAX” concept. We are using it in our application for refreshing the teacher page automatically without reloading the entire page and for sending the students voting to the server without having to reload the page.

* [\_\_\_/10 marks] **Prototypes are demonstrated to justify the proposed beta-release**.

**Prototype Agreement**

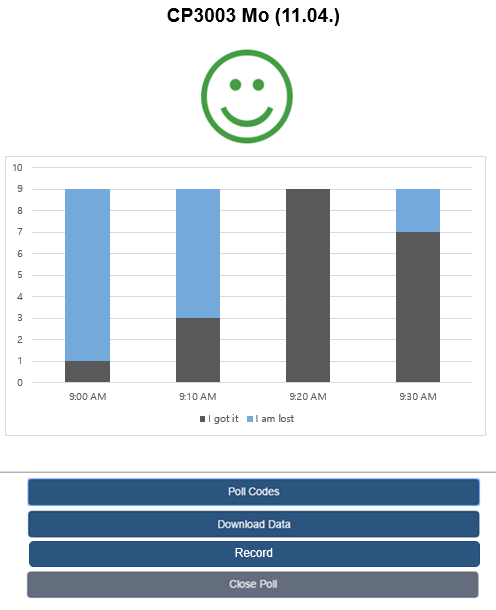
**“Lecture Poll Web App”**

**(Final Release ICT 2)**

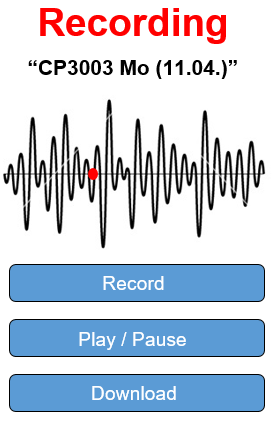
**Manual for Teachers and Students:**



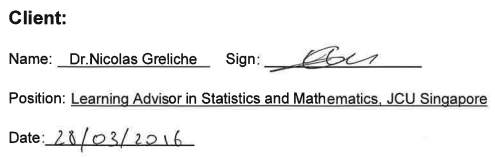
**History-Graph for teacher page:**



**Recording Options for the Teacher:**







**Attachment**

The following mail proofs that our client agrees with the scope and prototypes and that he is fine that we inserted his signature digitally.

**From:** Paul Guenter Bruno Wenzel [<mailto:paulguenterbruno.wenzel@my.jcu.edu.au>]   
**Sent:** Monday, 11 April, 2016 2:27 PM  
**To:** Greliche, Nicolas  
**Cc:** Nguyen Kim Anh ([nguyenkimthanh1410@gmail.com](mailto:nguyenkimthanh1410@gmail.com)); Park Ji Hyun ([jihyun5496@gmail.com](mailto:jihyun5496@gmail.com)); Hui Ching ([mashilim@gmail.com](mailto:mashilim@gmail.com))  
**Subject:** Final Scope Agreement & Screenshots

Hello Nicolas,

Attached you find the final scope agreement and some prototyped screens. Please tell us if you have any doubts or questions.

Since we have to get both agreements signed, I just took the signatures from the previous scope. I hope you are fine with that. Doing it this way we do not have to waste paper. But if you prefer signing both agreements individually, we can of course do that as well.

Best Regards

Paul

**From:** Greliche, Nicolas [mailto:nicolas.greliche@jcu.edu.au]   
**Sent:** Dienstag, 12. April 2016 10:50  
**To:** Paul Guenter Bruno Wenzel <paulguenterbruno.wenzel@my.jcu.edu.au>  
**Cc:** Nguyen Kim Anh (nguyenkimthanh1410@gmail.com) <nguyenkimthanh1410@gmail.com>; Park Ji Hyun (jihyun5496@gmail.com) <jihyun5496@gmail.com>; Hui Ching (mashilim@gmail.com) <mashilim@gmail.com>  
**Subject:** RE: Final Scope Agreement & Screenshots

Hi Paul,

Hehe, I do that sometimes too (the signatures copied), so I’m good with that.

[…]

Cheers,

**Nicolas Greliche, PhD**

**Learning Advisor in Statistics and Mathematics,**JCU Singapore

T +65 6709 3791

**Marking Rubric for Project Audit:** This assessment rubric provides you with the characteristics of exemplary, competent, marginal and unacceptable work in relation to task criteria.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **Exemplary (DISTINCTION-level)** | **Competent (CREDIT-level)** | **Marginal (PASS-level)** | **Unacceptable (FAILED-level)** |
| **Individual Team Feedback form is submitted to LearnJCU electronically. Assignment is done in a group with 2-4 students.** | **20**  Group demonstrated **exemplary ability** to work collaboratively, e.g. all team members always attended the same workshop. Feedback is objective to other members and for self-improvement | **15** Group demonstrated **competent ability** to work collaboratively, e.g. all team members nearly always attended the same workshop.Feedback is fair to other members and for self-improvement | **10**  Group demonstrated **marginal ability** to work collaboratively, e.g. all team members sometimes attended the same workshop. Feedback is not objective to other members and/or does not identify any area for self-improvement | **0**  Group did not demonstrated ability to work collaboratively. |
| **Report and demonstrate the ACTUALLY delivered alpha-release** | **40**  All delivered user stories are correct, correctly implemented, and tested. Technical information in the report is exemplarily. High quality presentation and formatting of the report. Delivered alpha-release is as per iteration-1 planning. | **30** Minor errors in: All delivered user stories are correct, correctly implemented, and tested. Technical information in the report is competent. | **20** More than half of delivered user stories are correct, correctly implemented, and tested. Technical information in the report is PASS-level. | **0**  Not done, or done unacceptably. |
| **Client signed acceptance of the alpha-release, and the proposed beta- and final-releases** | **20**  Clear evidence of client’s acceptance and running of the alpha-release. Clear and meaningful feedback on the alpha-release. Signed plans for beta- and final releases. And all user stories are correct, and correctly estimated | **15** Some evidence of client’s acceptance, and/or client is NOT running alpha-release. Signed plans for beta- and final releases. Or most user stories are correct, and correctly estimated | **10**  Minimal evidence of client’s acceptance, or client is NOT running alpha-release. Signed plans for beta- and final releases. Or more than half of user stories are correct, and correctly estimated | **0**  Not done, or done unacceptably.  No signed plans for beta- and final releases. Or less than half of user stories are correct, and correctly estimated |
| **Demonstrate the Project development and release ICT infrastructure** | **20**  Description very clearly communicates exemplary ICT solutions. | **15** Description clearly communicates competent ICT solutions. | **10**  Description communicates ICT solutions. | **0**  Not done, or done unacceptably. |

## Reference BIT & MIT course and learning outcomes

|  |  |
| --- | --- |
| BIT course & CP3046 subject learning outcomes | * SLO-1 Communicate technical information clearly through presentations, demonstrations and documentation; * SLO-2 Choose and apply the appropriate agile methodologies; * SLO-3 Evaluate and select appropriate tools and technologies to meet project requirements * SLO-4 Develop a general knowledge of industry standard project management approaches; * SLO-5 Critically reflect on progress to tailor self- learning goals to advance professional development. * BLO-K1 - Demonstrate essential knowledge for a career in technology related professions and practice; * BLO-K2 - Synthesise industry standard and underlying principles and concepts for decision making; * BLO-K3 - Critically analyse the core professional obligations, values and operations of organisations including sustainability; * BLO-S2 - Apply critical thinking to address IT related issues; * BLO-S5 - Demonstrate the ability to work collaboratively; * BLO-A1 - Apply technical skills, including numeracy, necessary for professional practice; * BLO-A3 - Identify and act upon, learning opportunities and self-improvements. |
| MIT course & CP5046 subject learning outcomes | * SLO-1 - Communicate technical information clearly through presentations, demonstrations and documentation; * SLO-2 - Choose and apply the appropriate agile methodologies; * SLO-3 - Evaluate and select appropriate tools and technologies to meet project requirements * SLO-4 - Develop a general knowledge of industry standard project management approaches; * SLO-5 - Critically reflect on progress to tailor self learning goals to advance professional development; * MLO-K1 - Critically analyse to select appropriate tools and technologies to meet project requirements; * MLO-K2 - Evaluate industry standard contexts for project management approaches; * MLO-S4 - Apply lateral and original thinking to conceptualise and evaluate a range of solutions to relevant problems; * MLO-S5 - Communicate technical information clearly through presentations, demonstrations and documentation; * MLO-A1 - Apply appropriate tools and technologies to meet project managements; * MLO-A3 - Critically reflect on progress to tailor self-learning goals to advance professional development. |