Terry Sheridan.

Supervisor: Vikas Sahni.

Technical Report.

BSIC – Mobile Application Development.

X13416168

Sheet for Project Submission

SECTION 1 *Student to complete*

Name:
Student ID:
Supervisor:

SECTION 2 Confirmation of Authorship

The acceptance of your work is subject to your signature on the following declaration:

I confirm that I have read the College statement on plagiarism (summarized overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature: **Date: 07/05/2018**

NB. If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the College's Disciplinary Committee. Should the Committee be satisfied that plagiarism has occurred this is likely to lead to your failing the module and possibly to you being suspended or expelled from college.

TABLE OF CONTENTS

ABSTACT

The number of college places is not increasing at the same rate as the number of students taking their leaving certificate examinations. This gap continues to widen annually, meaning the percentage chance for each student to get accepted into college is decreasing. If the government do not begin to address this, the gap is going to continue to grow at an unprecedented rate.

As this gap grows, the academic bar and the level of pressure each students face will rise. Our education system needs to be able to cope with these changes and ensure to provide the best means possible to all its students. There is no way to automatically change this system overnight, but there are ways to help improve it.

Communication between teachers, students and parents is a critical factor in reflecting student performance. If this is strengthened some problems will be avoided. Giving parents and students meaningful insights into academic performance via the use of dynamic charts, monitoring grades, attendance and behavior for each class attended, will ensure no nasty surprises await come annual examination periods.

Student data is captured by each teacher and stored in a dynamic interactive database. From here schools can track students' performance in finer detail giving teachers the ability to make informed decisions in less amounts of time. Benefiting each of their students.

The SKEWLEY system provides students with the ability socialize, address any concerns they may face and stay updated with all school work via the implementation of a social media section, which is monitored by the school ensuring students will not be bullied so they can voice their opinions freely. The system intends to provide support for each student, a step in the right direction for changes to come.

INTRODUCTION

I.I BACKGROUND

Combining a vast mixture of internal and external factors, inclusive of population growth and socioeconomic culture, competition within the Irish Education System is steadily increasing - *"The number of Leaving Cert students taking on higher-level subjects has climbed to a record high as pupils try to make the most from changes to the CAO points system." (The Irish Times, 2017).* Another important factor indirectly responsible for increasing levels of competition within the Irish education system is down to new reforms introduced by the Board of Education. Two influential aspects are: (1) – *"any student who fails an exam with an E grade at higher level will secure CAO points" (The Irish Times, 2017)*- Increasing the range of points allocated per subject will decrease the mean rate of skew per subject, - competition will increase due to the gap of achievement decreasing, making differentiation (on average) more difficult.

(2) – *"a record 36 per cent have registered at higher level to sit the more challenging paper, up from about 20 per cent in 2011.The increase follows the introduction of bonus points for higher-level math's" (The Irish Times, 2017)* – The introduction of bonus points for sitting a more difficult paper puts pressure on students. Even if the increase in workload equates to the additional points allocated, more students are taking the more difficult paper – increasing competition. It can be argued that - the higher the rate of "constructive criticism and encouragement initiatives", the heavier the academic burden per student. The below chart represents the increase in students taking Higher Level Math's at Leaving Certificate level.

A clear increase in the amounts of students sitting higher level Math's in recent years in depicted above. This continuation is correlative to the additional pressure students of the future are to face as more of their peers take the leap to do so. Yes, all students who make this leap will be awarded fairly and separated from those who do not take the leap. This results in students who take the lower level paper under-valuing themselves in many cases. Regarding those who are becoming more likely to take the more difficult paper, the level of stress they face continues to grow – which also will have a negative impact on the mental health of many students.

Finland consecutively tops education rankings worldwide - currently ranked 5th in the world. It is famous for having a no banding system, meaning that all pupils, regardless of their ability are taught within the same classes. The result of this is the gap between the strongest and weakest pupils is the smallest in the world. Relative to this, The National Geographic has recently rated Finland as being the 5th happiest country to live in the World. The correlative nature between Finland's education system and its level of happiness is clear.

On the contrary, Japan, with an Education System ranked 2nd in the world, employs an ultra-completive academic environment for all its students, excelling in Math's and Technology, students must pass a continuation of tests to attend Junior/Senior High schools of choice. Top ranked school's entrances exams vary. Many students must prepare for these exams in the 4th and 5th grade of elementary school, emphasizing the immense academic pressure they face from a very young age. – "*A recent study published by Japan's Cabinet Office examined the country's more than 18,000 child suicides from 1972–2013 and found that 131 of those suicides occurred on September 1—32 more suicides than the next most-frequent date of death. Mapping for frequency, the study found distinctly larger numbers of suicides at the end of August and beginning of September, as well as during the middle of April. The former coincides with schools reopening after summer vacation; the latter, as schools reopen after spring break.*" (Newsweek, 2015).

In comparison to the former, Ireland's education system is becoming more like Japans education system and will continue to do so in years to come as competition arises. The division, although rewarding to those capable, needs addressing before it progresses to something unchangeable. It is important that the current Education is accepted for both its pros and its cons. Students are going to be put under more pressure and their mistakes amplified as the bar rises. A suitable support system is needed for all. Although the Skewley system cannot resolve these pressing issues alone, it is a step in the right direction for connecting the dots between parents, students and teachers, by providing a means of support to students, getting parents more involved in students' lives, and strengthening the communication barrier between all, which will positively impact the student.

The Skewley system aims to provide parents with detailed representations of their Children's academic performance. The areas it monitors and updates parents regarding are derived from the three pillars of academic excellence, Grades, Attendance and Behavior. This data is provided by each of the students Teachers. Feedback is given per class, all results are saved for parent-teacher meetings, One of the Behavior metrics includes "please contact" which encourages engagement and for Parents to directly message the teacher, which will strengthen relationships. Parents are given a clearer insight into their Children's education, which will allow them to have a stronger impact in the decision-making process and strengthen support when needed. This may include

booking support classes, or even making it easier for students to explain exactly why things are happening and not hide the fact of them happening or not, which is undeniably important.

In addition to encouraging Parent/Student, Teacher/Parent, Teacher/Student communication, another important aspect of the Skewley system is to encourage Student/Student engagements/relationships. Students lack the ability to confidentially post about lessons/homework or ask about homework/misunderstanding on today's social media platforms. Skewley aims to provide this service by implementing a Facebook like social section which not only allows students to ask questions/post insights but encourages them to do so. These posts are also monitored by respected schools, meaning the likelihood of students being harassed by their peers and bullied, which are common occurrences on social media are greatly slashed.

The final way Skewley aims to help improve the current education systems comes down to its fore take on the automation process. Skewley aims to take pressure of Schools/Teachers by providing a system which houses data on all students. This data can be accessed from school/home and provides more meaningful insights via the use of high level dashboards which allow schools/teachers to speed-up their decision-making processes and not only make data backed decisions, but save time, which is an indirect benefit for the student.

I.II AIMS

The System aims to support each Student through the following:

1- Lessening their academic stress via strengthening school-related communication between their Parent and Schools.
2- Providing an environment where they can develop social relationships without the fear of being bullied by their peers.
3- Identifying issues faster giving them more time to deal with problems and lessen their academic burdens.

The idea of improving the current means of academic support available to all students today is the main goal of the Skewley System. To achieve this, it is intended to improve communication between Students and their Parents. The more Parents know about their Children's Education the more they can do to support it. This is to be done by the Students Teachers providing the Students Parents with more detailed and frequent feedback. One advantage of this support mechanism is a more realistic and fair approach to setting Student targets and goals, taking more pressure off the student whilst benefiting overall performance.

The other main way the System aims to support students is by creating a social platform where they can freely seek support without the fear of being made fun of by their peers. With increased internet usage in younger generations, this problem continues to grow. It is aimed to solve this issue by encouraging all online posts to be school related, with the addition of having a weekly school review to ensure nothing of counter intuitive nature occurred. Relieving students of peer pressure will increase the chances of them using this site to benefit themselves and help them to make/stay in contact with friends when doing so.

The system aims to support Schools and Teachers by automating many of their daily tasks which are inclusive of:

4- Attendance.
5- Recording of Student Results.
6- Representation of student performance per class.

The objective of the above is to give the School a deeper understanding of the student on a prolonged scale in addition to seeing how student attendance correlates with grades, how student's averages match up to the rest of their class/year group, and ultimately to allow them to begin predicting the possibility of student's performance based on recorded data and give them more time to implement a corrective approach to help improve this.

This data will be vital in helping Parents with their decision-making process and will help predict trends and patterns in student performance which would not have been possible without the implementation of such system. The student attainment section where all results for Students and their respected classes can be found in the Administration panel of the web application.

The system aims to support Parents by:

7-Providing them with more detailed and meaningful news about their children's School life.
8-Alerting faster providing more time to react.
9-Improving communication with both the School, Teachers and Student via a direct form of communication (messaging).

Aims have been set very high not only for this systems development, but also for what it is intended to achieve once implemented, and how it all correlates with the bottom line – supporting the student. Therefore, it is vital for the System to adapt a professional look and feel which also meets these expectations. It is intended to achieve this by ensuring all front-end interfaces encourage Students to use the System.

I.II DEFINITIONS, ABREAVIATIONS, ACRONYMS

SDLC	Software development lifecycle.
SRS	Software Requirements Specification.
GUI	Graphical User Interface.
IDE	Integrated Development Environment
SQLI	SQL Injection
SRS	Software Requirements Specification.
GUI	Graphical User Interface.
IDE	Integrated Development Environment

I.III TECHNOLOGY OVERVIEW

Cloud9 was selected as the IDE of choice for the projects development. This came down to numerous factors including; my familiarity with the environment having used if for projects in the past, its professional look and feel with its version control capabilities, and the fact it's pre-canned PHP workspace comes with Apache pre-Installed and allows you to run PHP code via the click of a button, connecting you straight to your accounts server.

Another Major benefit of using Cloud9 came down to the fact it provides the capability to easily integrate PHP workspaces with phpMyAdmin using a few quick installs via Cloud9s inbuilt terminal. This made making the applications vast, complex data structure much easier. Instead of having to write pre-determined SQL code which would involve having to both construct and populate numerous tables and then focus on their relationships, phpMyAdmin provided the functionality of adding new schemas via the click of a button, which you then could add columns/types/primary-keys, all via the use of simple button clicks which saved lots of time and also generated all underlying SQL code needed at the same time.

The front end of the system was developed using HTML, CSS, JAVASCRIPT, JQUERY, HIGHCHARTS and BOOTSTRAP. Having experience as a front-end developer I felt a combination of these technologies would not only benefit my current skillset but would also bring a professional look and feel to the application, should they be utilized correctly. High charts were used due to my experience using this dynamic library.

A custom MVC framework was used for the system to both provide a lightweight structure for the system to improve performance, and again due to my past exposure to said framework. PHP was selected as my rendering language of choice. Having no experience in this language, I decided to choose it due to most online social media related resources I came across utilizing it in conjunction for its performance and popularity amongst large scale web applications. Another main advantage of using PHP over Java for example, came down to the fact its Syntax was much easier to grasp and felt it would be of major benefit to the systems development. The fact it had been around for decades also reassured me there would strong resources and an online community should support be needed.

Composer was used for the main goal of installing PHP unit. PHP unit was selected as the testing framework due to it being recognized for being one of the best and most popular PHP testing languages of choice, due to it efficient, easy to use structure.

SYSTEM

II.I.I Requirements

The following section will discuss the data gathering, requirement building and requirement stages of the project. Three separate questionnaires were conducted on each distinct user group due to them falling into separate demographics in conjunction with the fact it is intended for them to use the system differently.

II.I.I Data Requirement means of getting data.

It was designed for the questionnaire for parents, teachers, and students to provide background information on the system and their thoughts about current forms of technology/lack of technology in their life's also.

The teacher survey was conducted from giving a teacher the teacher questionnaire to complete themselves in conjunction to giving it to 14 of their friends. The student questionnaire is to be given to my work colleagues in a social who have all relatively recently come through the Irish education system, hence will be able to suggests ways in which it can be improved. The parent survey was given to my Mam, to give to 14 of her friends (all parents).

The questionnaires were rated on a scale of one one to five based on a subject (1)agreeing to (5) disagreeing with said question. These initial questions vary based on subject group to provide qualitative information regarding the ideologies of each subject group, conducted numerically, but based on personal thoughts.

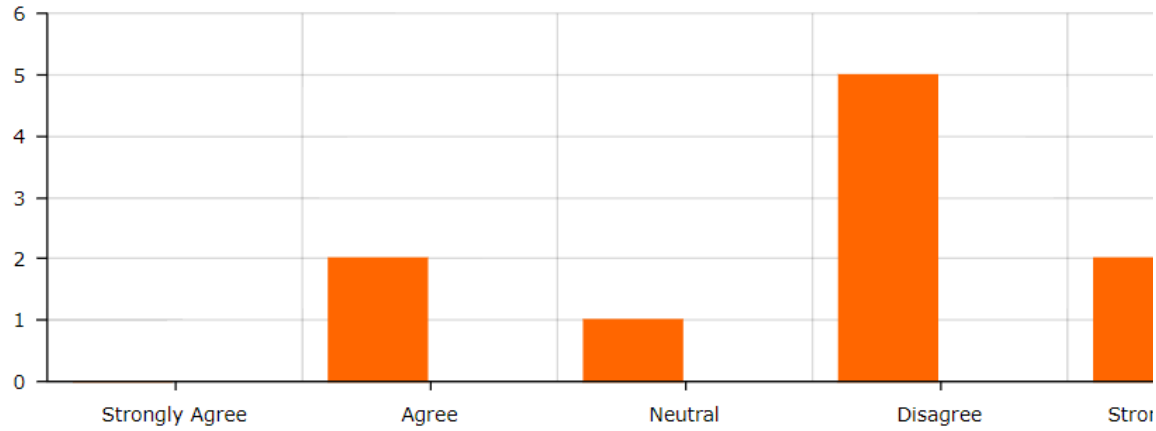
Each questionairre also contains a functionality scale where all users rate system funtionalitys based on their importance via the same 1 – 5 scales. (Quantitative) The first section which is different per group is to provide user group specific feedback, whereas the general functionality questions have been created to get an overall mean understanding of the systems components. These are both measured in a quantitative manner using the above numerical scale.

II.I.II Past Student Questionnaire.

Statement	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
General Statements					
There are many online resources available to help you with your school work.					
You use online resources to help you with your school work on a regular basis.					
On School nights you spend over 2 hours on social media per day.					
On weekends you spend more over 4 hours on social media per day.					
Lots of the content I see on social media is related to my school work.					
You would like for your parents to be notified about your school work on a more frequent basis.					
You or your peers are likely to be made fun of if you post something related to homework on social media today?					
If there was an easily available online resource for you and your peers to chat about classes, adopting a social media like look and feel, would you use it?					
Would you like for the social site to be monitored weekly by one of your teachers?					
Will a service like this help you catch up on missed school work?					
Functionality - Importance of the following being included;					
Messaging					
Notifications					
Friends Section					
Posts					
Viewing of grades					
Mobile phone responsive					
Comments					
Newsfeed					
Profiles					
Liking Content					
Simple Navigation					

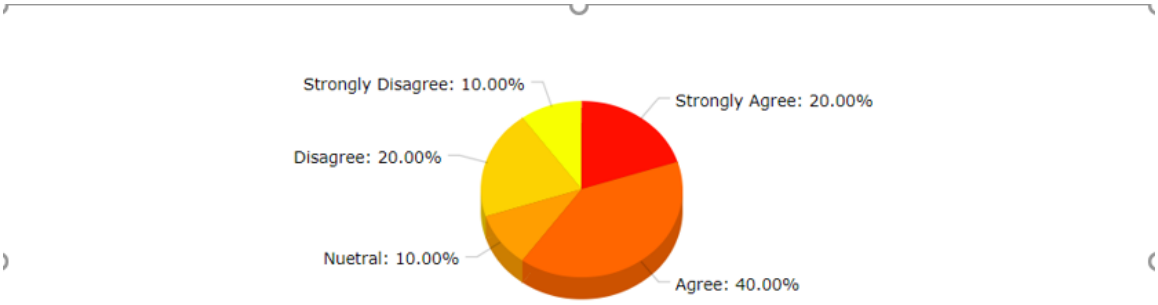
II.I.I Past-Student Questionnaire – Analyzation of Results.

The first question in the past student survey was created to get background information regarding past student's thoughts about if they felt there was many online resources available to help them with their school work. Please find the results of this question below:



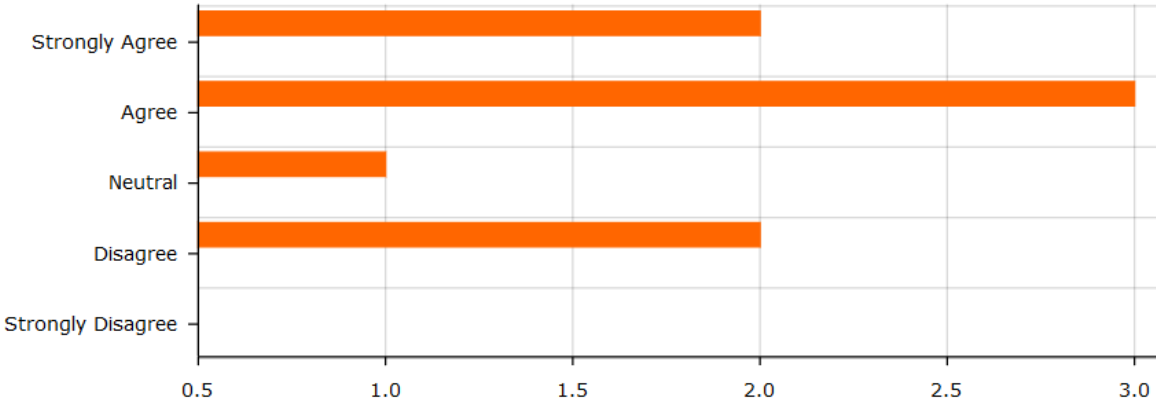
The above graph represents that out of the past students asked most disagreed that there were enough online resources available to help them. 20% strongly disagreed with this, whereas 20 percent agreed. This could potentially be down to the fact these students all attended separate schools.

The Second question which was designed to see if past student felt that they had regularly used online resource to help them with their school work. This was like the first questions however it was not student service specific and could be any online resource. Please find the results of this question below:



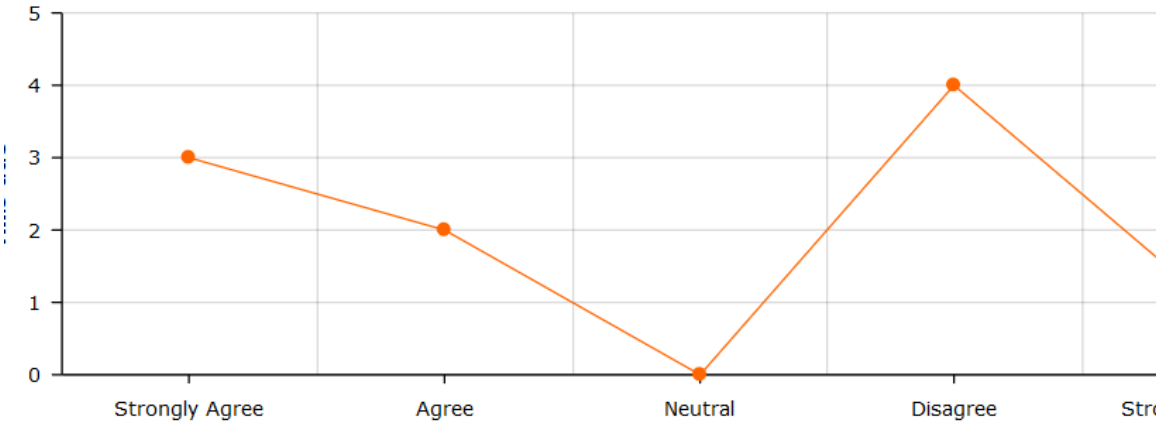
From this we can see there is a clear split between past students who agreed and strongly agreed with using online resources to those who disagreed and strongly disagreed. This may be due to the learner's academic style.

The third question in the academic survey was created to get a better understanding of the amount of time past students spent online during school nights.



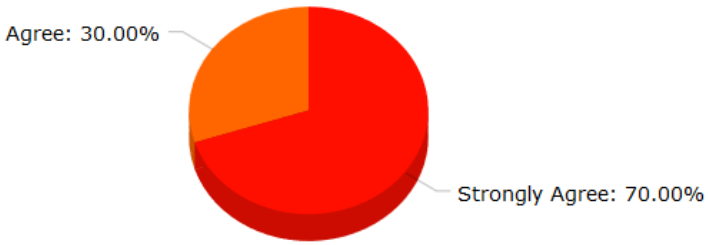
The results from this show that more students agreed with this statement however the answers for this question were very spread out. This is reflective of the fact the amount of time spent on social media varies per individual.

The fourth question in the academic survey was asked to determine if past students felt that they spent more than 4 hours on social media per day during their school years. Please find the results of this question below:



From this is can be determined that there was another even divide for the question asked. This could be influenced by the amount of extra-curricular activities a certain individual undertook and may also display different results if it were to be asked to a larger subject group from different academic backgrounds.

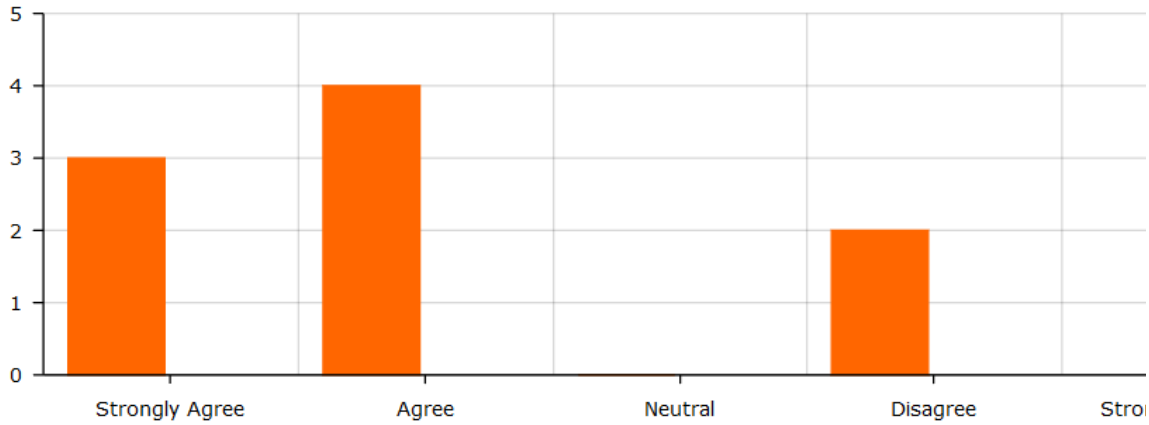
The next question pass students answered was regarding how much of the content they saw online they deemed to be related to school work. Please find the results of this question below.



It can easily be concluded that students strongly felt that most of the things they saw on social media was in no way related to their school work.

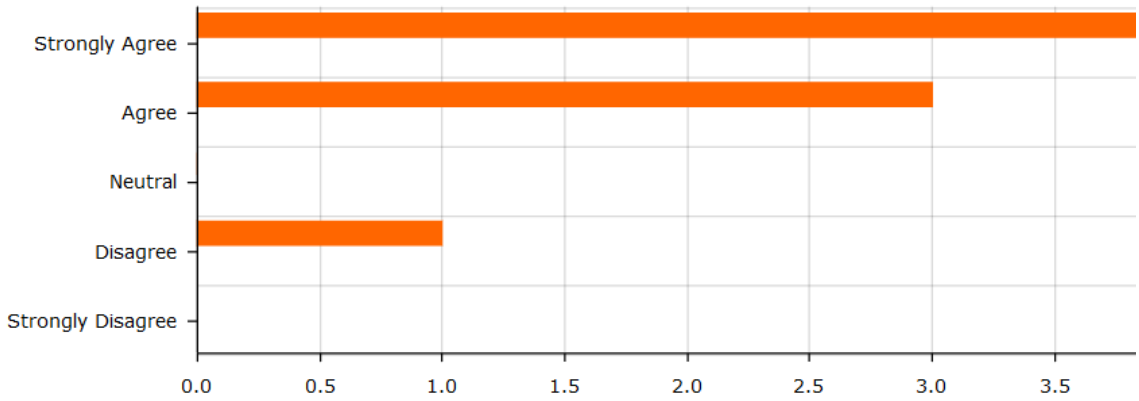
The 6th question in the survey was added to see how students felt about their parents being updated more regularly about their school work. This was a controversial question as the question

was asked to passed student who may now feel that it would have been of benefit to them if their parents were more frequently updated. However, it was emphasized to put themselves back in their school shoes, please find the results in relation to this question below:



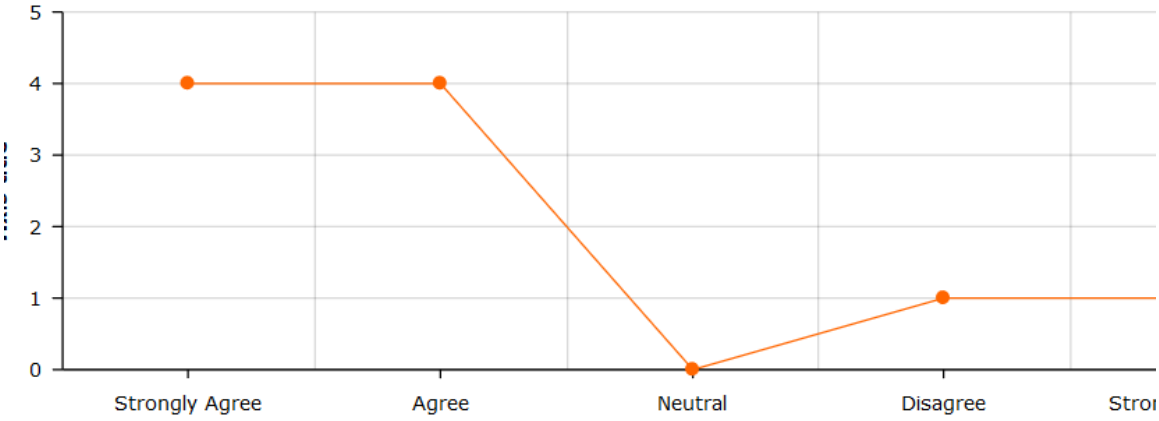
The above results suggest that most past students felt that having their parents updated more frequently about their results would be of benefit to them. However, 20% of students disagreed with the statement.

The next question in the above survey was created to determine how past students felt the probability of them being bullied or made fun of online under the assumption that they were to post school related content online.



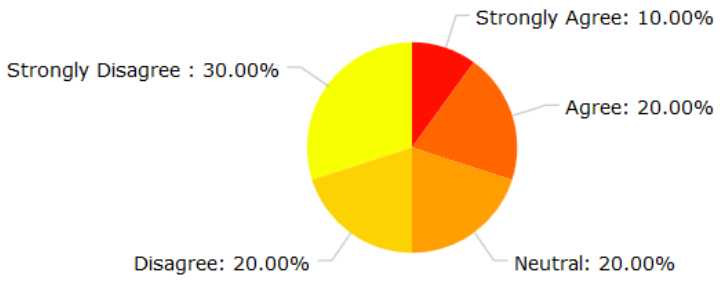
The alarming results conducted above show that 50% of students strongly agreed with the fact that they would be made fun of if they were to post anything school related on their social media profiles. Another alarming factor to these results show that this question was the only question no student strongly disagreed with.

The next question in the survey was created to determine if students felt they would use a social media platform tailored to talk about school related topics should there have been one at the time. Please find the results to these questions below.



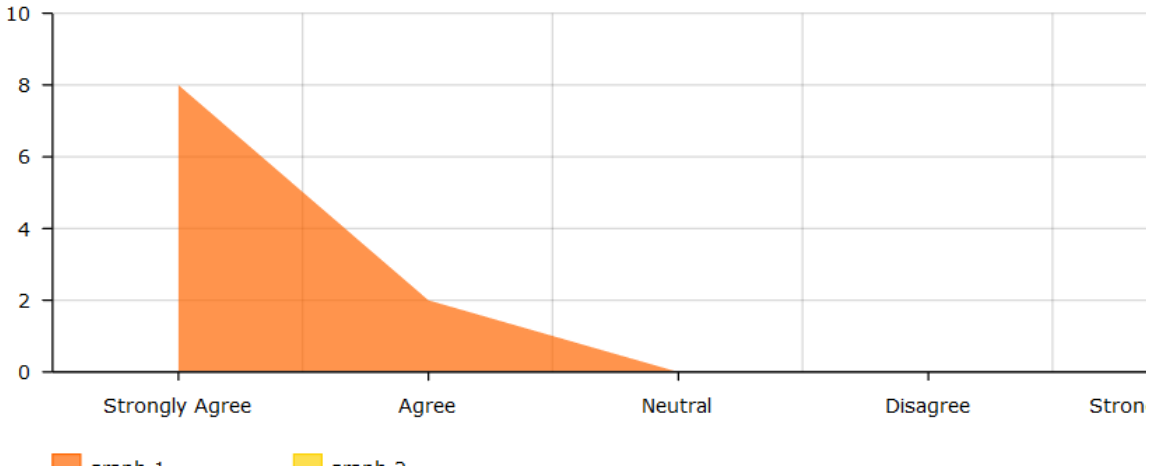
The above graphs show us that most past students (80%) felt as though they would have used a service like this should it have been available to them at the time.

The next question was created to see if past students felt that they would like the idea of a teacher monitoring their social media activity on a weekly basis. The results of the question can be found below.



These results showed an interesting lean towards students disagreeing with their social media account being monitored by their teachers. This could be down to them considering it an invasion of privacy.

The last question students were asked was short and simple. It was to see if past students felt that this type of service would be of benefit to them catching up on school work.



The results above show that 100% of students either agreed or strongly agreed that this service would enable them to catch up on any school work should they have missed any.

II.I.IV Past-Student Questionnaire – Conclusion.

Based on the answers provided from past students several things can be noted. To begin a consensus relating to past students feeling that there were not enough academic resources available to them while they were taking their leaving certificate examinations. This is correlative to the fact students feel this service is needed. Another of the key pieces of information concluded from the above questions comes down to the fact that past students were strongly under the impression that most things they came across online, even though a large percent of said spent lots of time on social media, was in anyway related to anything they were doing in school at the time. This goes to show the potential opportunity for a system like this should it be harnessed correctly and allow students to feel as though it is an outlet opposed to a responsibility.

Prior to conduction an assumption was made that participants would strongly disagree with their parents being more frequently updated about their education as a positive factor. However, the results concluded that this was not the case, 20 percent of students did strongly disagree with this though, but it shows potential for more influential parent support on students' academic life's being received positively from the past students.

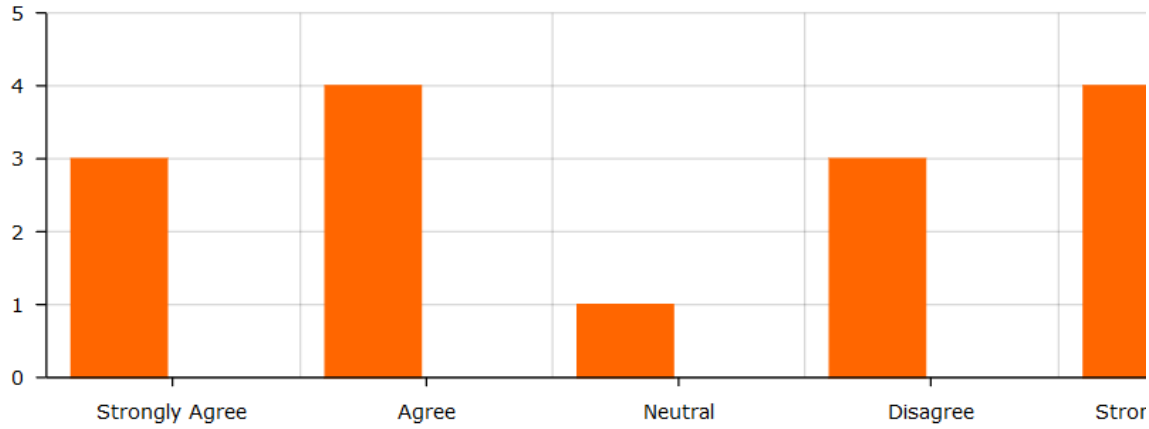
One of the most influential questions which found out if students felt as though they would be made fun if they were to post school related content on social media today. From this it can easily be determined that most students feel this to be true, which is a direct prevention from doing so. If students are unable to do this they need an alternative means of doing so, which is another reason for this system to be created. This coincided with the fact most students agreed with the that they would use a social media site like this should it have been provided to them at the time. Although some students disagreed with the idea of the site being monitored by their teachers on a weekly basis all students did feel as though a service like this would have been of benefit to them, based on this it can be concluded that the positives outweigh the negative regarding the probability of past students using the Skewley system.

II.I.V Teacher Questionnaire.

Statement	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
General Statements					
There are many online resources available to help you with work.					
If you saw a problem escalating with one of your students you'd like to have the option to reach out to said student's parents.					
You feel as though you are allocated enough time in the school year to effectively communicate with your students' parents.					
You spend more than 10 hours on social media per week.					
Having charts representing each of your student's grades/attendance/behavior would improve your decision-making process.					
Would having more detailed and quicker student evaluation data influence you to take corrective action quicker?					
You can see yourself using a system like this regularly.					
A system like this would be of benefit to my students.					
A service like this would boost clubs and extracurricular activities within the school.					
Functionality - Importance of the following being included;					
Messaging					
Notifications					
Friends Section					
Posts					
Viewing of grades					
Mobile phone responsive					
Comments					
Newsfeed					
Profiles					
Liking Content					
Simple Navigation					

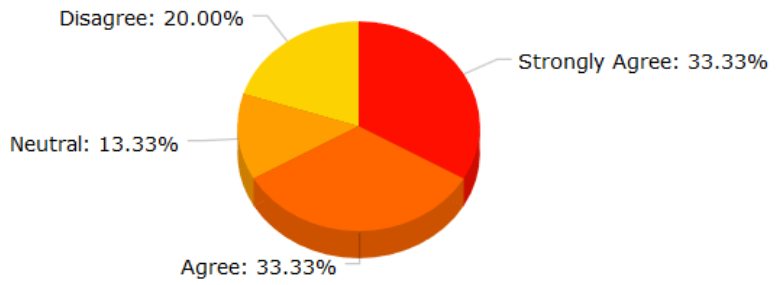
II.I.VI Teacher Questionnaire – Analyzation of Results.

Please find the results of the first question in the teacher questionnaire below:



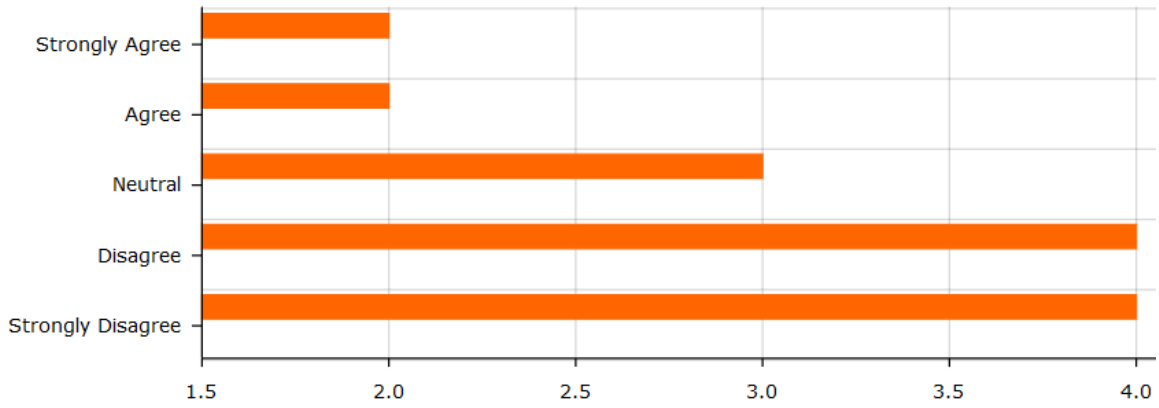
This question was directed at finding out if parents felt as though there were enough online resource available to them to help them with work. A clear divide can be seen in this answer, this may be down to teachers coming from different schools. However, the fact that half the teachers that almost half the teachers asked either disagreed or strongly disagreed with this system, allows us to identify the room for improvements regarding teacher's current online resources available.

The Second question in the questionnaire was designed to see weather or not teachers felt as though they would reach out to students' parents more frequently if they were provided with the option of doing so.



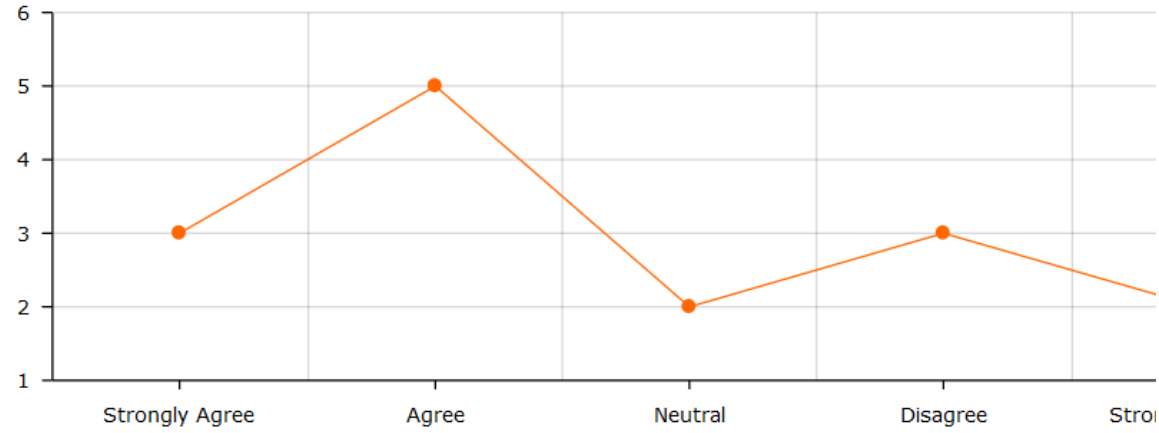
The above graph represents the responses to this question. From this it can be determined that 66% percent of teachers either agreed or strongly agreed with this statement which emphasizes one of the many benefits of the implementation of this system.

The next question in the survey was designed to see if teachers felt as though they are allocated enough time to communicate with their students' parents in the school year. Please find the results of this question below.



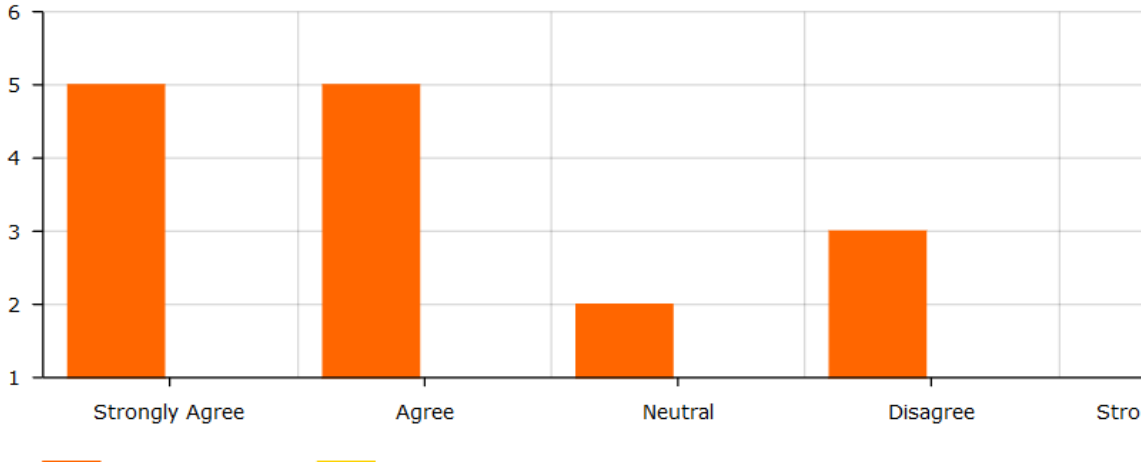
The above graphs show us that 8 of the 15 teachers asked felt that they were not allocated enough time in the school year to communicate with their student's parents effectively, this highlights the systems potential to resolve this issue and strengthen the communication process.

The next question in the survey was designed to get a more detailed representation of how much time teachers were currently spending on social media per week. To get a better reflection of the probability of them using another social site.



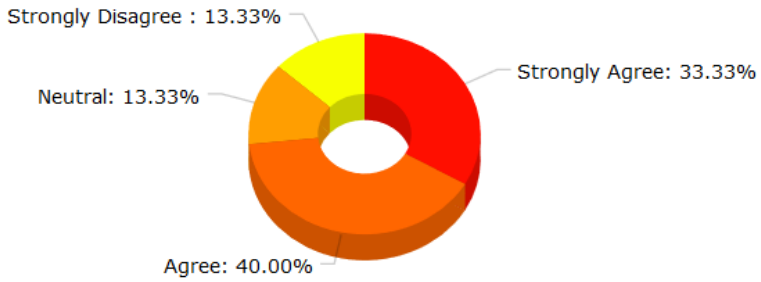
The above graphs show us that most teachers spend more then 10 hours on social media per week which is a positive factor regarding the chances of them using this service.

The 5th question was aimed to provide information regarding how teachers felt about the implementation of charts and graphs representing their student's performances and if this would be of potential benefit to their decision-making processes.



The above chart represents the fact that most teachers felt like an educational site like would be of benefit to their decision-making process.

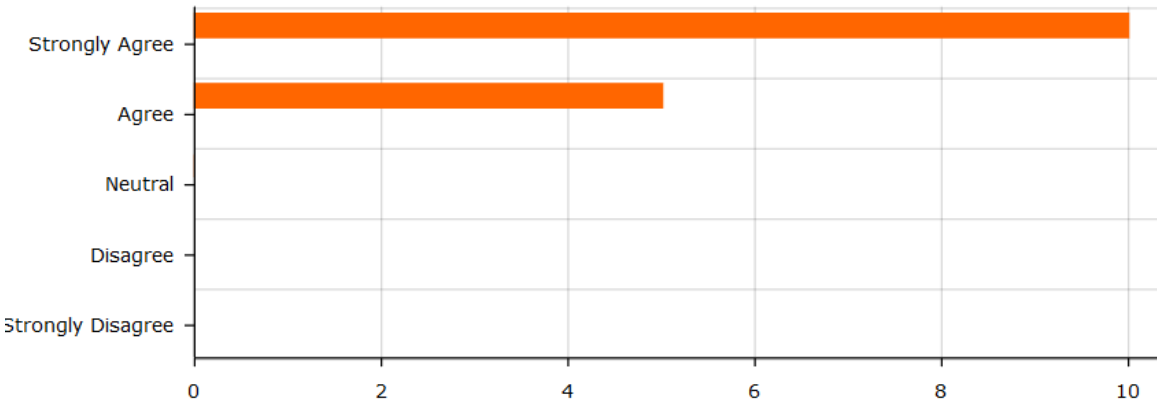
The next question was asked to find out if teachers felt as though a service like this would have help them act quicker to problems. Please find the results of this question below.



73.33% of teachers felt this to be true. This emphasizes one of the systems indirect benefits to the student as it provides support to them faster then current means of support provided.

.

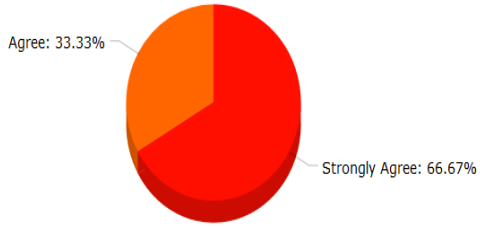
The third last question was created to determine if teachers felt that they would use an educational service like this regularly. Please find the answers to this question below:



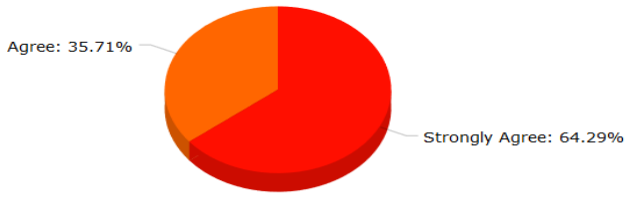
This shows us that most teachers do feel as though they would use a service like this frequently which is another positive response collected from the teacher's answers.

The final two questions of the teacher survey can be found below, for both questions teachers either agreed or strongly agreed with both the system being of benefit to their students and that a service like this would boost clubs and extracurricular actives within the school.

Question 9:



Question 10:



II.I.VII Teacher questionnaires – Conclusion.

This teacher questionnaires were a major success regarding the potential benefits of the systems on both the teachers work, how teachers felt to be a benefit for their students, and how teachers felt they would in fact like to strengthen the communication barriers between them and their respected student's parents.

In conjunction to the above, this survey showed us that most teachers were spending more than ten hours on social media per week which increases the probability of them using a service like this, which coincided with the fact the most teachers felt as though they would use this site. One of the major things to note about this feedback is that teachers felt not only would a service like this benefit their decision-making process, but it would also speed it up. This means that the teachers questions confirmed that this site will be of benefit to both the teachers and students based on decisions being made quicker which a direct prevention or problem escalation.

Another major thing to be noted from this study comes down to the fact all teachers agreed with the following facts:

1- This system will be of benefit to the teacher's students.
2- A system like this would boost clubs and extracurricular activities in the school.

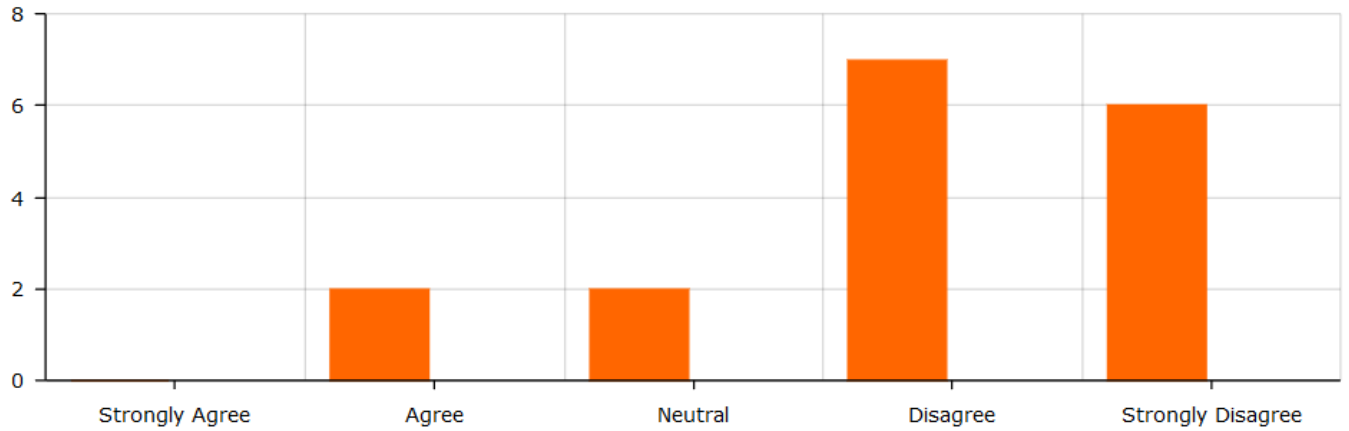
The fact that teachers felt so strongly about the two was a clear indication that they all felt this system will not only benefit the student, but also help to increase attendance within clubs and societies in the school which is a resounding positive for the systems development.

II.I.VX Parent Questionnaire

Statement	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
General Statements					
You feel as though you are constantly updated about your children's education by the school they attend.					
You spend over 10hr on social media per week.					
If you are updated about your children's academic performance more regularly you would be more likely to take corrective action sooner should there be a problem.					
Your children's version of how they're getting on in school is not always the same as the reports they receive at the end of the school year/teachers opinion.					
Being able to directly message your children's teachers would be of benefit to you.					
You can see yourself using a system like this regularly.					
You would like to view your children's academic performance on a weekly basis.					
You stay up to date with school news.					
The school should pay for a service like this.					
You would prefer your children to spend time on a service like this in comparison to other forms of social media.					
Functionality - Importance of the following being included;					
Messaging					
Notifications					
Friends Section					
Posts					
Viewing of grades					
Mobile phone responsive					
Comments					
Newsfeed					
Profiles					
Liking Content					
Simple Navigation					

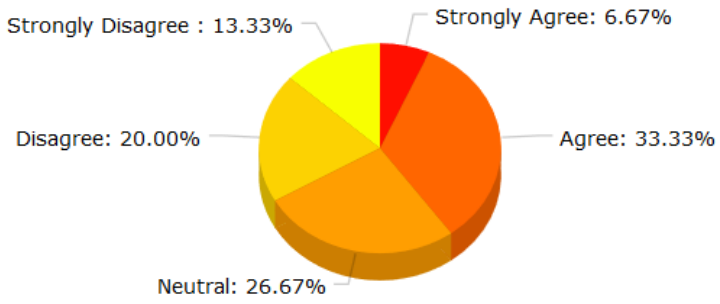
II.I.VIX Parent Questionnaires – Analyzation of Results.

The first question in the parent survey was conducted to determine if parents felt as though they were being constantly updated by the school their child attends/attended.



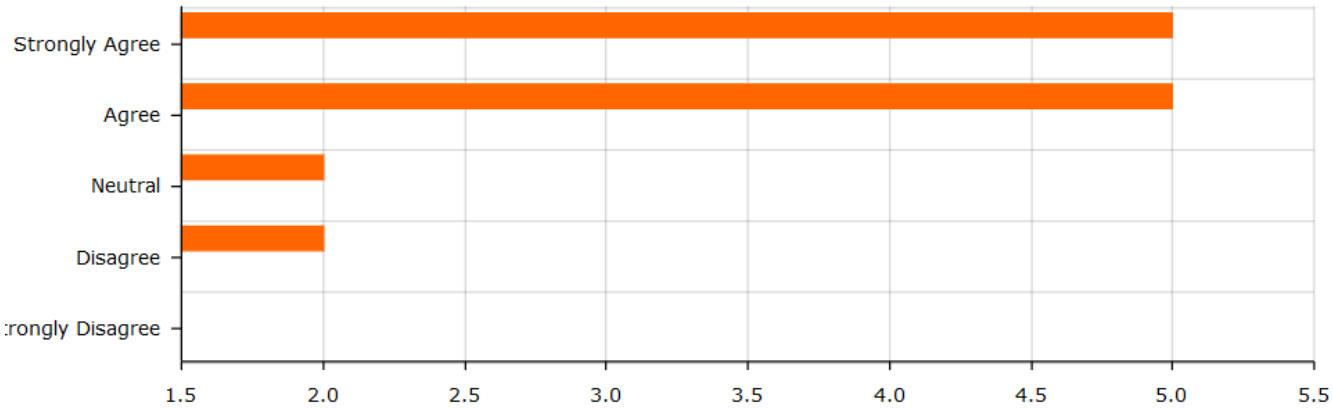
The answers to the first question are displayed above. This indicates that most parents either disagreed or strongly disagreed with this statement. Please note that the fact some of the participants children were out of education for a varying number of years may have influenced the results of this question.

The next question in the parent questionnaire was created to get more background information to see if parents were spending over 10 hours on social media per week. This question was created to determine the probability of them using another form of social media.



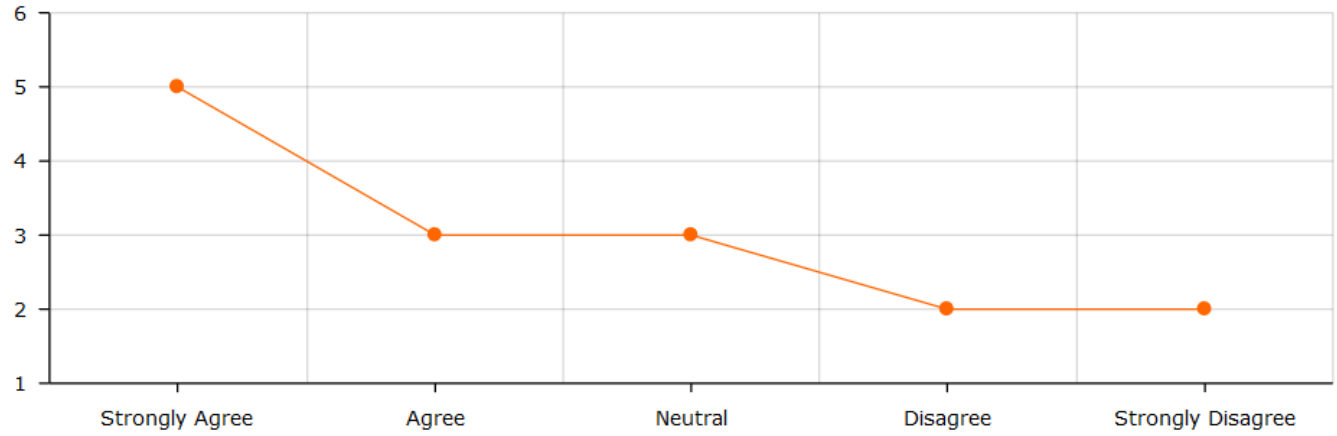
From this it can be determined that parents were using social media less frequently then other subject groups however 41% still did agree to this statement which indicates that a service like this may become a part of parent's social media weekly schedules.

Please find the answers to the third question asked below:

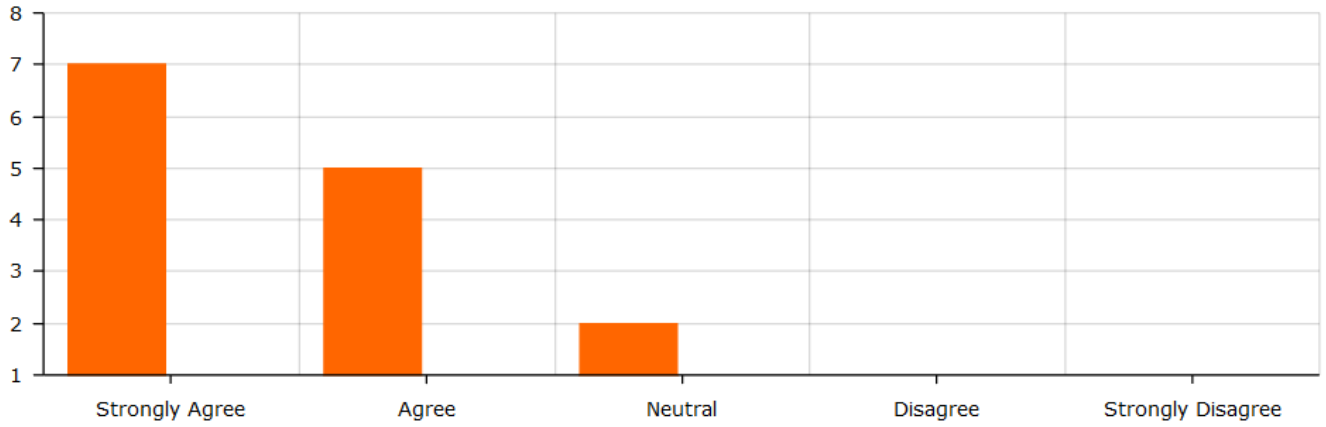


From this it can be determined that most parents felt as though a service like this would benefit their decision-making processes and allow them to make informed decisions faster. This highlights another indirect benefit the system has on the student.

The next question in this survey was designed to see if parents thought that their child and their child's school's version of their academic version were usually consistent. This results of this proved to be evenly spread, however this still highlight the fact tat in 40% of cases versions of events were different between the child and school, which shows major room for improvement.

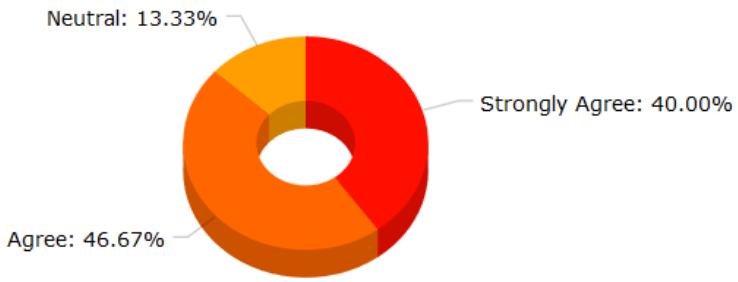


The fifth question in the parent survey was asked to determine if parents felt the being able to directly messages their children's parents would be of benefit to them. Please find the results of this questions below.



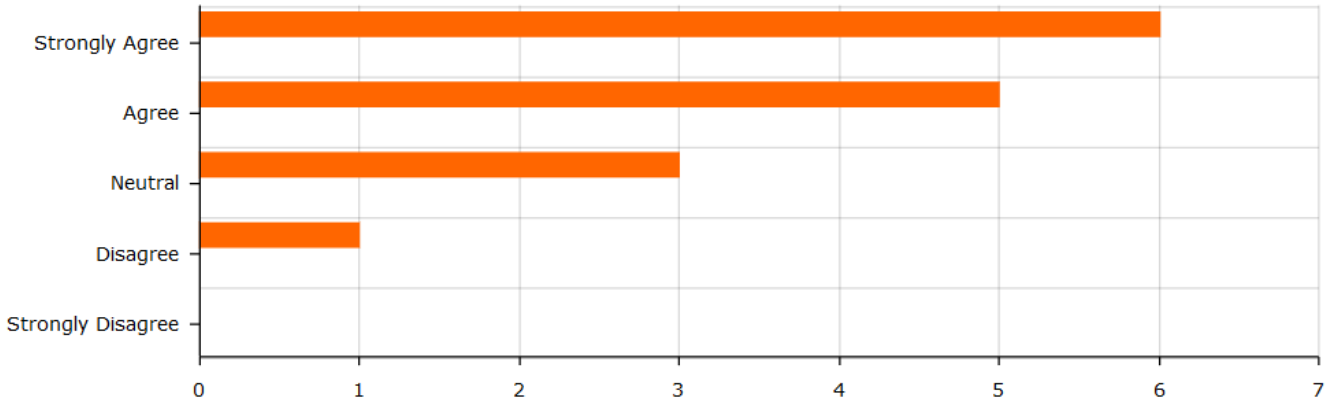
The above graphs emphasize the fact that almost every parent fell as though being able to directly message their children's teachers would be of benefit to them. This emphasizes one of the many ways this system will be a benefit to the parents using it.

The next question in the parent questionnaire was asked to see if parents felt that they would use a system like this regularly.



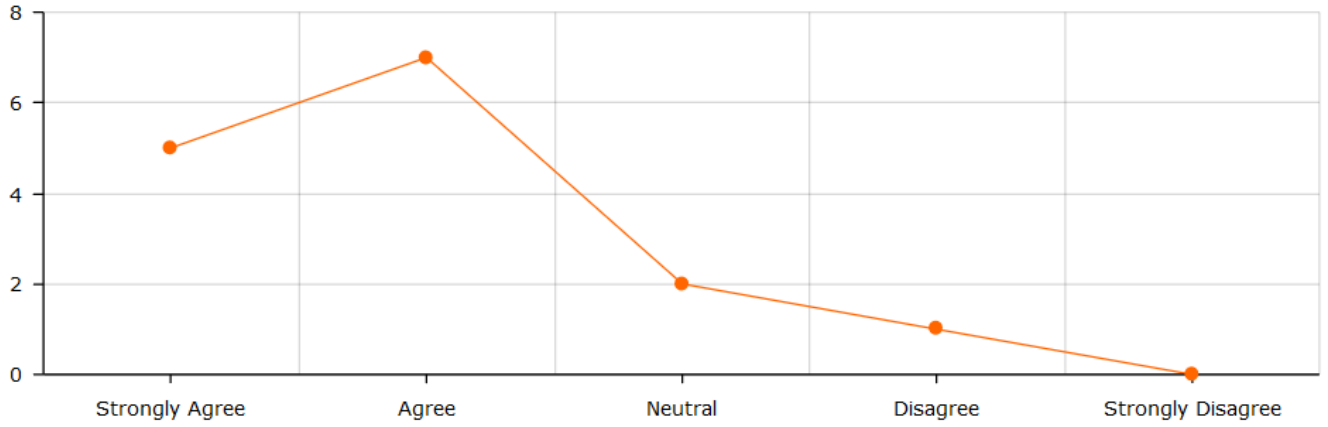
This provided promising results with the clear majority of parent users felling as though they would use this service. Which is a promising indicator for the systems development.

The next question was asked to see if parents felt as though they would like to be updated about their children's performance on a weekly basis.

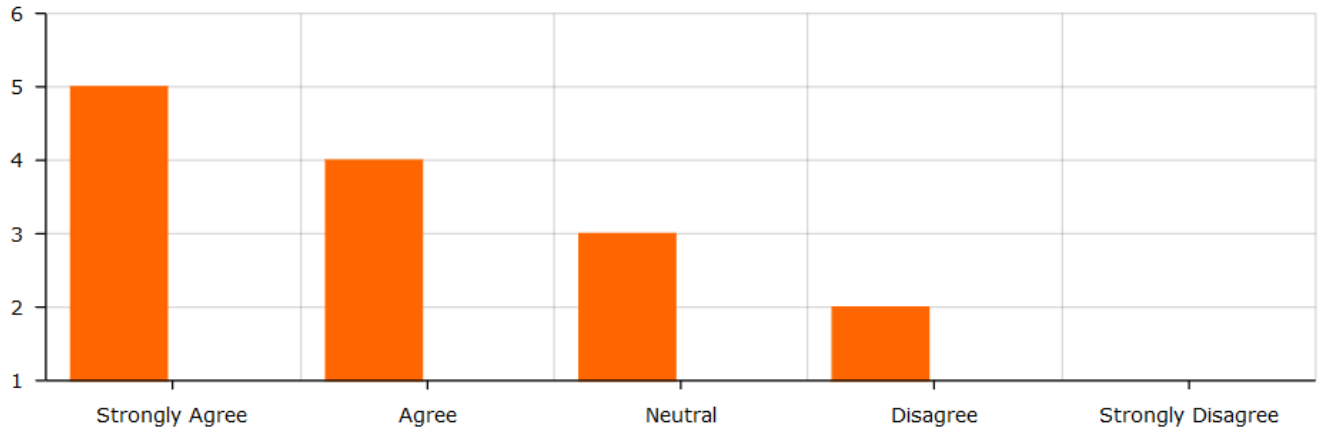


From this it was determined that nearly all parents agreed with this statement justifying the proposed benefits of the system to them.

The next question was asked to see if parents were staying up to date with the school's news. This was to be deemed attributive to the percentage chance of each parent staying up to date with their students results, as the process of doing both the above tasks are similar, however due the fact that parents are more interested in their child it is expected for the following results to be a smaller percentage of the actual likelihood of them staying up to date with the system.

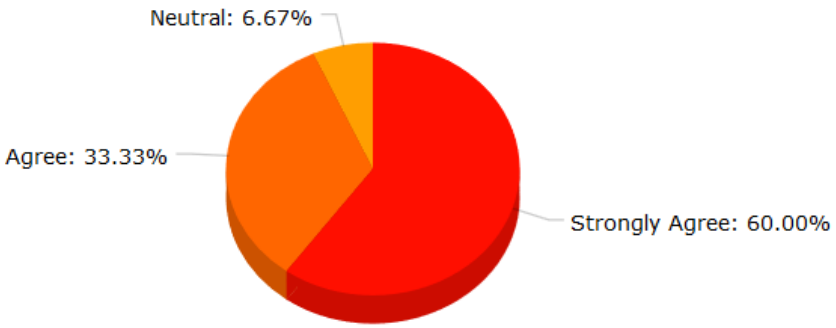


The second last question in the questionnaire was created to see if parents felt that the school just pay for a service like this. From this we could tell how strongly they felt a service like this was needed. Please find the answers to this question below:



9/15 parents felt a system like this should be payed for by the school which shows how strongly they felt about a system like this being needed.

The final question in the survey was created to see if parents would prefer for the children to spend time on a service like this opposed to other forms of social media. Please find the answers obtained below:



From this it can be determined that nearly every parent would prefer for their children to spend time on this opposed to other forms of social media, a clear positive regarding the web application.

II.I.XII Parent Questionnaires – Conclusion.

The parent's questionnaire identified several things of high importance regarding the web application. To being it identified the fact that most parents felt that they were not being updated constantly enough regarding their children's education. This is a problem which needed to be addressed and this is what the Skewley system sets out to achieve.

Another key factor identified by this questionnaire, like what was received by the teacher questionnaire came down to the fact that parents felt as though a system like this would benefit their decision-making process. One more of the numerous indirect benefits this system provides to the student.

Another major figure gotten from this questionnaire is regarding the fact that 40% of parents felt their child's version of how school was going was different to the version of said child's teachers. This shows a fault in communication the system also sets out to improve. In addition to showing these benefits this questionnaire also demonstrated how parents felt both being able to directly messages their child's parents in conjunction with being updated on a weekly basis to be another system benefit for parents using the web application.

The fact that most parents felt like a service like this should be paid for the school, this shows that not only would they use it, but they feel as though schools should be using it which is another major positive for the systems development.

II.I.XIII Functional Components – Importance feedback.

All participants in the study were asked to rate some of the systems most popular functional components. The results from the below indicates that users felt all the systems core components were in fact needed. Please note these are not all the components the system offers to see the full extent of each of the systems functionalities please see the functional requirement section of this report.

Components:	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
Messaging	35	5			
Notifications	32	8			
Friends Section	31	7	2		
Posts	39	1			
Viewing of grades	34	6			
Mobile phone responsive	40				
Comments	38	2			
Newsfeed	40				
Profiles	31	9			
Liking Content	29	2	5	4	
Simple Navigation	40				

From this it can be concluded the potential users felt that each of the components rated were important for the systems development. The fact that 100% percent of participants strongly felt that simple navigation were needed suggests that the usability of the system is evidentially important. All other components bar liking scored over 75% for users strongly agreeing with their importance which is another promising indication for the system.

The fact that users felt liking to be the least important issue is an interesting factor to these surveys. Perhaps removing this feature could be of benefit to students as it would erase their fears regarding not being deemed popular should their posts not receive enough likes which would encourage them to post content, develop relationships and learn more. However due to the fact 29 individuals strongly felt that this was an important feature cannot be overlooked.

II.I.I SYSTEM FUNTIONAL REQUIREMENTS

II.I.I.I ADMINSTRATION PANEL

User management/ administration section.

II.I.I.II DESCRIPTION

The idea to include an administration panel came from both the concern of system security combined with the amount of database tables/user details needed to be maintained/updated within the system itself. The administrators account is also situated within the administration section

A quick and efficient way for administrators to activate/deactivate accounts, combined with creating, reading, updating, deleting(CRUD) schools, teachers, parents, students, and classes within the system, to find any information regarding users in minimal time, and to maintain and filter all system data. Designed for school principals.

II.I.I.III ADMINISTRATION PANEL FUNTIONAL REQUIREMENTS

1. **Log In–** The administration panel must allow administrators to log into the system.
2. **User Type–** The user must be an administrator.
3. **Profile–** Admin profiles will be created for administrators which will be codependent from the rest of the system.
4. **Search –** Admins will be able to invoke a search method within the system to find any information on any Users/Classes.
5. **Activate Accounts-**The admin must be able to easily activate accounts.
6. **Deactivate Accounts-**The admin must be able to easily deactivate accounts.
7. **Create/Read/Update/Delete Classes-**The admin must be able to add/read/update/delete all classes and their details in the system.
8. **Create/Read/Update/Delete Parents -**The admin must be able to add/read/update/delete all parents and their account details in the system.
9. **Create/Read/Update/Delete Teachers-**The admin must be able to add/read/update/ delete all teachers and their account details in the system.
10. **Create/Read/Update/Delete Students-** The admin must be able to add/read/update/ delete all students and their account details in the system.
11. **Create/Read/Update/Delete Schools-**The admin must be able to add schools to the system. In the instance, the admin oversees more than One School.
12. **Manage all Classes/Parents/Teachers/Students/Schools-** The administrator must be able to manage on sub systems classes/users and all user information.

13. **Log Out-** Administrators must be able to easily log out of the system.

II.I.II.I STUDENT PANEL

The student panel is the section of the system responsible for input/recording student grades attendance and behavior.

II.I.II.II DESCRIPTION

The student section is a sub section of the social media section. It gets its name from the fact this is the area of the system where Students and Parents log in to receive information regarding their/the student's grades/attendance and behavior in combination with the fact that this is the area of the system where teachers are expected to input grades, attendance and behavior data regarding each of their students.

Although all users are going to use this part of the system they are all going to experience it differently. This differs in the sense each Teacher in a given school will access the system via logging in through their unique log in and then navigating to the student section via their social media panel. Once the Teacher has navigated to the student section they will be promoted to select their subject based on the number of subjects they teach. The level of the subject they teach is the following data field expecting data, this is broken into three distinct categories', higher, ordinary and foundation level. In addition, the teacher must also select the year group of the subject and subject level they have selected, this ranges from 1st to 6th year.

Students and Parents experience the system in a near identical manner. Both users login via their unique login which leads them to their unique home screens. The main user difference between these two users comes down to the fact that once they've logged in all parent users are subject to limited functionality, as they do not have the ability to view student posts/post statuses etc. From the home screen both users can navigate to their relative student sections in the same manner, via navigating to the student section via their side navigation bars. Once they navigate
to the student section parents are lead to a page which allows them to pick between children should there be more than one attending the school. This differs to students as they are lead directly to the grades page their parents are led to after making their child selection. On this grade page both users are given a list of classes they take, from here they can then navigate into each class dashboard and check their grades, attendance and behavior for a given term.

II.I.I.III STUDENT PANEL FUNTIONAL REQUIRMENTS

1. Logged In– Users of the Student panel section must already be logged into the Social panel of the system.

2. User Type– The user must be either a teacher, parent, or student. Only teachers can input data.

3. Filter Classes– The system must adequately allow teachers to filter between all classes they teach, and input data based on their selection.

4. Filter Level– The system must adequately allow teachers to filter between the level of the class they have selected. The level of the class is based on the current leaving cert options (Higher, Ordinary, Foundation).

5. Filter Year– The system must allow teachers to select the year of the class they have inputted data for. This allows derived from today's education system ranges from 1st to 6th year.

6. Select student– The system must allow teachers to select all students within the class they have selected via system filtering.

7. Data Input – The system must allow teachers to input data based on all student's grades, attendance, and behavior.

8. Retrieve data– Parents and Students must be able to retrieve data for all their/ their child's classes.

9. Select class– Parents and Students must be able to select all classes the student attends.

10. Select performance– Parents and Students must be able to select their child's performance for each class selected.

11. Select grades – Parents and Students must be able to select their child's grades for each class.

12. Select attendance – Parents and Students must be able to select their child's attendance for each class.

13. Select behavior– Parents and Students must be able to select their child's behavior for each class.

14. Averages– Parents and Students must be able to view average grades for each class attended.

15. Data Visualization/Display High Charts- Parents and Students must be able to view their performance regarding their grades, attendance and behavior via the use of interactive Line, Pie, and Bar charts.

16. Download charts- This section must use high charts in built file downloader, allowing the charts to be downloaded in PNG, JPEG, PDF, SVG file formats.

17. Storage- This section must store all student's performance regarding their grades, attendance and behavior results for their school life

II.I.III.I SOCIAL MEDIA PANEL

Social media panel.

II.I.III.II DESCRIPTION

The social media section is the main section of the SKEWLEY system. Here users share functionalities; however, some functions are user specific.

When users log in successfully they are lead to the home page. From here they have the option to access all social media pages and function. From here users are given the option between to toggle the side navigation which links to the news, profile/settings, student, friends, and logout views. The side navigation and header navigation panels and fixed for all the social media sections pages.

The header navigation bar provides users with the options to view their profile/sign out, to view all messages received from friends via the instant messenger pane/ send messages instantly and to view all system notifications. The header bar also contains the functionality to navigate to the friend's section of the system, for the user to view their timeline and provides a search functionality. When users scroll to the bottom of the timeline their friends list is displayed.

From the homepage/news section users can also view all recent post from all friends/groups/clubs they are friends with or are a part of. They have the option to like and comment on these posts, in conjunction with being able to post their own statuses, where they also upload images from their own personal device. This allows all users to navigate the site effortlessly alongside socialize freely.

From the users profile/account setting pages they are provided with the option to upload/change their profile picture. They account details including their email and their location is also displayed. From users are also provided with the option to edit all their account information. This includes changing username/password. Users are also provided with the ability to remove their accounts.

Another of the options users are given is to access the student section of the system. Please view details above for full description. In conjunction, users are also provided with the option to navigate to the friend's page where they can view all their friends and groups they are following. The final option user is given is the logout function.

II.I.III.III SOCIAL MEDIA PANEL FUNTIONAL REQUIREMENTS

1. **Log In–** Parents, Teachers and Students must log into the system via separate log ins.
2. **Account Activation –** Parents, Teachers and Students must all have their accounts activated by the administrator.
3. **Home Screen–** Once the user has logged into the system they will be granted access to their respected users home screen.

4. Sidebar Navigation- From the side navigation section users will be able to access on social media sections of the system.

4. Toggle Sidebar- Users will be given the option to hide/show the side navigation bar.

5. Header Navigation- From the header navigation of the systems users will be able to quickly access most of the social media functions with the additional functionality of the search method and instant messenger.

6. Check messages- From the header navigation users will be able to check all messages they have received from all friends/groups alongside the date of receival. Users will also be notified should they receive a new message

7. Send messages- From the header navigation users must be able to send messages to all their friends/groups/teachers as well as view the message history of all their conversations.

8. View notifications- Users must be able to easily access all notifications via the header navigation.

9. Like Notification- Within this notification panel users must be notified any time a user likes any of the content a user has posted (comments/statuses).

10. Friend Request Notification- Within this notification panel users must be notified any time they receive a friend request.

11. Comment Notification- Within this notification panel users must be notified should another user comment under any of his/her statuses. In conjunction, the user should be notified should another user post on a status the user has commented on.

12. Profile Header Link- Within the header navigation bar users must be able to view a profile overview panel.

13. Profile Overview panel- This panel provides information regarding user type and user picture.

14. Profile link- A link to the user's profile is included within the profile overview panel.

15. Sign Out- A user must be provided with the option to sign out via the profile over view panel of the system.

16. Friends - An option to navigate to the friend's section must be provided via the header navigation of the system.

17. Timeline- Users must be given the option to go back to the start of their timeline via the header navigation.

18. Search- Users must be able to search for all their friends/groups via the header navigation.

19. News- The news/home screen section of the system allows users to view all their groups and friend's activity on the system.

20. News Post/Status- Users can post statues from the news/home section.

21. Upload images- Users can access their devices images and upload them with their status.

22. Restrictions- Maximum image size allowed is 500kb.

23. Like/Post Comments- Users are given the option to comment/like under all news items they have access to.

24. Like/comment under posts- Users are given the option to like/comment under all posts in their news section.

25. Scroll ending – When users scroll to the bottom of their timeline he/she are given a list if all theirs friends/ encouraging communication.

26. Profile/Account Settings section– From the side navigation/ header profile overview panel users can access their profile/Account settings section of the system.

27. Profile Picture Upload- All users can upload their own custom profile picture.

28. About User- User email and location are to be accessed from via the profile section.

29. Edit Details- Users must be able to edit/update all account details via their profile section including user name and password.

30. Remove Account- Users must be provided with the option to remove their accounts via the profile section.

31. Student Section- Users must be provided with the option to navigate to the student management section of the system via the side navigation bar.

32. Friend Page- Users must be able to access their friend's page via the side navigation bar of the system.

33. Friend Zone- A list of all user's friends and groups or clubs they follow will be displayed here.

34. Delete friends- Users must be given the option to delete friends via the friend's section of the system

35. Friend Requests- Users must be able to view all friend requests via the friend section of the system.

36. Logout – Users must be able to log out via the social media section of the system.

37. Device/Browser Responsive– The system must be responsive across all browsers and devices.

II.II DATA REQUIREMENTS

1. Administrator- All data the administrator manually inputs regarding all User types and classes must be stored accordingly.

2. Teacher- After system filtration based on class, year and level teachers must effectively input data on their classes/students via the student section of the system.

3. Activities- All activities regarding messages, notifications, posts, statuses, and likes must be connected to the database effectively.

4. MYSQL – All data required for all views within the system must be effectively queried in said views corresponding controller properly.

5. Sessions – Some user data will be stored client side via the use of sessions.

II.II.I DATABASE LOGIC

The following list describes how all user's types are connected within the database and how student data was specifically tailored to all individual students and their parents.

1. Create Users- All users accounts are created and activated in the admin panel.

• **2. User types-** All users were categorized to the Teacher (13), Student (20), and Parent (20) Tables in the admin panel.

• **3. Linking Students to Parents** - All Students were interlinked to their parents ID.

• **4. Teachers are linked to Classes-** All Teachers are linked to the class/classes they teach.

• **5. Linking Students to Classes -** Creation of student classes table (Interlinking all user types).

• **6. Assigned Students to Classes-** All (20) students were assigned to 7 classes (Leaving Certificate) these classes are linked to all Teachers (13). Classes: Irish. English, Math's = compulsory. Languages are optional; German, Spanish, French. (20 % 3) The additional 7 subjects were broken into 4 groups of 3 subjects. (20 % 4) = 5, meaning streams were created for groups of 5 students, allowing all students to partake in different classes.

• **7. Posting student Performance-** After all tables were created, Teachers are now able to select a class they are assigned to. All students assigned to this class will now appear. Teachers can now input data based on Grades, Attendance and Behavior and Date. This is done via in the social media section of the application.

• **8. Getting student Performance-** Now all performance metrics are linked to Classes which are interlinked to the Student which is interlinked to Parents, allowing Students and parents to view their/ their children's grades, attendance and behavior for any class the student attends.

II.III USER REQUIREMENTS

This section will look what is required of all users to use the system.

1. Device- To use SKEWLEY, users must have access to either a smartphone, tablet, laptop, or computer.

2. Internet Access- All users must have access to the internet to use SKEWLEY.

3. Users – All user types must be linked to their system by either working, attending of have a child attending the respected school.

II.IV ENVIORMENTAL REQUIREMENTS

The following section will discuss all environmental requirements needed during the development of the SKEWLEY system.

1. Laptop/PC- A Laptop/pc is needed to develop/execute code.

2. IDE- An IDE with apache pre-installed was needed to run the code (cloud9).

3. Internet- Internet was needed to track the systems development in addition to be using used for various resources used by the project.

II.V USABILITY REQUIREMENTS

1. Understandable. - All components of the web application must be easy to understand. This is to ensure there is no confusion whilst navigating. In conjunction, ensuring everything is in a proficient manner will also allow users to navigate through the website quicker.

2. Learnability. - Another of the main aims of the SKEWLEY system is to ensure users remember the layout when returning to the site. This will be achieved my making all functions available via the homepage.

3. Operable. - All actions included should be consistent and reliable and if any errors do occur, error messages should be sent accordingly.

4. Attractiveness. – The web applications must appear to a very high professional standard, this will be achieved by implementing usability testing

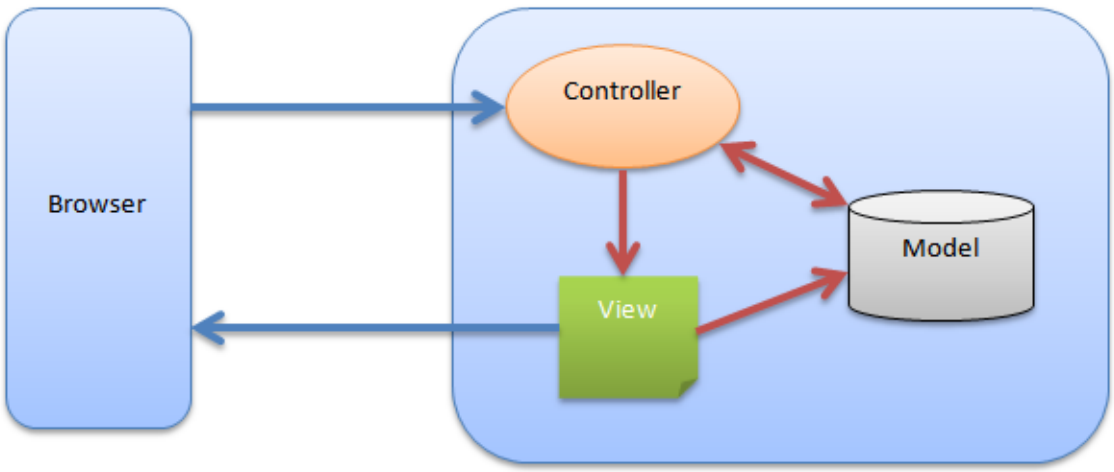
II.VI DESIGN AND ARCHITECTURE

To achieve seamless integration for all SKEWLEYS components a MVC framework was used. The main benefit of using Models, Views and Controllers was ensuring that code was stored efficiently. This framework provides a means that ensures server and client-side code is always separated to ensure code is categorized properly. This modularizes your code allowing it to be built upon in the future and is a means of quality assurance, and an example of good coding practice.

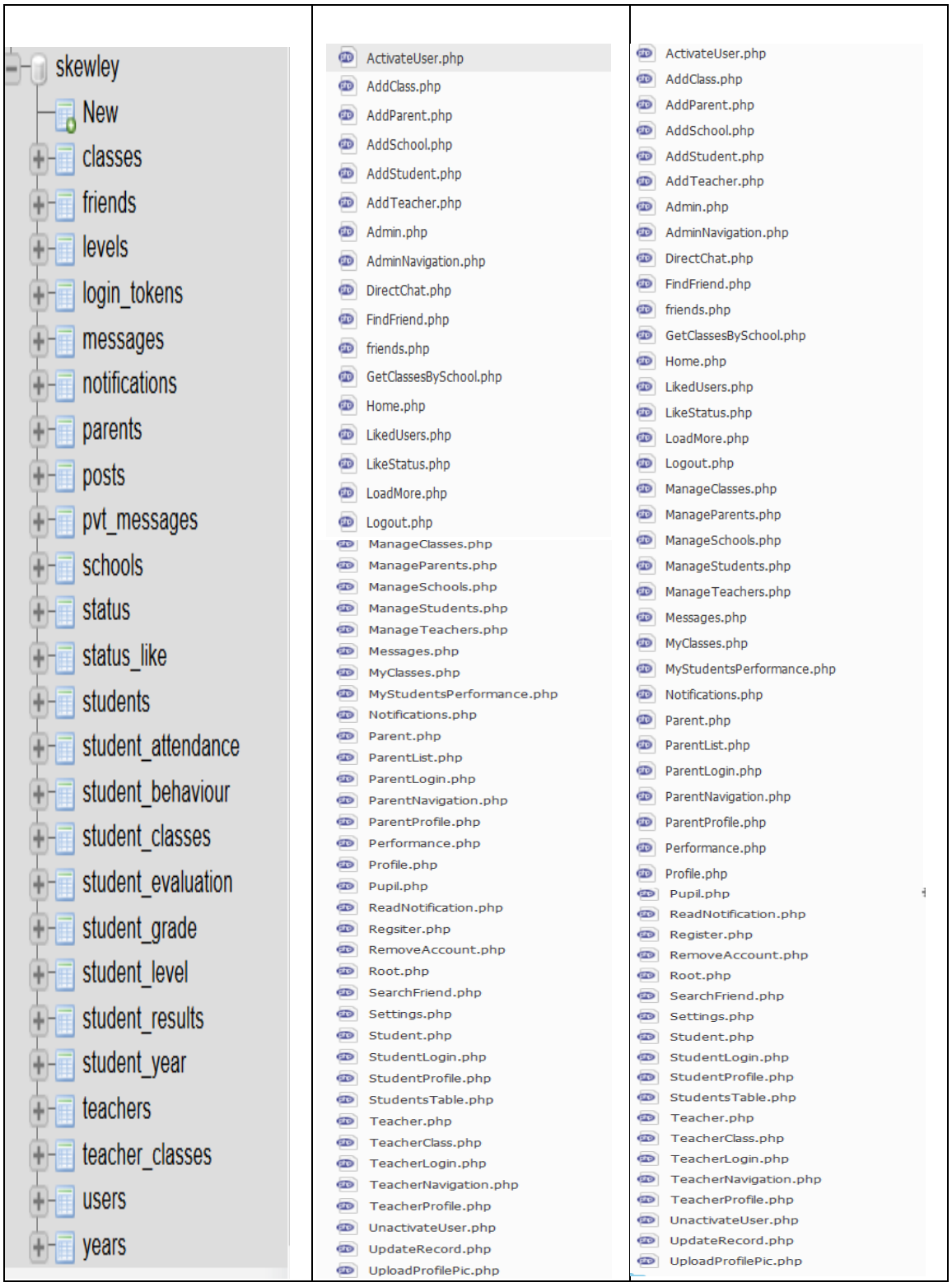
1. Model. - Underlying logical data structure – MySQL Database. (user never directly interacts with). This MySQL databased was created and housed on phpMyAdmin which allowed all controllers noted in the next section.

2. View. – What the user sees – GUI (only component users directly interact with). These were created using html, css, jquery, javascript, jquery and bootstrap. To provide a rich combination of front end interfaces for all users.

3. Controller. - Intermediary between the Models and Views. Allowing the Two to communicate effectively whilst ensuring they are separated. Exclusding the home controller which brought together numerous functions, controller in the Skewley application usually served a direct purpose for data retrieval. I.e the find friends section would specifically query the database to search for a username etc. Each controller varied in purpose and all of these are listed in the pages below and explained the implementation section in further detail.

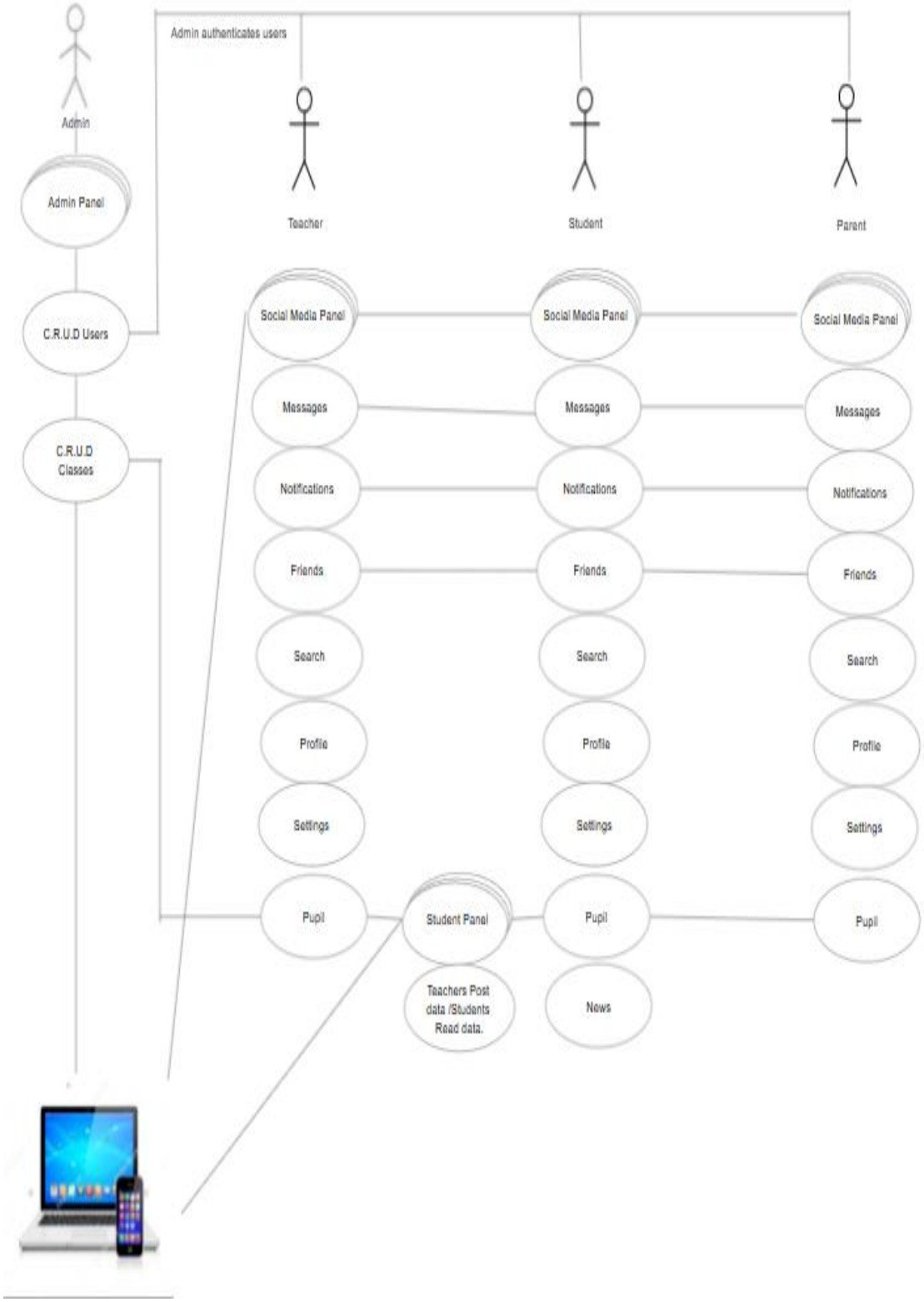


II.VII Models. Views. Controllers.



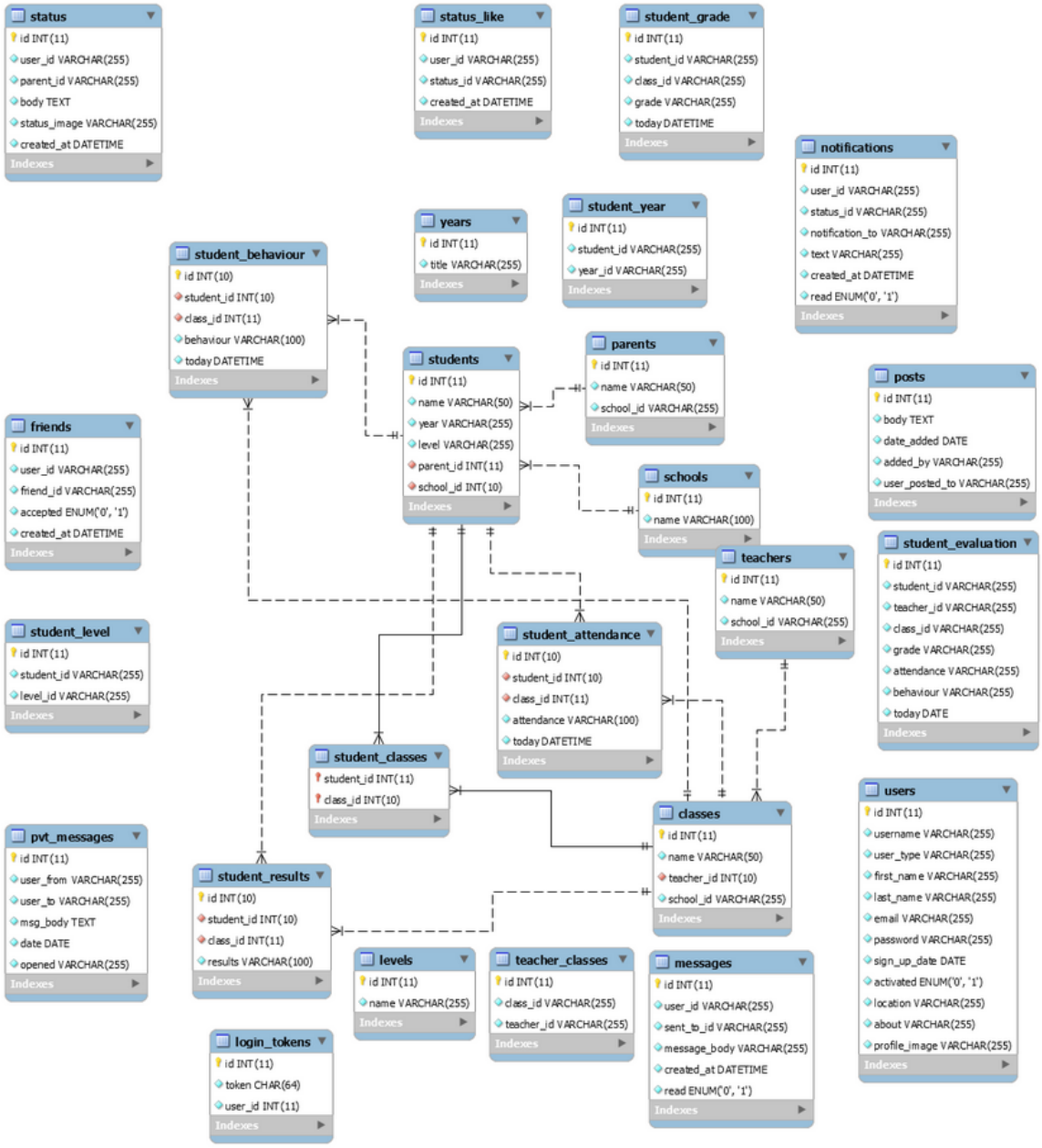
II.VII Use Case Diagram.

The following diagram represents how each user it to use the system, what they can use the system on and the functionality provided to each individual user group.



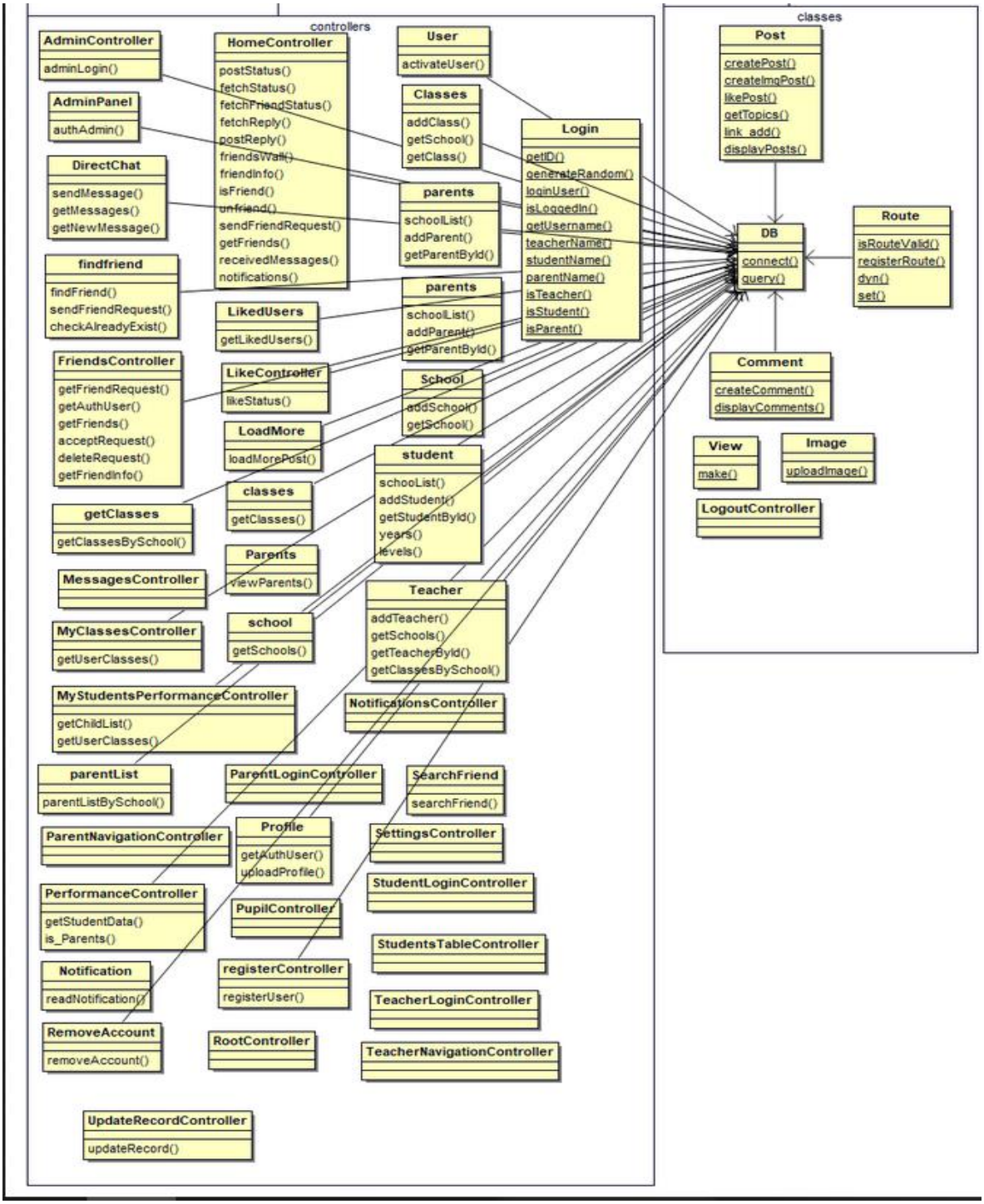
II.IX DATA MODEL DIAGRAM

The data model diagram below illustrates a graphical representation of the database schema used for the Skewley system and shows the logical link between the student table and how it connects to parents and teachers through the creation of intermediary tables.



II.X Class Diagram

This diagram illustrates how each controller connects to the databases and lists each function within each controller. All these functions interact with the database differently.



IMPLEMENTATION

II.XI.I MVC implementation.

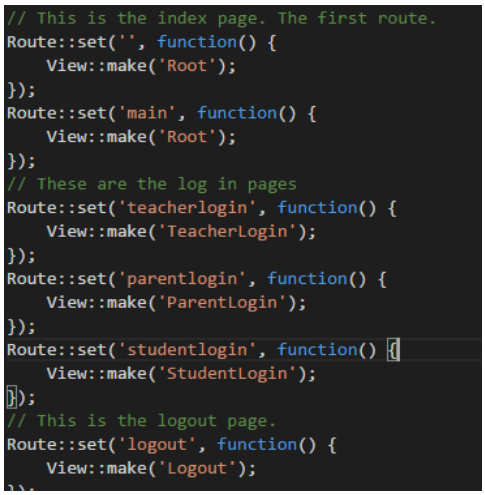
The stages, order, and methodologies used while implementing the Skewley systems will now be discussed. To begin, After the midpoint presentation an alternative means of creating the codebase to ensure it was as efficient as possible was identified. This was achieved by using an MVC framework. I decided to create my own framework as opposed to using popular alternatives like Laravel and cake PHP to broader my understanding of the inner mechanics as to how MVCs are put together. Although very difficult to get my head around in the beginning, the benefits of the systems lightweight structure made the decision to do so completely worthwhile.

```php
<?php
ini_set('display_errors',0);
/*
 * The first thing we do is autoload all of our classes. We're using the older
 * __autoload function as opposed to spl_autoload_register because all of our
 * classes are located in the /includes/classes/ directory.
 *
 */
require_once ('./includes/_Globals.php');
/*
 * By including routes/Routes.php we get access to the $Routes
 * array containing all of the valid routes for our app.
 */
require_once ('./includes/Routes.php');
function __autoload($class_name) {
    require_once './includes/classes/' . $class_name . '.php';
}

// We create a new instance of the 'Skewley' object and execute the run method.
$skewley = new Skewley();
$skewley -> run();

 ?>
```

The above code snippet shows the systems index.php file. Here all required classes for Skewley's creation are included. This includes the Globals, Routes and Classes PHP files.

```php
<?php
/*
 * The array 'Routes' stores all of the valid routes, you can add
 * new routes by editing the file 'routes/Routes.php'.
 */
$Routes = array();
/*
 * Define the BASEDIR. This is the directory that the site is stored in
 */
define('BASEDIR', '/Schoooly/');
define('base_url', '/Schoooly');
```

The above snippet shows the Globals.php file. Here the Routes array was declared. In conjunction- both constants were also instantiated regarding the systems Base directory and base URL. This file was essential in allowing the system to know all permitted URLS based on the files storage and routes created.



```php
// This is the index page. The first route.
Route::set('', function() {
    View::make('Root');
});
Route::set('main', function() {
    View::make('Root');
});
// These are the log in pages
Route::set('teacherlogin', function() {
    View::make('TeacherLogin');
});
Route::set('parentlogin', function() {
    View::make('ParentLogin');
});
Route::set('studentlogin', function() {
    View::make('StudentLogin');
});
// This is the logout page.
Route::set('logout', function() {
    View::make('Logout');
```

```php
    // Register the route and run the closure using __invoke().
    public static function set($route, $closure) {
        if ($_SERVER['REQUEST_URI'] == BASEDIR . $route) {
            self::registerRoute($route);
            $closure -> __invoke();
        } else if (explode('?', $_SERVER['REQUEST_URI'])[0] == BASEDIR . $route) {
            self::registerRoute($route);
            $closure -> __invoke();
        } else if ($_GET['url'] == explode('/', $route)[0]) {
            self::registerRoute(self::dyn($route));
            $closure -> __invoke();
        }
    }

}
```

The upper left snippet was taken from the Routes.php. Within this file all the systems views/accessible paths were created. In this file, the static function set(Route.php) is called (upper right image). In the set met the register route function is called which is also located in the Route.php file. Within here the route is registered, and placed inside the Routes array. After this has occurred the routes controllers and classes can now be made.

```php
<?php
class View {
    /*
     * If the route is valid create the view and the view controller.
     * If the route is invalid do nothing and if something goes wrong
     * checking the route return 0;
     */
    public static function make($view) {
        if (Route::isRouteValid()) {
            // Create the view and the view controller.
            require_once ('./includes/controllers/' . $view . '.php');
            require_once ('./includes/views/' . $view . '.php');
            return 1;
        }
    }

}
```

After the route has been registered, the make function within the views.php class is called. This then goes about calling the isRouteValid function (ensures another root with the same name does not exist). Furthermore, creating a controller and view file for each valid route. If creation is successful/routes and controllers were created- 1 is returned, if not it returns 0.

II.XII Functional Component Implementations

II.XI.I Add User to the system

```php
<?php
class student{
    public function schooList(){
        $school = DB::query("SELECT * FROM schools ORDER BY id");
        return $school;
    }

    public function addStudent($data){
            $user_name = $data['user_name'];
            $first_name = $data['first_name'];
            $last_name = $data['last_name'];
            $email = $data['email'];
            if(!empty($data['pwd'])){
                $pwd = password_hash($data['pwd'], PASSWORD_BCRYPT);
            }
            else{
                $pwd = '';
            }
            $school = $data['school'];
            $classes = $data['classes'];
            $parent = $data['parent'];
            $sign_up_date = date("Y-m-d");
            $edit_id = $data['edit_id'];
            $year = $data['year'];
            $level = $data['level'];
            if(!empty($edit_id )){
                if(empty($pwd)){
                    $user = DB::query("UPDATE users set first_name='".$first_name."',
                                last_name='".$last_name."', email='".$email."',
                                username='".$user_name."'
                                WHERE id='".$edit_id."'
                                ");
                }
                else{
                    $user = DB::query("UPDATE users set first_name='".$first_name."',
                                last_name='".$last_name."', email='".$email."',
                                username='".$user_name."',password='".$pwd."'
                                WHERE id='".$edit_id."'
                                ");
                }
                $student =  DB::query("UPDATE students set id='".$edit_id."',
                                name='".$user_name."', school_id='".$school."',
                                parent_id='".$parent."', year='".$year."',
                                level = '".$level."'
                                WHERE id='".$edit_id."'");
                $delete_previuos_class = DB::query("DELETE FROM student_classes WHERE
                                                student_id='".$edit_id."'");
                    foreach($classes as $key=>$value){
                     $student_classes = DB::query("INSERT INTO student_classes
```

The above code snippet depicts the add student controller, whic is like the add parent, teacher, classes, and School controllers. These share similar functions however they all access separate tables in the database.

The opening function schoolList returns a list of all schools. This is followed by the add student function which allows the admin to manually add student data. After this the admin is given the option to update existing data should fields not be empty, which also caters for accounts with/without password via the use of the above if statement. This final section of the above code snippet allows the admin to delete student classes.

II.XII.II Messages

The implementation of the messaging system used for the system will now be discussed. This was done via the creation of the directChat.php class which is responsible for the core functionality regarding the sending and receiving of messages on the system. The DirectChat class is broken into three separate functions:

1 – sendMessage.

```
class DirectChat{
    public function sendMessage($data){
        $friend_id = $data['friend_id'];
        $message = $data['message'];
        $user_id = $data['user_id'];
        $created_at = date('Y-m-d H:i:s');
        $send = DB::query("INSERT INTO messages (user_id,sent_to_id,message_body,created_at)
                           VALUES('".$user_id."', '".$friend_id."', '".$message."',
                                  '".$created_at."')");
        $last_id = DB::query("SELECT messages.id as message_id FROM messages
                              ORDER BY id desc LIMIT 1");
            return ['response'=>'success', 'last_id'=>$last_id[0]['message_id']];


    }
```

The above snippet depicts the function used to allow messages to be send on the system. Enlisting the data variable allowing for it to be passed multiple user defined variables to facilitate the posting of date into the messages table in the databases can be seen here.

The inclusion of the $friend_id and $user_id variables are critical here as they allow for the data structure to remain static for all users of the web application, however post different information to the database based on 1) the user of the system 2) their friend they have selected to send the message to, which is received by retrieving selected_friend information based on user input.

After all data is inputted correctly the $last_id variable is created to distinguish a list of messages in descending order. Once this has been carried out the response is set to success to record data regarding the message being replied to/data sent back/inserted. The message_id associated with this data is now added to last_id for it to be retrieved again by either user and to keep record of their conversation.

2 – getMessages.

```
public function getMessages($id, $sid){
    $read = DB::query("UPDATE messages set `read`='1'
                    WHERE user_id = '".$id."'
                    and sent_to_id = '".$sid."'
                      ");
    $message = DB::query("SELECT messages.user_id, messages.message_body,
                    messages.created_at, users.username,users.profile_image,
                    messages.id as message_id
                    FROM messages
                    INNER JOIN users ON messages.user_id = users.id
                    WHERE (messages.user_id = '".$id."'
                    and messages.sent_to_id = '".$sid."')
                    or (messages.user_id = '".$sid."'
                    and messages.sent_to_id = '".$id."')
                    ORDER BY messages.created_at asc
                    ");
    return $message;
}
```

The code snippet above is responsible for the receival of messages of all users on the system.

To begin, the read column of messages table of the database is updated to read by sending user and sent to ids of the associated message to the database.

Once the read column has been updated, all relevant messages data is retrieved via user and messages ids from the message table of the database. The Order by function is used to order message by date. The message variable is then returned which returns a list of all messages between users.

3– getNewMessages.

```
    public function getNewMessage($id, $sid, $lid){
        $read = DB::query("UPDATE messages set `read`='1'
                        WHERE user_id = '".$id."'
                        and sent_to_id = '".$sid."'
                          ");
        $message = DB::query("SELECT messages.user_id, messages.message_body,
                        messages.created_at, users.username,users.profile_image,
                        messages.id as message_id
                        FROM messages
                        INNER JOIN users ON messages.user_id = users.id
                        WHERE ((messages.user_id = '".$id."'
                        and messages.sent_to_id = '".$sid."')
                        or (messages.user_id = '".$sid."'
                        and messages.sent_to_id = '".$id."'))
                        and messages.id > $lid
                        ORDER BY messages.created_at asc
                        ");
         return $message;
    }
}
?>
```

The getNewMessages function is the same as the getMessages function bar it accepts the $lid variable which is definitive to all new messages on the system.

The functions mentioned prior are responsible for retrieving and posting messages to the database. The following receivedMessages function is housed in the home controller of the

system and has been created to provide a list of all messages received per user, accessible via the messaging section of the application.

```
public function receivedMessages($id){
    $message = DB::query("SELECT messages.user_id, messages.sent_to_id,
                          messages.message_body, messages.created_at,
                          messages.read,users.username,users.profile_image,
                          messages.id as message_id
                          FROM messages
                          INNER JOIN users ON messages.user_id = users.id
                          WHERE messages.sent_to_id = '".$id."'
                          ");
    return $message;
}
```

II.XII.III Notifications

The creation, population and updating of notifications and notification status on the system will now be explained.

The three instances in which the notification table is populate within the database are all like the following:

```
            else{
                $notification = DB::query("INSERT INTO notifications
                                          (user_id,status_id,notification_to,text,created_at)
                                          VALUES('".$user_id."','".$status_id."',
                                                 '".$status_original_user[0]['user_id']."',
                                                 '".$text."', '".$created_at."')");
                $like = DB::query("INSERT INTO status_like (status_id, user_id, created_at)
                                   VALUES('".$status_id."','".$user_id."','".$created_at."')");
                 $fetch_liked = DB::query("SELECT * FROM status_like
                                          WHERE status_id='".$status_id."' ");
                 $data = ['res'=>count($fetch_liked)];
                 return $data;
            }
    }
}
?>
```

The table is populated by a combination of user specific input and notifications details which are distinguished by comments, likes and friend requests. The above snippet shows the process of the notifications table being populated based on the liking of a given status, in addition with providing the capability of fetching all users who have liked a status. This is the common procedure used across all notification types.

```php
<?php
class Notification{
    public function readNotification($data){
        $notification_id = $data['notification_id'];
        $read = DB::query("UPDATE notifications set `read`='1'
                           WHERE id='".$notification_id."'");
        return 'success';
    }
}
?>
```

The above code snippet shows how the read column of the notification table is changed based on user input. This is how the notification count on the homepage is displayed.

II.XII.IV Student Performance

The studentPerformance controller is used to retrieve all data regarding students per class. This is achieved by passing both student_id and class id to the below function. This is how the system knows identifies 1) the student, 2) the students specific class.

```php
<?php
class PerformanceController {
    public function getStudentData($student_id, $class_id){
        $details = DB::query('SELECT students.name as student_name,classes.name as class_name,
                              student_evaluation.grade,student_evaluation.attendance,
                              student_evaluation.behaviour,teachers.name as teacher_name,years.title as year,
                              student_evaluation.today
                              FROM students
                              INNER JOIN  student_classes ON students.id = student_classes.student_id
                              INNER JOIN classes ON student_classes.class_id = classes.id
                              INNER JOIN student_evaluation ON classes.id  = student_evaluation.class_id
                              INNER JOIN teachers ON student_evaluation.teacher_id = teachers.id

                              INNER JOIN years ON students.year = years.id
                              WHERE students.id="'.$student_id.'"
                              and classes.id="'.$class_id.'"
                              and student_evaluation.student_id = "'.$student_id.'"
                              ORDER BY student_evaluation.today asc');
                return  $details;

    }


    public function is_Parents($pid, $sid){
    $parent = DB::query("SELECT * FROM students WHERE parent_id='".$pid."'
                        and id='".$sid."'");
                        return $parent;
   }
}

$userid = Login::isLoggedIn();
```

Once these arguments are provided, the database is queried to receive relevant student and class information of the student information passed. This data is then used to interlink student data with relevant class data, student evaluation (grades/attendance/behavior data) and teacher data. This enables data to be inputted about specific student's classes based on

the class Id associated to the teacher inputting the data. This is finally joined on student year to separate year groups whilst querying the database. Student evaluation data is stored in ascending order from today, this provides the feed to call charts in the system, allowing plots for the x and y axis (Date and Result(integer)).

II.XII.V Get student Classes

```php
<?php
class MyClassesController {
    public function getUserClasses(){
        $token = $_COOKIE['SNID'];
        $login_token =  sha1($token);
        $user_id = DB::query('SELECT user_id from login_tokens where token="'.$login_token.'"');
        $student_classes = DB::query('SELECT login_tokens.user_id, students.name as student_name, student_classes.class_id,
                                    classes.name as class_name
                                    FROM students
                                    INNER JOIN login_tokens ON students.id=login_tokens.user_id
                                    INNER JOIN student_classes ON students.id = student_classes.student_id
                                    INNER JOIN classes ON student_classes.class_id = classes.id
                                    WHERE login_tokens.token="'.$login_token.'"
                                    ');
        return $student_classes;
    }

}

$userid = Login::isLoggedIn();

?>
```

The MyClasses controller houses the get User Classes function. It firstly queries the user's id from the login tokens table. It then go's about querying all relevant data regarding the students including their classes and class names.

II.XII.VI Commenting

```php
<?php
class Comment {
	public static function createComment($commentBody, $postId, $userId) {

		if (strlen($commentBody) > 160 || strlen($commentBody) < 1) {
			die('Incorrect length!');
		}

		if (!DB::query('SELECT id FROM posts WHERE id=:postid', array(':postid'=>$postId))) {
			echo 'Invalid post ID';
		} else {
			DB::query('INSERT INTO comments VALUES (\'\', :comment, :userid, NOW(), :postid)', array(':comment'=>$commentBody, ':userid'=>$userId, ':postid'=>$postId));
		}

	}

	public static function displayComments($postId) {

		$comments = DB::query('SELECT comments.comment, users.username FROM comments, users WHERE post_id = :postid AND comments.user_id = users.id', array(':postid'=>$postId));
		foreach($comments as $comment) {
			echo $comment['comment']." ~ ".$comment['username']."<hr />";
		}
	}
}
?>
```

The above code snippet is used to facilitate the posting of comments within the system. This is achieved via the above function being called when a user creates a comment via any user Status on the system they have access to (Posts from School/Friends). Once they post the comment the following occurs:

1) Comment, post and user information is passed.
2) Comment is checked to ensure it is within reasonable length.
3) Relevant Post is selected based on post Id.
4) Comment is inserted based on arguments passed to the function.

The above function is used to create comment, this is followed by the displayComments function which is used to display comments within the system. This is achieved by querying the database to retrieve all relevant comment data corresponding to the post Id passed to the displayComments function.

II.XII.VII Newsfeed/home.

The home controller calls upon a large amount of functions within the Skewley system. The Skewleynavigation file also calls upon this controller to display data regarding, messages, friends, and notifications. The first function in the above snippet is the post status function. This function is responsible for creating all relevant data record depending on the status activity is occurring. Once the tables have been updated the page is reloaded to show these changes. The next section of this function goes about checking if the status image meets size requirements. It then goes about moving the image from the tmp to status folder providing it has met requirements. This is followed by inserting the images corresponding data records.

```
public function fetchStatus(){
    $token = sha1($_COOKIE['SNID']);
    $user = DB::query("SELECT user_id FROM login_tokens WHERE token='".$token."'");
    $fetch_status = DB::query("SELECT status.user_id, status.body, status.created_at,
                               status.id as status_id, users.username
                               FROM status
                               INNER JOIN users ON status.user_id = users.id
                               WHERE user_id='".$user[0]['user_id']."'
                               and parent_id=''
                               ORDER BY status.id desc");
                return $fetch_status;
}

public function fetchFriendStatus(){
    $token = sha1($_COOKIE['SNID']);
    $user = DB::query("SELECT user_id FROM login_tokens WHERE token='".$token."'");
    $fetch_id = DB::query("SELECT friends.user_id as uid from friends
                               WHERE friends.friend_id='".$user[0]['user_id']."' and
                               friends.accepted='1'
                               UNION
                               SELECT friends.friend_id AS uid  from friends
                               WHERE friends.accepted='1' and
                               friends.user_id='".$user[0]['user_id']."'
                               ");
    $user_id = [];
        foreach($fetch_id as $fetch){
            $user_id[] = $fetch['uid'];
        }
       array_push($user_id,$user[0]['user_id']);
            $id = implode(",", $user_id);
        $status = DB::query("SELECT status.id as status_id,status.body,status.user_id,
                            users.username,status.created_at,users.profile_image,
                            COUNT(status_like.status_id) as like_count,
                            status.status_image
                            FROM status
                            INNER JOIN users ON status.user_id = users.id
                            LEFT JOIN status_like ON status.id = status_like.status_id
                            WHERE parent_id=''
                            AND status.user_id IN($id)
                            GROUP BY status.id
                            ORDER BY status.id desc
                            LIMIT 10
                            ");
        return $status;
}
```

Following the post status, the fetch status and fetchfreind functions are created allowing, allowing users to fetch all updates regarding their or their friend's statuses.

The above functions allow users to reply and view their friend's walls.

```
public function friendInfo($fid){
    $user = DB::query("SELECT username, id, profile_image,about,email,location FROM users WHERE id='".$fid."'");
    return $user;
}

public function isFriend($fid, $uid){
    $friend = DB::query("SELECT * FROM friends
                         WHERE (user_id='".$uid."' and friend_id='".$fid."')
                         OR (friend_id='".$uid."' and user_id='".$fid."')

                         ");
        return $friend;
}

public function unfriend($fid, $uid){
    $del_friend_request = DB::query("DELETE FROM friends
                                     WHERE (friend_id='".$fid."'
                                     and user_id='".$uid."')
                                     or (friend_id='".$uid."'
                                     and user_id='".$fid."')
                                     ");
        header('location: friends?res=deleted');
}
public function sendFriendRequest($fid, $uid){
    $created_at = date("Y-m-d H:i:s");
    $check = DB::query("SELECT * FROM friends
                        WHERE user_id='".$uid."'
                        and friend_id='".$fid."'");
  if(count($check) > 0){
      echo '<script>window.location="/Schoooly/friends?res=already"</script>';
  }
  else{
      $status_id = '';
      $text = 'friend_request';
      $notification = DB::query("INSERT INTO notifications
                                    (user_id,status_id,notification_to,text,created_at)
                                    VALUES('".$uid."','".$status_id."',
                                           '".$fid."',
                                           '".$text."', '".$created_at."')");
      $friend = DB::query("INSERT INTO friends (user_id, friend_id, created_at)
                           VALUES('".$uid."', '".$fid."',
                                  '".$created_at."')");
      echo '<script>window.location="/Schoooly/friends?res=sent"</script>';
  }
```

```
public function fetchReply(){
    $fetch_reply = DB::query("SELECT status.user_id, status.body, status.created_at,
                              status.id as status_id, status.parent_id, users.username,
                              users.profile_image
                              FROM status
                              INNER JOIN users ON status.user_id = users.id

                              ORDER BY status.id desc");
    return $fetch_reply;
}

public function postReply($data){
    $reply = addslashes($data['reply']);
    $reply_id = addslashes($data['reply_id']);
    $created_at = date("Y-m-d H:i:s");
    $token = sha1($_COOKIE['SNID']);
    $user = DB::query("SELECT user_id FROM login_tokens WHERE token='".$token."'");
    $status_original_user = DB::query("SELECT user_id FROM status WHERE id='".$reply_id."'");
    $text = 'commented';
    $notification = DB::query("INSERT INTO notifications
                                  (user_id,status_id,notification_to,text,created_at)
                                  VALUES('".$user[0]['user_id']."','".$reply_id."',
                                         '".$status_original_user[0]['user_id']."',
                                         '".$text."', '".$created_at."')");
    $post_reply = DB::query("INSERT INTO status (user_id, parent_id, body, created_at)
                             VALUES('".$user[0]['user_id']."','".$reply_id ."','".$reply."',
                             '".$created_at."')");
    header('location: home');
}

public function friendsWall($fid){
    $fetch_id = DB::query("SELECT friends.user_id as uid from friends
                              WHERE friends.friend_id='".$fid."' and
                              friends.accepted='1'
                              UNION
                              SELECT friends.friend_id AS uid  from friends
                              WHERE friends.accepted='1' and
                              friends.user_id='".$fid."'
                              ");
    $user_id = [];
        foreach($fetch_id as $fetch){
            $user_id[] = $fetch['uid'];
        }
       array_push($user_id,$fid);
            $id = implode(",", $user_id);
        $status = DB::query("SELECT status.id as status_id,status.body,status.user_id,
```

The above snippet shows the unfriend function in conjunction with the send friend request function and the relationship each of these functions have with their corresponding tables/records.

```
public function getFriends($uid){
    $friend = DB::query("SELECT u1.username, u1.id,u2.id as uid, u1.profile_image,
                                friends.accepted,u2.username as uusername,u2.profile_image as uprofile_image
                                FROM friends
                                INNER JOIN users as u1 ON friends.user_id = u1.id
                                INNER JOIN users as u2 ON friends.friend_id = u2.id
                                WHERE (friends.friend_id='".$uid."'
                                or friends.user_id = '".$uid."')
                                and friends.accepted='1' ");
        return $friend;
}

public function receivedMessages($id){
    $message = DB::query("SELECT messages.user_id, messages.sent_to_id,
                          messages.message_body, messages.created_at,
                          messages.read,users.username,users.profile_image,
                          messages.id as message_id
                          FROM messages
                          INNER JOIN users ON messages.user_id = users.id
                          WHERE messages.sent_to_id = '".$id."'
                          ");
    return $message;
}

public function notifications($uid){
    $notification  = DB::query("SELECT notifications.user_id, users.username,
                               notifications.status_id,notifications.created_at,
                               notifications.text,notifications.read,
                               notifications.id as notification_id
                               FROM notifications
                               INNER JOIN users ON notifications.user_id = users.id
                               WHERE notifications.notification_to = '".$uid."'
                               ORDER BY notifications.created_at desc");
     return $notification;

}
```

The End of the home controller houses the getFriends, receivedMessagaes and notifications function in the home controller. The getFriends function users the inner join function to create a list of friends. The received messages function and notifications functions gets users notifications and messages.

II.XI.VII Profiles

```php
<?php
class Profile{
    public function getAuthUser(){
        $token = sha1($_COOKIE['SNID']);
        $user = DB::query("SELECT users.username, users.first_name, users.last_name,
                           users.email, users.location, users.id, users.about,users.profile_image
                           FROM users
                           INNER JOIN login_tokens ON users.id = login_tokens.user_id
                           WHERE login_tokens.token='".$token."'
                           ");
            return $user;
    }

    public function uploadProfile($data){
        $about = $data['about'];
        $user_name = $data['user_name'];
        $first_name = $data['first_name'];
        $last_name = $data['last_name'];
        $location = $data['location'];
        if(!empty($data['pwd'])){
            $password = password_hash($data['pwd'], PASSWORD_BCRYPT);
        }
        else{
            $password = '';
        }

            $token = sha1($_COOKIE['SNID']);
       $user = DB::query("SELECT user_id from login_tokens WHERE token='".$token."' ");
        if(empty($password)){
             $update_user = DB::query("UPDATE users set username='".$user_name."',
                               first_name='".$first_name."', last_name='".$last_name."',
                               location='".$location."', about='".$about."'
                               WHERE id='".$user[0]['user_id']."'
                               ");
        }
        else{
             $update_user = DB::query("UPDATE users set username='".$user_name."',
                               first_name='".$first_name."', last_name='".$last_name."',
                               location='".$location."', about='".$about."', password='".$password."'
                               WHERE id='".$user[0]['user_id']."'
                               ");
        }



            header('location: profile?suc=updated');

    }

}
?>
```

The above code snippet depicts how users are authenticated in the system using their relevant session id which is generated per user login. This login token is then associated to the relevant user which is derived from querying all user details from the table based on associated log in token.

All user data is then passed to the upload profile function which associates users to relevant profile data fields displayed on the profile page of the social media section. These fields are furthermore updated with all relevant logged in user data.

II.XII.VX Liking

```php
<?php
class LikeController{
    public function likeStatus($data){
        $status_id = $data['status_id'];
        $user_id = $data['user_id'];
        $created_at = date("Y-m-d H:i:s");
        $status_original_user = DB::query("SELECT user_id FROM status WHERE id='".$status_id."'");
        $text = 'liked';
        $check = DB::query("SELECT * FROM status_like
                            WHERE status_id='".$status_id."'
                            and user_id='".$user_id."' ");
            if(count($check) > 0){
                $data = ['res'=>'liked'];
                return $data;
            }
            else{
                $notification = DB::query("INSERT INTO notifications
                                           (user_id,status_id,notification_to,text,created_at)
                                          VALUES('".$user_id."','".$status_id."',
                                                 '".$status_original_user[0]['user_id']."',
                                                 '".$text."', '".$created_at."')");
                $like = DB::query("INSERT INTO status_like (status_id, user_id, created_at)
                                   VALUES('".$status_id."','".$user_id."','".$created_at."')");
                $fetch_liked = DB::query("SELECT * FROM status_like
                                          WHERE status_id='".$status_id."' ");
                $data = ['res'=>count($fetch_liked)];
                return $data;
            }
    }
}
?>
```

The above code snippet displays how statuses are liked within the system. This is achieved by relevant status and user data being checked to ensure said user has not liked a given status yet. Should they not have done so they are enabled to.

After the user has liked the status a notification is created for this process. This is followed by all like data being inserted to relevant status id. To retrieve this like data the fetch_liked query is called which selects all status like per status. The count of each like associated is then calculated to be displayed on each status.

II.XII.VXI Searching

```php
<?php
class SearchFriend{
    public function searchFriend($uid, $q){
        $friend = DB::query("SELECT u1.username, u1.id,u2.id as uid, u1.profile_image,
                            friends.accepted,u2.username as uusername,u2.profile_image as uprofile_image
                            FROM friends
                            INNER JOIN users as u1 ON friends.user_id = u1.id
                            INNER JOIN users as u2 ON friends.friend_id = u2.id
                            WHERE (friends.friend_id='".$uid."'
                            or friends.user_id = '".$uid."')
                            and friends.accepted='1'
                            and (u1.username LIKE '".$q."%' or u2.username LIKE '".$q."%')
                            ");
        return $friend;
    }
}
?>
```

The above code snippet shows us how searching is executed within the Skewley System. When a user inputs a friends or Club page the above query is executed. This query selects some of the user friends profile information, this is then joined with user tables to ensure both users are in one another's friends list which is finally confirmed by ensuring that the 'friends.accepted' column is set to true. Once each of these conditions have been met the LIKE condition returns all friends names in the database which are like the users input.

II.XI.VII Load more homepage posts

```php
<?php
class LoadMore{
    public function loadMorePost($uid, $off_id){
        $fetch_id = DB::query("SELECT friends.user_id as uid from friends
                               WHERE friends.friend_id='".$uid."' and
                               friends.accepted='1'
                               UNION
                               SELECT friends.friend_id AS uid  from friends
                               WHERE friends.accepted='1' and
                               friends.user_id='".$uid."'
                               ");
        $user_id = [];
            foreach($fetch_id as $fetch){
                $user_id[] = $fetch['uid'];
            }
           array_push($user_id,$uid);
                $id = implode(",", $user_id);
            $status = DB::query("SELECT status.id as status_id,status.body,status.user_id,
                                users.username,status.created_at,users.profile_image,
                                COUNT(status_like.status_id) as like_count,
                                status.status_image
                                FROM status
                                INNER JOIN users ON status.user_id = users.id
                                LEFT JOIN status_like ON status.id = status_like.status_i
                                WHERE parent_id=''
                                AND status.user_id IN($id)
                                GROUP BY status.id
                                ORDER BY status.id desc
                                LIMIT 10 OFFSET $off_id
                                ");
            return $status;
    }
}
?>
```

The above function is executed to load more posts on the social media panel should a user scroll to the bottom of their news feed.

II.XI.VIII Sending Friend Requests

```
public function sendFriendRequest($id){
    $token = sha1($_COOKIE['SNID']);
    $auth_user = DB::query("SELECT user_id from login_tokens
                            WHERE login_tokens.token='".$token."'
                            ");
    $created_at = date("Y-m-d H:i:s");
    $check = DB::query("SELECT * FROM friends
                        WHERE user_id='".$auth_user[0]['user_id']."'
                        and friend_id='".$id."'");
  if(count($check) > 0){
      echo '<script>window.location="/Schoooly/friends?res=already"</script>';
  }
  else{
      $status_id = '';
      $text = 'friend_request';
      $notification = DB::query("INSERT INTO notifications
                                    (user_id,status_id,notification_to,text,created_at)
                                   VALUES('".$auth_user[0]['user_id']."','".$status_id."',
                                          '".$id."',
                                          '".$text."', '".$created_at."')");
      $friend = DB::query("INSERT INTO friends (user_id, friend_id, created_at)
                           VALUES('".$auth_user[0]['user_id']."', '".$id."',
                                  '".$created_at."')");
      echo '<script>window.location="/Schoooly/friends?res=sent"</script>';
  }
```

The code snippet above represents the process of sending friend request within the application. A check is done to see if the users are already friends, if so the already friends script is executed. If they are not friends, a notification is created to alert the other user of the friend request. A record of this is sent into the friends table which waits for the potential friend to accept or decline the invitation.

II.XI.XX Activate User Controller.

```php
<?php session_start();
class User{
    public function activateUser($id){
          $previous_location = $_SESSION['previous_location'].'?suc=activated';
         // echo $previous_location;
              $user = DB::query('UPDATE users set activated="1" WHERE id="'.$id.'"');
              header('location:'.$previous_location);
    }
}
?>
```

This controller houses the activateUser function which is called every time an account is needed to be activated. This function can be called from anywhere. It sets activated to 1 for activated users, and once complete goes back to previous location.

II.XII UNIT TESTING

Unit testing was conducted on the system to ensure all code functioned as intended. PHP unit was used. Known for being the most popular testing language for the PHP rendering language, PHP unit proved difficult in areas to use in the beginning stages, however overall was the most appropriate means of ensuring the system functioned as efficiently as possible.

The below snippet is an example of a simple unit test which was ran on the system. Please view the unit testing folder in the project source code to examine all unit tests conducted in greater detail.

```
terry1221:~/workspace $ vendor/bin/phpunit
Failed loading /usr/local/lib/php/extensions/no-debug-non-zts-20100525/xdebug.so:  /usr/local/lib/php/extensions/no-debug-non-zts-20100525/xdebug.so:
uch file or directory
PHPUnit 4.8.36 by Sebastian Bergmann and contributors.

.

Time: 82 ms, Memory: 4.50MB

OK (1 test, 1 assertion)
terry1221:~/workspace $ vendor/bin/phpunit
Failed loading /usr/local/lib/php/extensions/no-debug-non-zts-20100525/xdebug.so:  /usr/local/lib/php/extensions/no-debug-non-zts-20100525/xdebug.so:
uch file or directory
PHPUnit 4.8.36 by Sebastian Bergmann and contributors.

.
```

II.XII TRUNK TEST

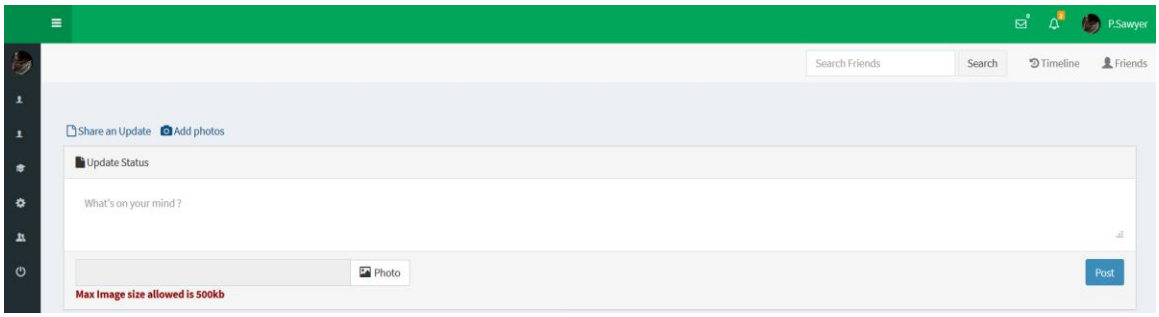
The trunk test was used on three of the systems potential users. These were inclusive all user types. The goal of trunk test in this specific scenario was to determine whether users where able to navigate and know all options to navigate available when let out at a random section of this website. This was determined by conducting numerous questions which were aimed to accurately test user's system navigational understanding.

The trunk test was created to analyze websites navigability, a common form of usability testing used on websites, which is derived from the ideology of a blindfolded subject being put within the boot of a car, to then be drove around at random and let out. Once the subject is "let out" they are then prompted with the task of figuring out where they are. This is metaphorically a representation of the systems users being let out randomly on a site to test how easy it was to figure them out where they are on said website.

To ensure this testing method on the SKEWLEY system I first went about ensuring the user could not see any of the systems web pages(blindfolded). Out of the test subjects view, I then went about navigating to the bottom of the home/news section of the SKEWLEY system. (drove around) This was followed by conducting several navigation questions (figure out where they are).

This form of usability testing is used to determine the amount of taught needed for users to effectively get around the system. The aim of the system is for users to be able to navigate with minimal taught/simply as possible. This testing method also identifies all pros/cons for users navigating the system.

The following snippet depicts where our 3 test users were let out on the system;



The following is a list of questions users were asked once they were "let out" on the SKEWLEY system.

1. ***What website are you on?***
2. ***What web page are you on?***
3. ***What are the major sections of the website?***
4. ***What options do I have on this level***
5. ***Where am I in the grand scheme of things?***

The following grid represents the answers given by all participants who took part in the trunk test

Questions:	Parent	Teacher	Student
1. What website are you on?	Skewley	Skewley	Skewley?
2. What web page are you on?	The homepage.	News section	Newsfeed
3.What are the major sections of the website?	Home, profile, settings, Pupil	The news and pupil sections, also there seems to be an option to view notifications and messages which must be important.	Newsfeed, messages, search, notifications?
4. What options do I have on this level?	I seem to have a tonne of options at this level.	Hmm let's see. I seem to be able to do most things most social networks can do with the addition of the student section.	I can do lots of cool stuff like message my friends, or post stuff, or see what's going on.
5. Where am I in the grand scheme of things?	The home page, probably the most important part of the site?	I am currently on the homepage, which seems to be One of/ if not the most important features the system has to offer.	Newsfeed.

Analyzation of results;

All answers went according to plan, no major outliers were identified. One thing noticed was how all users referred to the homepage differently, however I believe this could be down to all the common phrases are given to the home pages of many of the most popular social media sites today.

The objective of this study was to get a firmer understanding into each participants user experience whilst using the web application. The questions were aimed to test individual's ability to differentiate individual sections of the web application in conjunction with being able to identify the web application itself. Based on the responses provided it was apparent the users easily picked up and remembered the name of the education service they were on and what section. Based on this, the first two questions were deemed to be a success and went according to plan due to the users effectively answering the questions provided.

Question 3 and 4 of this trunk test provided a deeper understanding of the navigational aspects of the web application. The idea of testing their knowledge of the main sections of the web application emphasized their understanding of the navigational options available to them. This was done to ensure that none of the systems main section were overlooked and users both knew where they were and where they were able to go from the homepage, arguably the most important section of the web application, which is evidentially important for a service like this. The difference between question three and four comes down to fact they separate the user's idea of the most important section of the website and their understanding of all options available to them.

In relation to the prior, this was intended to ensure all options appeared available to users, and what were deemed most important, this gives a deeper understanding of the effectiveness of the navigational element of the web application. The objective of the design of this was for users to navigate the system error free, which deemed to be a success based on the answers acquired from this subject group.

II.XIV CUSTOMER TESTING - SYSTEM USABILITY SCALE (SUS)

The system usability scale can be described as reliable and popular tool used for measuring the usability of a website. This scale consists of 10 questions, each of which have five response options for respondents, which range from agreeing to a statement to disagreeing with said.

Some of the benefits of the system usability scale are inclusive of:

1- It is very easy for participants to understand the scale from first glance.
2- Has the benefit or providing accurate results on small sample sizes.
3- It provides meaningful feedback and validates the difference between systems being usable or unusable.

Please find the industry standard, system usability scale below. This helped to determine whether users felt the system to be usable of not.

Questions:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

Systems Usability Scale:

Strongly Disagree 1	2	3	4	Strongly Agree 5
○	○	○	○	○

The following grid represents the answers given by all participants who took part in SUS testing;

Sus Questions:	User 1	User 2	User 3
1	2	3	5
2	2	1	1
3	5	5	5
4	2	2	1
5	4	5	4
6	2	1	1
7	5	5	5
8	1	1	1
9	5	5	4
10	1	1	1
SUS TOTAL:	29	29	28
*2.5	72.5	72.5	70
Total:	215		
SUS:	72		

Analyzation of results;

The average SUS score is 68. Anything over 68 is considered above average and usable whereas anything under 68 is below average. The lower the score the less usable the system. The equivalent to receiving an A top (over 90 percent) in this test you would need to score 80.3. The fact this project scored above average (72) reflects the fact that the system was deemed usable, a positive reflection on the web application, due to the fact this scale is reflective of commercial grade applications and systems, which the system intends to be.

II.XV Customer Evaluation

Neighbor	*"I wasn't expecting such a professional look and feel, no offence! I can totally see myself using this system to stay up to date with school news, and make sure the kids are up to date with everything!"*
Teacher	*"This system will definitely be beneficial for my day to day, being able to directly contact parents is very important, and will be a big help for myself"*
Student	*"The system looks cool and I like the way all my school friends are on here"*

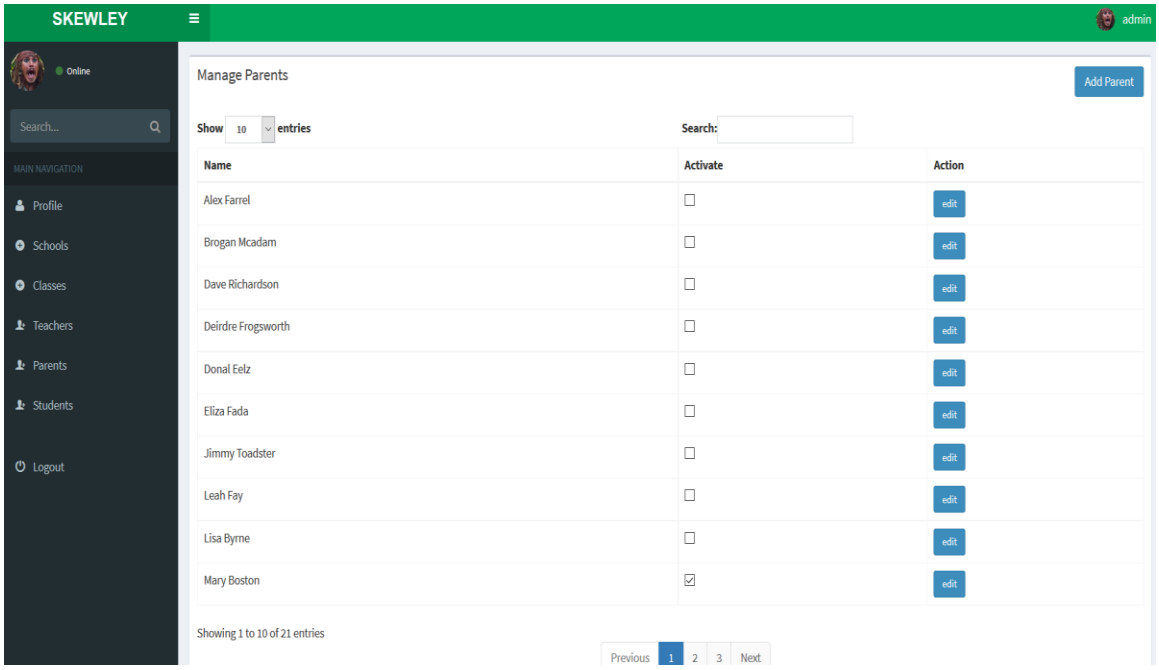
The above customer evaluation reassured the fact the system delivered a professional looking application, the fact my neighbor was surprised by the quality of the web application emphasized this.

The fact the teacher felt that it will be of benefit was good to here as this was one of the systems main user goals for teachers, and it being considered a help would provide the teacher with more time to spend on their students.

The students comments were also a positive reflection of the site and showed us that they taught the site look cool, which really meant a lot given the standard of other popular web applications today.

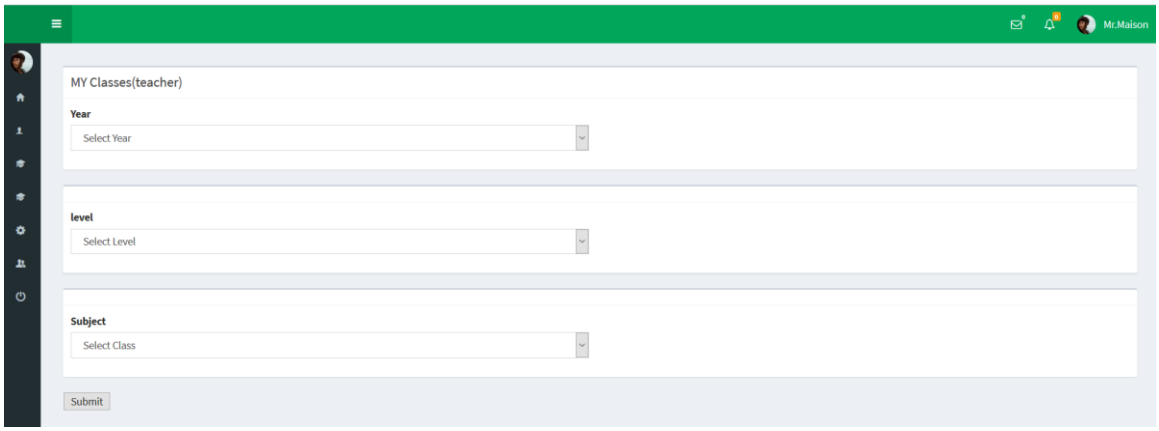
II.XVI G U I

II.XVI.I Administration panel

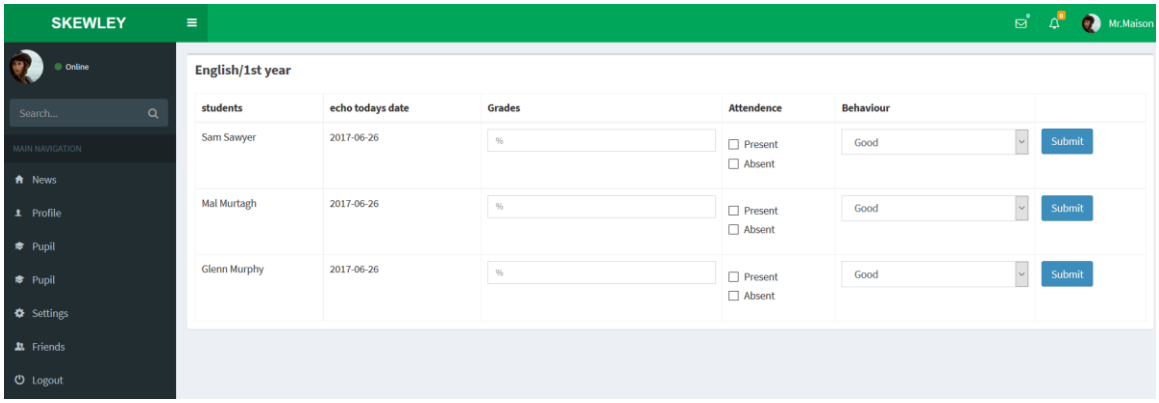


The above snippet is taken from the manage Parents page on the Administration panel.

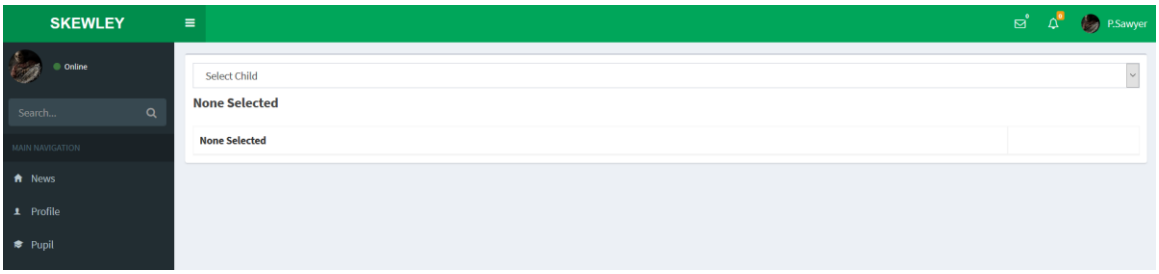
II.XVI.II Student Panel



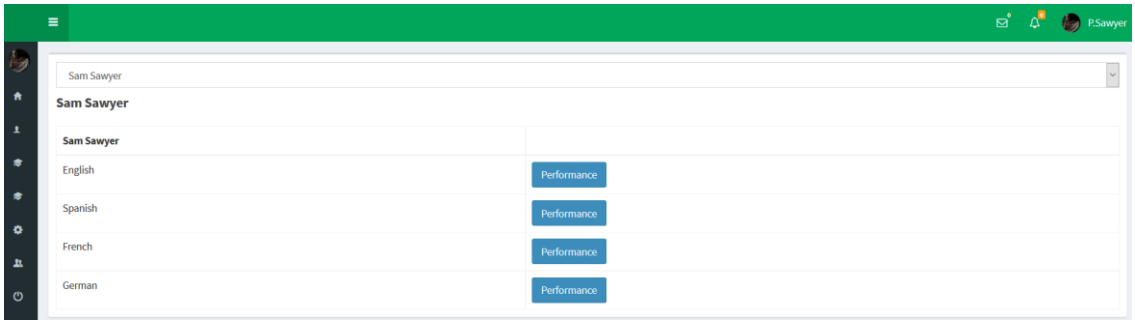
The above snippet is the the result of a Teacher clicking the Student section, here they can filter between Year, Level and Class/Subject.



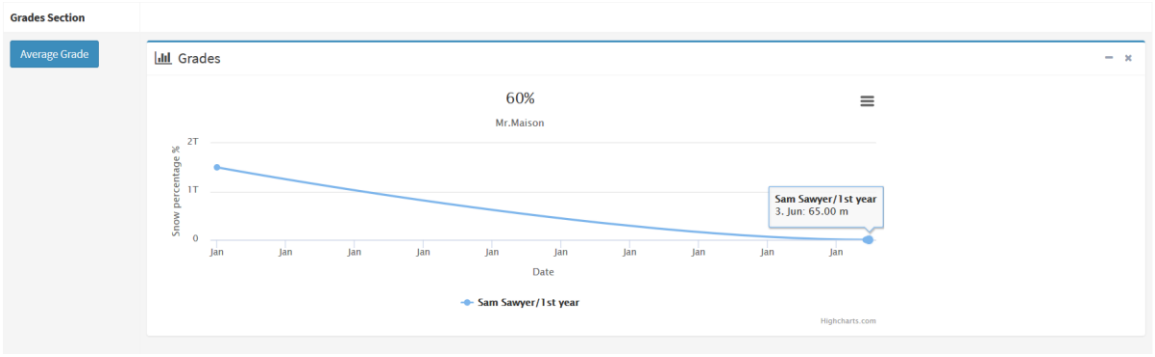
Once a Teacher selects their desired Class, they can then input data on all Students within that Class via the above.



The above snippet shows the Parents viewpoint once they click the student option of their side navigation bar, allowing them to select their Child/Children.

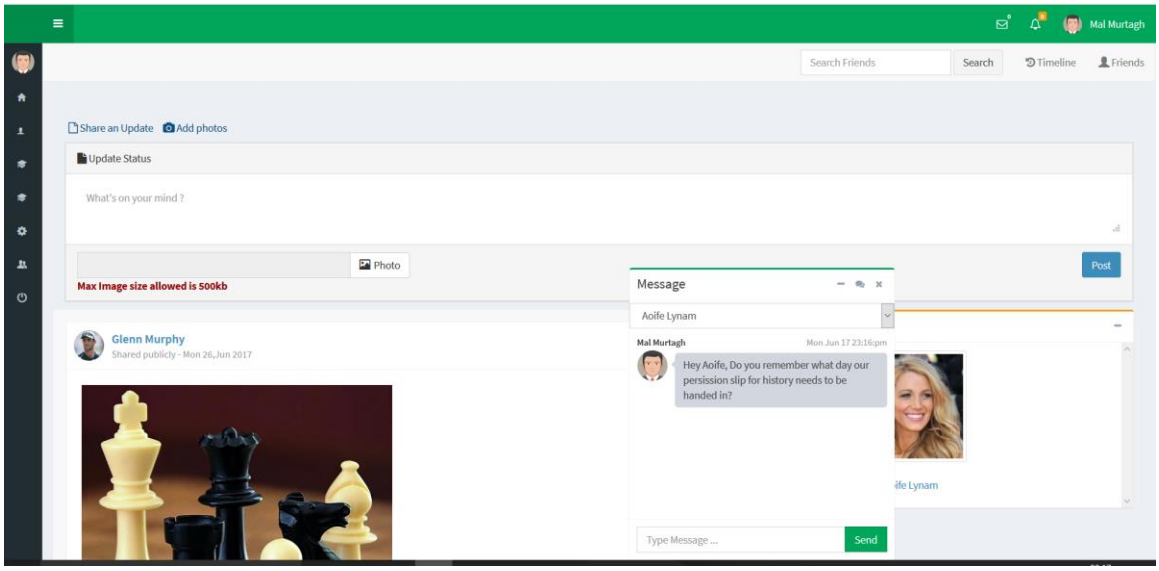


The above snippet shows the results once the parent has selected their Child, this is also the page the Student sees when they click the Pupil section on their side navigation bar.

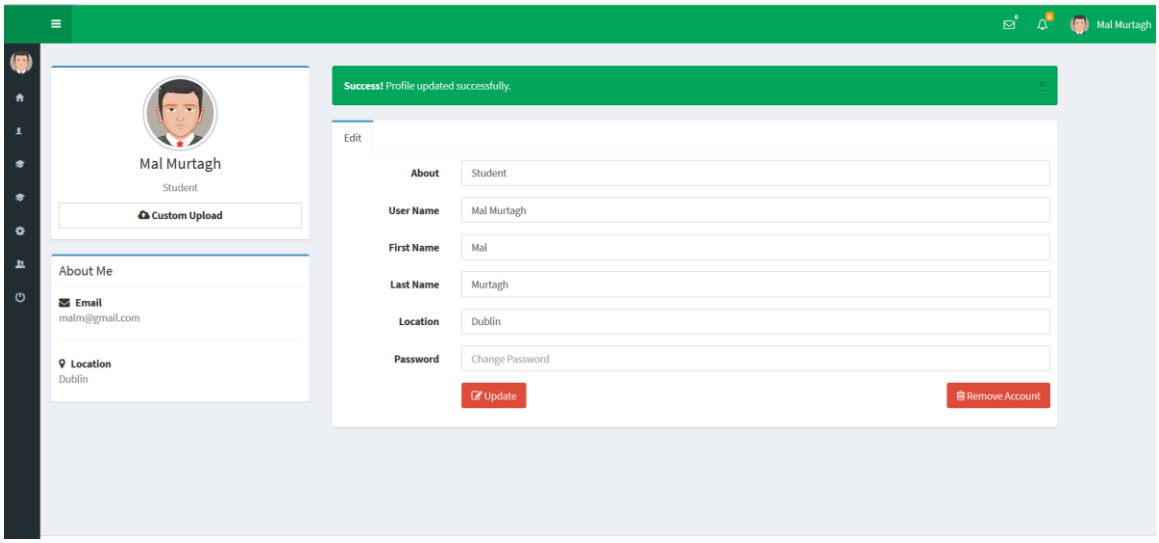


The above snippet is an example of One of the interactive charts that was used

II.XVI.II Social Media Panel

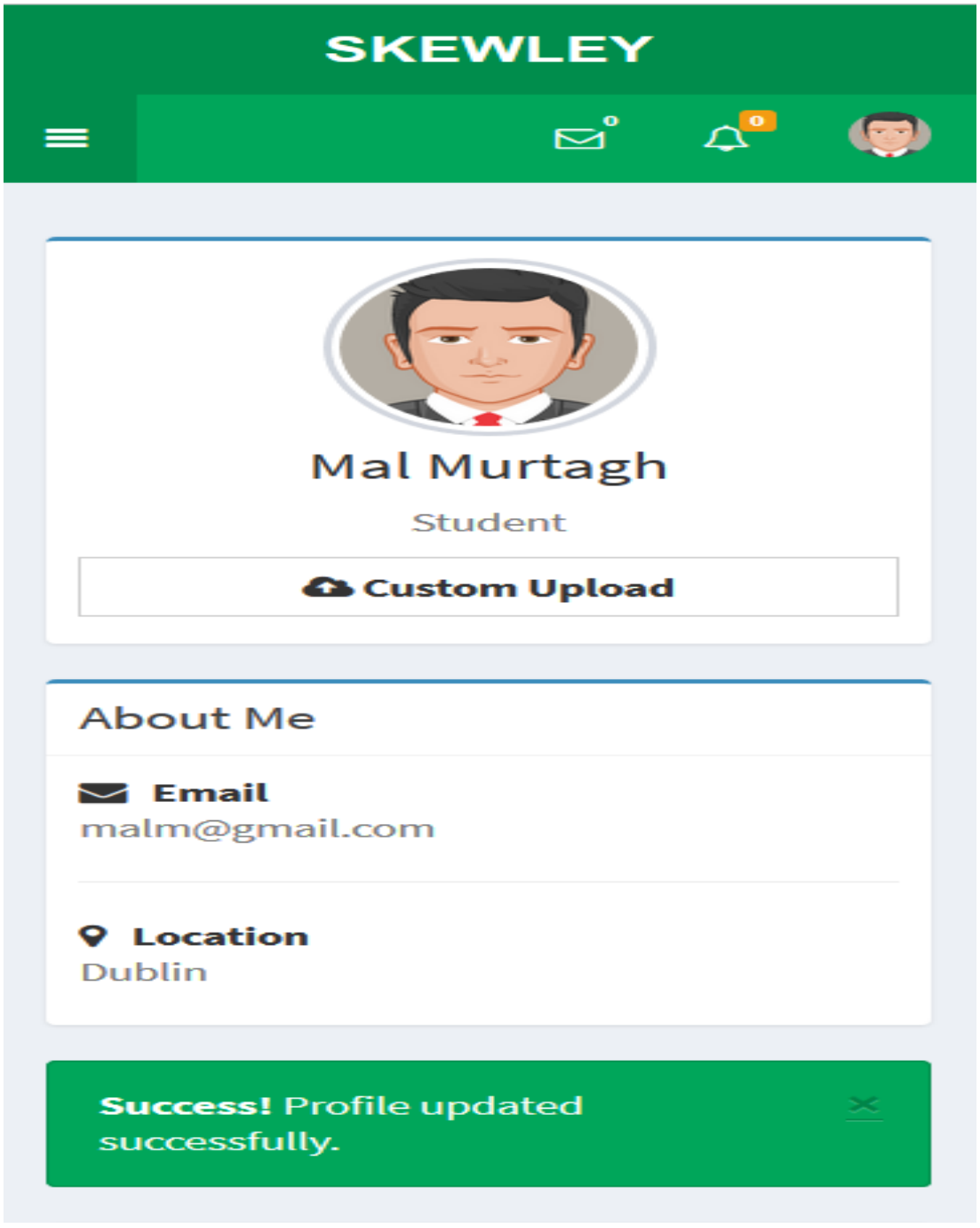


The snippet above shows Skewleys news/home section.



The snippet above shows a users profile section.

II.XVI.IV Mobile Interface



The above snippet shows Skewley being used on a mobie.

II.XVI EVALUATION

II.XVI.I CONCLUSIONS

Developing the SKEWLEY web application has been a truly rewarding and skill testing experience for myself. Choosing PHP as opposed to JAVA (My strongest programming language), came with a combination of both positive and negative consequences. The negatives being everything in class which I thought could contribute to SKEWLEYS developed needed to be self-taught in my own time- which did impact my studies and was time consuming. The benefit of this is that I can now competently code both Android applications and PHP mobile responsive web applications.

If I were to do things again I would consider using JAVA and learning PHP separately in my own time, which would have greatly reduced all risks. The benefits of doing so are- I would have encountered far less problems during development and my project would have tied into what I had been learning throughout the year in class. This was the safe option however, and with the quality of the finished SKEWLEY product, the idea to put myself into a more challenging situation reaped the most rewards, ensuring I fully committed to learning PHP, which will stick to me, making all those late nights and stressful moments worth it.

This experience will make sure that I plan all my projects properly in the future. The idea of rushing into a projects development and missing some crucial elements is a perfect example of the “turtle and hare” conundrum, which I will make sure to never do again.

II.XVI.II MILESTONES AND HURDLES

A multitude of milestones were faced during the development of the SKEWLWY system. To begin the initial system did not cater for Students as a user. I understand my taught process behind this was completely off and I did cater for this once my mistake had been identified during the midpoint presentation. To fix this, I had to remap my database to connect all user types in the system, whilst creating additional log in/ navigational/ functional requirements for this user type.

Another major hurdle this project faced came down to the fact the idea to make the web app a MVC framework was taught off after the midpoint presentation based on the benefit it would have on data storage and retrieval for the system, due to the additional of many classes the original system did not cater for and the direct impact the administration panel had server side which needed to be separated.

During the midpoint presentation, the systems security was brought into question, as the system will be handling very sensitive information about students. This was addressed by both improving password encryption methods implemented (to avoid SQLI and other forms of cyber-attacks) in conjunction with creating the administration panel, which allowed the admin to disable all accounts

with the click of a button. This was harder to do based on the fact it should have been made at the very start of the projects development.

The fact the Initial system I had planned to develop was less complex, my development schedule was impacted as I was left with the dilemma of getting more done in less time. Another difficulty I faced while following an online tutorial to implement sessions and the creation of some basic social media functionalities I did not realize the tutorial was using the old version of PHP, which I then had to recode for it to function properly with the system. The final hurdle I faced was changing the systems front-end from HTML/CSS/JS to bootstrap to ensure a professional look and feel.

II.XVI.III FUTURE DEVELOPMENT

SKEWLEY has the potential to be reconfigured for schools to compare not only students, but classes and class years. This means if the school begins to identify similar trends between a current class and a past class, actions can be taken to ensure the highest amount of student support and prevent the repetition of reoccurring mistakes.

Another route this projects development could take would be regarding implementing a payment system, so students could perhaps pay all annual school fees in One place. Also, an additional books page could be created with the specific task for parents to buy and sell school books from past students/parents to take financial pressure of some parents. This can be done via the current system, however, to have it co functioning is a more appropriate alternative.

Another potential area this project could move into would be to implement Machine Learning. Although the applications architecture may need re tweaking, the potential to make all information directed at all users as relevant and as beneficial as possible is a resounding positive and would help ensure the quality of information posted on the system and for all its users.

As mentioned prior, I will be changing my code slightly for my brothers to be able to use this web application for the Gym. To achieve this, I must 1- combine the SKEWLEY landing page with their current website, via slightly tweaking the front end where needed and 2- delete all current administration panel data, re name schools to clubs and change all classes from current subjects to the martial arts classes they teach.

II.XVII REFERENCES

Cronan, J. and S Matthews, M. (2017). *Dynamic Web Programming: A Beginner's Guide: A Beginner's Guide (ebook) Book by John Cronan and Martin S Matthews*.

GitHub. (2017). *almasaeed2010/AdminLTE*. [online] Available at: https://github.com/almasaeed2010/AdminLTE [Accessed 22 Jul. 2017].

GitHub. (2017). *howCodeORG/how*. [online] Available at: https://github.com/howCodeORG/how [Accessed 22 Jul. 2017].

GitHub. (2017). *howCodeORG/Social-Network*. [online] Available at: https://github.com/howCodeORG/Social-Network [Accessed 22 Jul. 2017].

Hughes, S. (2002). *PHP developer's cookbook*. Indianapolis, Ind.: Sams.

YouTube. (2017). *How to Make a Social Network from Scratch: Part 42*. [online] Available at: https://www.youtube.com/watch?v=Tz0diJIIM58&list=PLA7F9875BD031DC16 [Accessed 22 Jul. 2017].

Nationalgeographic.com. (2018). *These Are the World's Happiest Places*. [online] Available at: https://www.nationalgeographic.com/magazine/2017/11/worlds-happiest-places/ [Accessed 7 May 2018].

Newsweek. (2018). *Why Do So Many Japanese Schoolchildren Kill Themselves?*. [online] Available at: http://www.newsweek.com/why-do-so-many-japanese-schoolchildren-kill-themselves-391648 [Accessed 7 May 2018].

The Irish Times. (2018). *Leaving Cert: Record number taking higher-level subjects*. [online] Available at: https://www.irishtimes.com/news/education/leaving-cert-record-number-taking-higher-level-subjects-1.3108553 [Accessed 7 May 2018].

II.XVIII APENDIX- Project Proposal.

Project Proposal

SKEWLEY – STUDENT DEVELOPMENT APP.

Terry Sheridan| X13416168 | 20/10/2016
BSc (Hons) in Computing.

Mobile Application Development.

Objectives.

Everyone wants to succeed. My primary objective for this project is to in fact succeed and to be able to stand over my result proudly. To be able to commercialize SKEWLEY would truly be a dream come true for me, and I am aiming to have this application developed to such a standard, no matter how much work that takes, that I can begin to make money of it, which ultimately will allow me to peruse creating more apps after college and not have to worry about money.

My aim is to create student development technology which is not only using the most recent technology of today but also serves purpose. To see students, schools, teachers and parents all reaping positives from an app I developed would truly be phenomenal and an extremely proud moment for myself.

By the end of this project I hope my coding skills will greatly improve for me to ultimately be a more confident, reliable and employable coder. By the end of the first semester, should all go well, I will have developed a student registration system, an online platform where parents and teachers can communicate and message each other and a multitude of other functionalities in which to optimize their sons/daughter's chances of succeeding in school.

I aim to create an online environment where parents are no longer kept in the dark when it comes to their children's education. Parents are only informed, bar annual parent teacher meetings, if their child is doing good or bad. There's no in the middle. No time frame to show a list of behavioral issues progressing or reoccurring for parents to deal with the issue before it's too late.

My goal is to put as much time as possible into this project weekly to obtain maximum results for myself and the application itself. You get what you give. I believe should the right amount of time and effort be put into this project I will get to see and understand how different technologies and languages work together which will be a great benefit to me and I set out to obtain new coding functionalities and practices which I don't currently have.

I am to develop an application which allows parent to view their children's progress online, I set out to do this by using P.H.P to allow teachers to post content online to SKEWLEY. Parent can then come online and view this content and message teachers with any concerns. To connect parents and teachers is a big goal of mine, this had been attempted before in other apps however these apps lose relevance quickly. I aim to keep this app relevant via constant updates and that by the end of the second semester I will have a product good enough to launch, and to impress in the project show-case come next summer.

To finalize I am planning on making this app development process as enjoyable as possible by picking technology's which interest me and give me the confidence to be able

to help others if they are ever developing anything similar or using any of the technology's which I set out to use and to get all the help and support I can along away, form both my supervisor, my work friends, my class mates, learning support and my lecturers.

Functionalities aiming to cover. Messaging, Notification System, Account updating, PHP registration system, Behavior Reports, Site A- Parents Interface Site B- Teachers user face. Reminders, Forums, a liking and following system and if I think of any other functionalities I will attach.

Parents get that teachers post data connected by PHP displayed using angular.

Background.

I came up with the idea for SKEWLEY and finalized with this decision over a series of events prior to the beginning of this semester. SKEWLEY proves relevance and significance, executes a cross use of technology and helps with a major problem, which is behavior and performance - strong correlation; for today's students and students of years to come.

At the final year project show case I remember wandering through the packed atrium, an array of impressive projects to be seen. Glancing through I came across some great ideas, and many others which I taught could be used to create or benefit my own project. I was thrilled to see the AngularJS logo plastered everywhere which gave me confidence as I wasn't completely sure I would be able to use it heading into college that day.

I saw numerous great app ideas many of which solved problems in different ways. Ultimately leading me to try and solve a problem which resulted in SKEWLEY. Although leaving cert results have stayed similar enough within the last decade I identified the fact that as technology begins to take more and more of a hold on generations to come, school work will ultimately be affected.

With this is mind I began to think of apps that would tackle this problem, however that's when I realized that if students are given the option between a new game on their phone compared to SKEWLEY its chances of delivering to a high standard would be compromised. I thereafter concluded that if I could bring the benefits the app proposed in via the students' parents as opposed to the student themselves.

Student Development Technology in Ireland is minimal at best. The Irish way of dealing with problems is going to get grinds. Parents assume their child is under achieving however they don't get to see how their child's grades have declined gradually, they just receive the brunt of it. When a lot of the time the damage is irreversible. SKEWLEY keeps parents aware of the gradual changes making the likelihood of a problem developing into something worse less likely.

SKEWLEY is in fact a bridge between students and good grades/ behavior it helps where schools today are not. This is down to the fact that schools just don't have the time in today's day age to be able to be effectively persistent and daily send reports for every student. It is not affordable as it is extremely time consuming.

I brought this concern up with my sister who is a Secondary School teacher and she believed an effective form of communication between her and her students' parents would

Do the world of good in terms of student development. She also mentioned the fact that as everything is automated SKEWLEY will in fact work as a diary for parents and allow them to look over everything at a time that suits them should they busy, as opposed to being met with a phone call about their child which will leave them stressed and anxious for the rest of their day in work.

It is a well determined fact that the education is becoming more and more competitive as the years go by. This is down to the fact the number of students sitting their leaving cert examinations amongst the increase of places being filled in post leaving certificate programs. Although many colleges are expanding to face this need the increase in students cannot be overlooked. With this comes an increase for in points for courses. SKEWLEY helps students reach their full potential by being constantly monitored. It goes into the finer details of student performance and gives back data in which to make sure they can improve in all areas of their study'.

Tackling this problem for my final year project proves a good challenge to engage in as it hits home due to the fact I fell as though software like this could have helped improve my grades and behavior as my parents never had a clue what was going on in my school life.

In today's education system students are evaluated by two sets of examinations and one parent teacher a year; from the parent's perspective. The school is doing no wrong however, they implement tests weekly or whatever the case may be and bring in measure to deal with bad behavior. However, how much of this is reaching home and in what format.

SKEWLEY gives schools the opportunity to improve student performance by connecting all the dots together. Instead of parents thinking their child is doing okay in school to then find out they are failing is a devastating reality and to be able to change this by painting a bigger picture for parents, giving them a broader understanding makes this project come to life for me and allows me to not only do this project to succeed and do well for other students who need of support. Support that today's education system simply can't provide for them, emphasizing SKEWLEYS importance.

Technical Approach.

Research- as I have not found much technology which focus on student development however I have come across numerous sporting applications and lots of accounting solutions. My aim is to combine the methodology of both these applications to optimize my own. When it comes to finding a well-presented platform for allowing parents and teachers to communicate I will look at popular messaging systems, like Facebooks chat.

Literature review- I am going to research the most effective and efficient ways to code my project using PHP. I am also going to consider the best ways to display all data the teachers post and the best PHP methods to post this information via a GUI. I will also do research on behavioral and grading correlations for my own understanding, which I will be able to use to positively impact my project.

My Main Technical Goals are as follows.

Home page- created using angular.

Login system- created using PHP.

Teacher home screen- Multitude of technology's mainly backend.

Parent home screen- Multitude of technology's mainly frontend.

Notification system – PHP

Messaging System – Unknown.

Registration system – PHP.

Progress Report System – PHP.

Behavior Report System – PHP.

Follow Teachers System – PHP.

Current news section- PHP should timing allow.

Allow these functionalities to run via a home screen.

Account settings – unknown.

If I can think of any other functionalities I will include them within my final report however I believe the following cover the clear majority of what is to be included within my final project.

Special Resources required.

In terms of special resources needed for the completion of this project. I will need access to phones, tablets, and a laptop as I am making it device compatible. I will need access to all angular libraries. I am also going to install gulp and other directories should I think they are appropriate during development to maximize my apps performance.

These will include technology used to compress my code and get rid of all extra space to maximize performance speeds and load times.

Project Plan.

Task Name	Start Date	End Date	Duration	Predecessors	% Complete	Status
PHP main elements beginig	**10/29/16**	**11/29/16**	**23d**		**100%**	**In Pro**
Registration system	10/29/16	10/29/16	1d		100%	In Prog
Behavior System	11/19/16	11/21/16	2d		100%	In Prog
Progress System	11/26/16	11/29/16	3d		100%	Not St
Front end optimizing	**12/02/16**	**01/05/17**	**25d**		**100%**	**In Pro**
Home screen	12/14/16	12/15/16	2d		100%	In Prog
parent screen	12/28/16	12/30/16	3d		100%	Not St
teacher screen	12/31/16	01/05/17	5d		100%	Not St
Project Prototype	12/02/16	12/02/16	~0		100%	Not St
MID Way	**05/17/16**	**12/16/17**	**414d**		**100%**	Not St
Presentation	12/16/17	12/16/17	1d		100%	Not St
Bringing app togather and adding funtioanities	04/12/17	04/12/17	1d		100%	Not St
<sub-item>	**05/17/16**	**05/17/17**	**262d**		**100%**	Not St
Showcase Materials	04/17/17	04/17/17	1d		100%	Not St
Final Project hard copies	05/17/17	05/17/17	1d		100%	Not St
Software & Doc Upload	05/17/16	05/17/16	1d		100%	
Project Presentations	05/17/16	05/17/16	1d		100%	

Technical Details.

This project will be developed in angular and PHP using visual studio as my text editor.

I will use PHP to make all registration, notification, messaging systems and will use angular to display all the content at the front end.

The database I am going to use is done in MySQL. This will be used for storing all crucial information.

Evaluation.

To carry out final testing procedures for this technology I will use numerous methodologies. I will test my app via the Teacher Home Screen and the parent home screen. This will be done by creating an account for both parents and teachers and test how when a parent is posting something that parents can see it.

I will test to see that all data teachers upload in being stored and represented correctly on the parent's screens. I will make sure all parents content and messaging between the teacher is also being gotten and sent correctly. I will make sure the process is automatic and takes no time to process these different queries.

I will test data entered for student progress reports is correct and the data being displayed is valid and that the login system is secure and reliable also.

II.XVIII APENDIX- Requirements specification.

1.0Introduction.

The aim of this document is to give an in-depth insight into of the complete SKEWLEY application. This (SRS) document defines high-level product features and all application requirement are entailed accordingly.

1.1 Purpose.

The purpose of the following documentation is to give all requirements relevant to the development of SKEWLEY. This includes a detailed list of all functional and non-functional requirements and emphasizes an in-depth exploration of all components, functionalities, and methods required in correlation to the creation of this application.

These requirements are in conjunction with the fact that the system will be used for a multitude of purposes based on the user and his/her expected result from using SKEWLEY and entails how the system will be tested, implemented and optimized.

To conclude, the function of this SRS documentation is to provide a detailed overview of SKEWLEY including its goals and parameters. It distinguishes how I, my clients and all users view the system and its corresponding functions. It also helps designers with the software delivery lifecycle (SDLC).

The intended users/customers for this application are both parents and teachers of secondary/primary school students.

1.2 Scope.

The scope of creating SKEWLEY is to ultimately develop a cross platform app/website which can be used at any time of the day, anywhere, once of course the user has internet access. This application will provide a direct form of communication between parents and teachers in a fun social environment, encouraging engagement.

Teachers will be able to post all data in regards to students and keep a reliable diary in regards to these postings. This allows them to look back and conduct evaluations on students based over a period as opposed to just the preset, allowing more in-depth and meaningful comparisons and evaluations can be conducted.

Parents can access the platform and view all information in regards to their child, be given a new opportunity to socialize as their will be a recent posts section and a messaging system developed.

Strengthening the means of communication between teachers and parents will have a direct impact on the student. I asked family members and friends about their ideas and advice when creating this application and short listed several things this application is required to do.

2.0 User Requirements definitions.

The User Must have access to the Internet.
An existing user can access the application using his/her email and password.
If a user forgets his/her password an option to send the password to his/her email must be implemented.
If a user is new to the application, he/she must be able to register.
All users must be able to create, update, delete and post via their profiles.
All users must be able to adjust their account settings manually.
All users must be able to message each other at any given time.
All users must be notified should any relevant news concerning them be posted.
All users must be given the option to spread their thoughts and ideas wherever they feel necessary.
All users can access the application on any device.
All users must be able to easily navigate through the application.
The system must be able to store all information it receives appropriately.
Teachers must be able to fill out systems online which will interlink and be displayed on the application.
An existing user (teacher) can view a list of all parents following their profiles.
A new/existing user (teacher) must be able to complete all relevant information about the student and post it online.
A existing user (parent) can view a list of all teachers he/she are following.
An existing user (Parent) must be able to view all content the teachers he/she are following post at any time.
A new/existing user (Parent) must be able to follow his/her child's teachers using their username.

3.0 Requirements Specification.

3.1 Functional requirements.

Navigation;

The application must navigate from page to page in the shortest possible time.
A simple interface ensuring that users do not face any confusion when using must be implemented.
All application functions, services and features must be clearly defined and distinguished.
All options on each of these features must be easy to access and cause no confusion.
Navigation will be mostly similar on all devices unless of course sizing adjustments must be implemented.

Database;

The database must effectively store all essential details in regards to all profiles on the sight. It must also be up to date with all content gotten and sent via the application.
The database must be in constant connection with the web application.
The admin must be able to keep constant track and monitor all news.

Home screen functions;

Search bar.	A fully functioning search bar.
Profile.	Profiles for all teachers/parents and schools.
News.	All current news in relation to the user.
Student.	All information about the student.
Settings.	All account settings.
Messaging.	A messaging service.
Following/followed by.	Users can see who they are following/being followed by.
Home.	All functions must be able to return to the home screen.

Search bar functions.

The search bar must be able to locate all relevant searches based on content stored on the database and content relevant to the user's profile.
This will be done through get and post request via the database.

Profile functions.

The user must be able to personalize his/her profile.

The user must be able to add pictures to his/her profile.
The user must be able to add all relevant information in regards to his/her profile.
The profile must display if the user is on/offline.

News functions.

The news section's primary function is to display news in relation to users and their interests.
An option will be given for users to display the full story in more detail.
The user will be given an option to refresh the news.
The user must be able to return to the new sections after he or she is done reading a story.
The news section must be filtered to ensure all content displayed is suitable.

Student functions.

The student section must display all relevant information in regards to the student.
The student section is where all content posted via PHP registration, behavior and grading systems from the teacher.
The student section is where parents can come online to view his or her student's attendance, behavior and results to date.
The student section must be able to store a record of all students and their past data and all parent's students past date (results, grades, behavior)
If there are any concerns that need to be addressed the student's system links with the notification system.
Teachers must be able to leave comments about performance in any of the mentioned areas as a direct form of feedback.

Settings functions.

Users must be able to access all account settings.
Users must be able to delete accounts.
Users must be able to block and ignore certain users and sources of information they do not wish to.

Messaging functions

A list must be displayed of all users a user is being followed by or is following.
Users must appear online or offline on the messaging service.
Users must be able to view all past messages, delete messages, send and receive messages.
An emoji service may be incorporated.

.

Following/followed by functions

All users must be able to see who they are following or being followed by.
All profiles must be displayed in a list.

.

Registration, Behavioral and grades functions.

Teachers must be able to enter all details in regards to the student via all registration systems.
Teachers must log in to use these registration and grading systems.
All content entered must be displayed to a high standard.
Process time between data entered by teacher into the systems and the time it takes for parents to be able to display these systems must be optimized.

3.1.1Use case Diagram; Homepage functions.

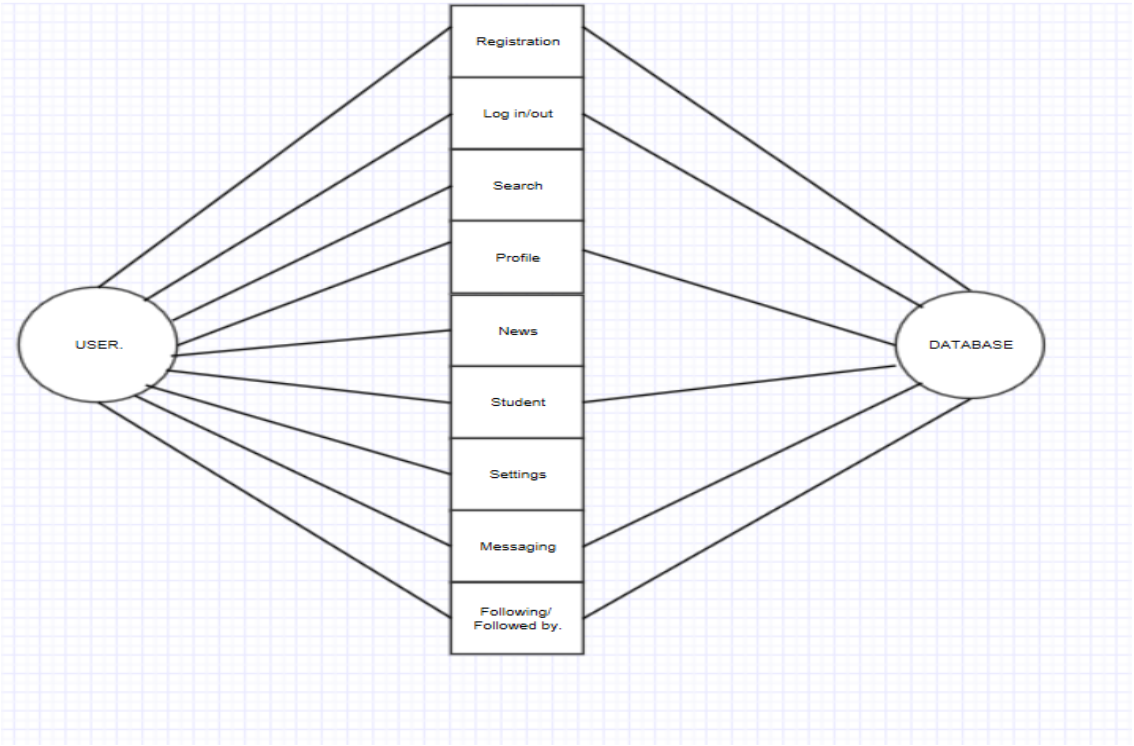
Scope.

The scope of this use case diagram is to display a list of all primary functions to the user and how his/her actions interact with the database.

Description.

This use case diagram describes the key functions offered to teachers and parents using this application.

Use Case Diagram.



Flow Description.

Precondition.

The application is idle.

Activation.

The above use case runs once the application is run in any internet explorer.

Main flow.

1. The user clicks on the register button.
2. The user clicks on the log in button.
3. The user clicks on the search bar.
4. The user clicks on the profile button.
5. The user clicks on the news button.
6. The user clicks on the student button.
7. The user clicks on the settings button.
8. The user clicks on the messaging button.
9. The user clicks on the following/followed by button.

Alternate flow.

A teacher/parent exits the application.

Exceptional flow.

An admin can view all content without creating a profile.

Terminations.

This use case is terminated when the user exits the application or loses access to the internet.

Post condition.

If a user follows any of the functions depicted above and exits the application, he/she will be prompted to log in again.

3.1.2Requirement One, Student Process.

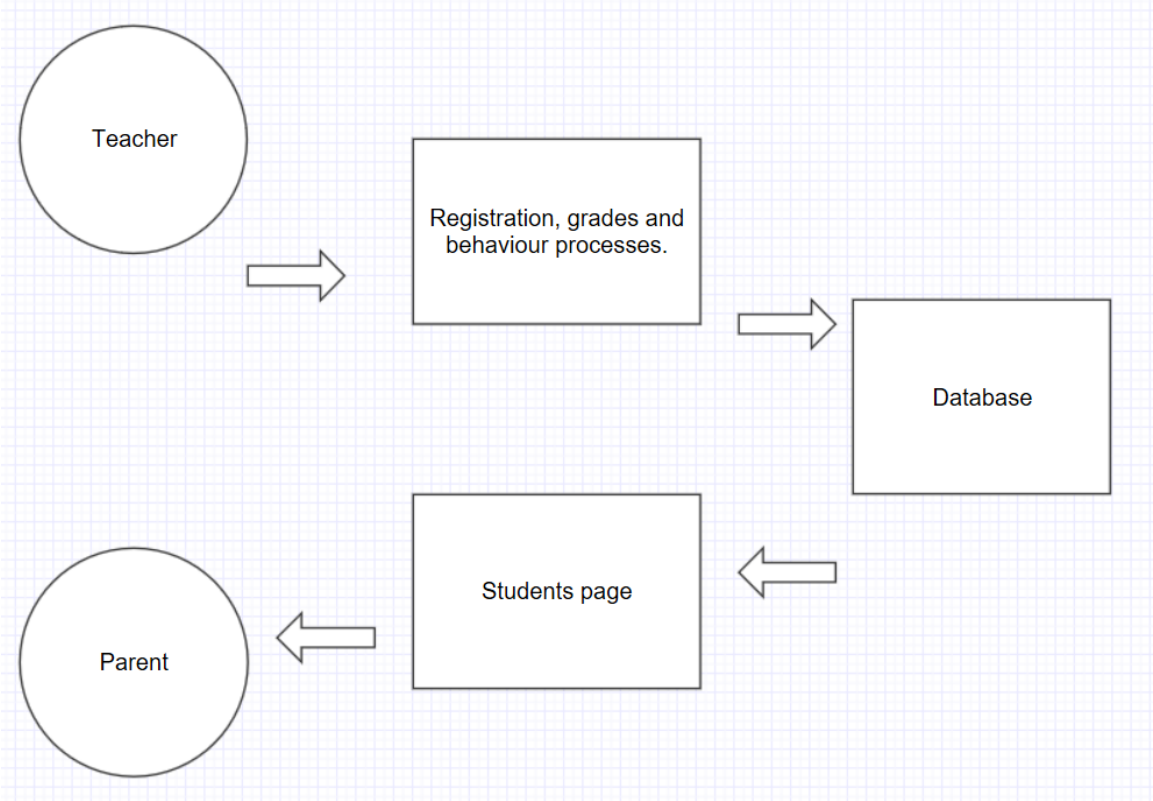
Scope.

The scope of the student process case diagram is to illustrate when a teacher inputs content in relation to a student and how this student's parents view this content.

Description.

This case diagram shows how parents get information in regard to their child.

Use Case Diagram.



Flow Description.

Precondition.

A user is on the applications home page.

Activation.

The use case begins once a click onto the student section.

Main flow.

1. The teacher clicks on the student page via the homepage.
2.The teacher is presented with the option to choose what form of data he/she is inputting. (i.e.) grades, behaviour, attendance or all.
3. The teacher inputs his/her data.
4. This data is then sent to the database.
5. The database then stores all data.
6. A parent clicks on the student button.
7. The database sends all data to the student page.
8. This data in then formatted accordingly.
9. All data requested is now given to the parent.

Alternate flow.

1. The User clicks onto the student button.
2.The application loses connection to the internet.
3. The application prompts the user with an error message.
4. The user is returned to the last page he/she was on while the application waits for a reconnection to the internet.

Exceptional flow.

1. The Parent clicks onto the student button.
2.The parent is prompted a system error.
3.The parent is brought back to the home page.
4. The parent clicks onto the messaging button.
5. The parent clicks the teacher he/she is requesting data from.
6. The parent requests data via message.
7. The teacher receives the message.
8. The teacher sends the parent a message with the data he/she requested.
9. The data requested is now given to the parent.

Terminations.

The use case in terminated when the user posts her gets relevant data and returns to the homepage or exits the application.

Post condition.

All data stored is now filed to keep a record/track of all students.

3.1.3Requirement Two, Login/Registration Process.

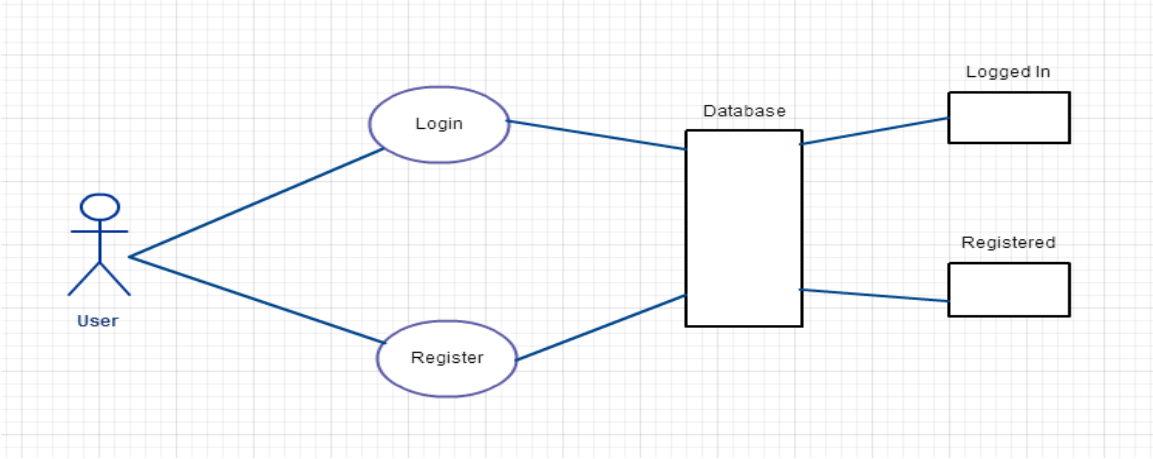
Scope.

The scope of this use case is in relation to logging in.

Description.

The use case bellow depicts the process of logging in and/or registering.

Use Case Diagram.



Flow Description.

Precondition.

The application is idle.

Activation.

This use case starts when a user attempts to log in or register.

Main flow.

1. The User click on the log in button.
2. After this button is clicked a page is displayed asking the user for his/her username and password.

3.The teacher/parent inputs their information.
4. The application checks the database for matching credentials.
5. The system verifies/declines the credentials and the user is granted access or declined.
6.The system sends a message to say the users log in details were successful/unsuccessful.
7. The user is brought to the homepage.

Alternate flow.

1. The User click on the register button.
2. The application displays the registration page.
3.The user enters his/her registration details.
4. The application registers all details entered by the new user to the database.
5. The user is now registered.
6.The user is prompted with a log in page.
7. The user now follows the main flow to get to the homepage.

Exceptional flow.

If a user forgets his/her details they will be a sent them via email and will then be free to follow the main flow accordingly.

Terminations.

The use case is terminated when the user either successfully logs in or is sent to the registration follows that process and then logs in.

Post condition.

User is given the option to have his/her credentials remembered by the application for future use.

3.2Non functional requirements.

3.2.1Performance requirements.

One of the main aims for my application is to perform at maximum speed. This is crucial for ensuring the apps success. I intend to achieve this by installing several dependencies including one to get rid of all unused space and another to compress all code I use to maximise performance speed.

I feel as though through taking these small precautions I will be able to optimizes processing times, in correlation to a strong impact on the applications overall performance.

3.2.3Availability requirements.

In regard to availability; My goal is to commercialise this application so it is available to help all secondary school/high students worldwide.

In the beginning stages of production, I plan to contact schools directly and pitch the product with an annual fee, if successful SKEWLEY will be available to all teachers, parents and students in said school.

As SKEWLEYS popularity increases so will its availability to all potential user.

SKEWLEY will only be available to schools, parents and teachers.

3.2.4Security requirements.

SKEWLEYS security features will include several procedures to ensure a stable app environment for all its corresponding users.

One major security concern I must overcome is the threat of sequel injection. This is due to the fact a large portion of my application will be using PHP which is the most venerable computing language for SQL injection.

I aim to achieve this by ensuring all parameters in relation to search results are well defined and do not provide any links to either my PHP constraints or my database.

Another form of security with I will optimize will be in relation to ensuring only confirmed users can access the site, those who forget passwords will be send reminders to his/her email.

To finalize my security features, should time permit, I aim to include both password and messaging encryption system to optimize my applications security.

3.2.5Realiabilty requirements.

In relation to app reliability If the application ever fails/crashes, the identified problem/error will be stored in the database to ensure it does not reoccur.

3.2.6Portability requirements.

Regarding portability requirements; I intend however am not restricted to ensuring that this application can be run cross platform.

These proposed computing platforms, in addition to being run on mobile are Computers, laptops and tablets.

Ensuring the applications functions all appear to a high standard no matter what device you are using is very important for application development.

This will be achieved by setting screen restrictions based on screen width and length in my initial code base for the SKEWLEY application.

4.0 Interface requirements.

SKEWLEY will be accessed and used via a graphical user interface (GUI). All proposed functions of this application will be carries out by using a touchscreen or by keyboard and mouse. The application is connected to a MySQL database, allowing it to be changed when needed.

4.0 System evolution.

Regarding system evolution; My primary objective for the evolution of SKEWLEY is to enter new markets to help primary school student. The core chunk of the software will be similar however it will be more appealing to the younger generation.

On top of this I intend to increase my application functions complexity. I aim to do this by developing all functions to a standard not available with today's technology.

I also intend to add more features to the student section to give a clearer and better understanding of the student in terms of overall performance.

Reflective Journal 1-3;

x13416168.

Please note these journal catered up until midpoint – it was at this stage I decided to completely redesign the system. Please find these journals attached in the project folder.

Reflective Journal 4;

x13416168.

The month of January showed promising advances in the development of Skewley for a multitude of reasons.

The Three main areas I focused on which were down to feedback from my supervisor and the lecturers present during my presentation were to improve the design, functionality, and complexity of the web application. I set out to achieve these goals from a numerous development procedures and methodologies.

To begin I firstly got external feedback to help with the implementation, because fixing an application in later stages of development is far more complex and time consuming then doing it at the start. However, I feel I have dealt with this concern appropriately.

To ensure the application was functioning to a higher standard and becoming more

To conclude, the final development work I did on Skewley in the month of January was optimizing the web applications design to give for it to look as good as possible. I did this by copying an already popular framework and tweaking it to achieve the desired results I was looking for. I am now a little past half way through development. However, the work I have done to date has been far more time consuming then the work to come. This month plan to implement the remainder of Skewley functionalities. Which I look extremely forward to doing.

Reflective Journal 5;

x13416168

Down to the fact I deferred my project the following journal caters from February to May of the systems development. With the month prior came the redesign of the system to cater for more users and to also include an administration panel. To achieve this, I firstly began redesigning the Database (from scratch) amongst getting external support from my place of work. After ensuring all database tables were linked accordingly, I went about searching for a lightweight MVC framework to support all data transmission within the application. After finding a useful structure online I began reconfiguring it to include all the code I had done to date. This included the loading of individual profiles, login/registration, and a very small part of the home section.

After I had broken my code up appropriately, I then delved into completing the student section of the system, this was completed within around three weeks. To achieve this, I had to ensure my database was structured perfectly to execute all queries to the DB accordingly. This was done by adding all appropriate tables to the db and performing C.R.U.D and join operations for teachers to be able to distinguish and filter between their classes and students and students/parents to read these operations. I used online tutorials and dynamic web programming- a beginner guide, which features a class system I then remade to achieve all in app data transmission in relation to the student section of the system which occurred in house/during use of the app. Once the inner mechanics were completed I went about creating a professional looking front end for the system. To achieve this, I download the (mobile responsive) adminLTE template and rebuilt it to ensure it cater for all sections of the system. This included building all navigational bars and the front end of the student section of the system. The systems charts were also created during this stage of development. To achieve this the high charts liberty has been linked and fed in app date to represent teachers input.

Reflective Journal 6;

x13416168

The following section caters for all development made between May and July for the projections development. I started this time with ensuring that all administrational functions were appropriately with the database. To achieve this, I ensured all input fields for administration panels (pre-made with bootstrap) perfectly connected with the database. Excluding all student tables which were created in house via teacher input. I began by ensuring a student could be added to the system and then went about reusing the code for all other sections- ensuring they were being stored appropriately. After the admin panel was created I went about finish off the social media section of the system. To achieve this, I used the how to create a social media platform from scratch tutorial to understand the inner mechanics of how the social sections would interconnect. I combined this with external support, reading the PHP cookbook which broadened my understanding of the language drastically and made the process of remaking premade code to perform differently depending on the section needed much easier. I firstly went about remaking the social network code, however ensuring it connected with my system, and catering for it being it being made in the older version of PHP.

The tutorial helped me with developing some of the home page and settings section of the system. However, to make the friends, messaging, notifications, and friends pages to the standard achieved I went about remaking preexisting code from tutorials- however, the majority of these had to be completely remade from scratch to fit into the system accordingly. One major benefit of creating the student section first was ensuring that each user could access all user dependent views before I began implementing social functions. I believe this could have resulted in numerous issues.

The final month of development (weekends only) has focused on unit testing the system to ensure it performs as intended. The only errors I am now facing come down to messaging, which I plan to fix before the deadline.

Reflective Journal 7;

x13416168.

The main aim of project development this time around is to fix all errors that were made last year.

For the month of march the focus of the project was to create a suitable schedule which would accommodate for all 15 points of feedback given for my last thesis in conjunction with fixing a list of bugs associated with my codebase.

Major progress has been made in this regard and currently close to having all changes needed included, the main aim is to have all changes made by the end of April, giving time to make any further changes should they be necessary during the month of May/prior to deadline.

These theory changes involve me re writing large sections of my thesis and fixing crucial bugs in my codebase. Inclusive of searching, liking of statuses/comments, friend requests, messages, and changing charts to display data in a more suitable manner.

Questionairre Results.

Please find in project code folder. PDF format was not compatable here.