[1]

AccelMath - An Android application developed by Group 15

Jagrajan S. Bhullar, Ionwyn Sean, Eric Lin and Duc-Phuong (Jason) Nguyen, *Student, CMPT 276*

***Abstract*—This report focuses on overview of the project, motivation and roles, as well as detailed features and architectural/database designs. We will explain some tools and concepts that were used in the development of the application. User interface will be mentioned in the last chapter to give readers better ideas of how the app works on a mobile device.**

I. INTRODUCTION & MOTIVATION

MATH is one of the most important subjects that has been continuously taught in every big university in the world. We found that students, especially those who have recently made a transition from senior year of high school to first year of college, have a lot of struggles to get a get a decent grade in math courses. Keeping that in mind, we want to use our new and fresh knowledge of Android development to create a mobile application that is able to run on multiple devices to help them broadening and sharpening their mathematics skills. This project will hopefully help the computer science and mathematics community as we are engaging students of future generations to adapt and keep pace with the ever-changing and demanding world through the purest discipline of math. Though the project is not initially aimed for profit, the project may open to business negotiations as we are aware that the project may be used in educational institutes. Therefore, we encourage that this software is used for educational institutes under our permission.

II. DOMAIN

We will mainly focus on three major topics that are mostly chosen by first-year college students: Discrete Mathematics, Calculus I&II and Linear Algebra. Each topic is divided into three sub-topics which cover the most essential concepts and problems of it. We will demonstrate each one of them as followed:

A. Discrete Mathematics

Discrete mathematics is the study of mathematical structure that are fundamentally discrete rather than continuous

1) Binomial Theorem
2) Rule of Sum and Product
3) Set Theory

B. Calculus I&II

Calculus is the mathematical study of continuous change, in the same way that geometry is the study of shape and algebra is the study of generalization of arithmetic operations.

1) Limits of Functions
2) Derivatives
3) Integrals

C. Linear Algebra

Linear algebra is a branch of mathematics concerning vector spaces and linear mappings between such spaces. It includes the study of lines, planes, and subspaces, but it is also concerned with properties common to all vector spaces.

1) Determinant Computations
2) Matrix Operations
3) Vectors and Linear Equation

III. ROLES

To ensure everyone on the team is contributing equal amount of work while working on specific tasks they are best excel at, we have divided the most important tasks into four different main roles: Scrum Master, Product Owner, Repository Manager, and Team Member. Even though each of

[1]This report is a part of Sprint 3, submitted on April 5, 2017 for marking. All developers of this project is currently studying Computing Science major at Simon Fraser University.

the team members are given a role, individuals can collaborate with each other. For example, the current Repository Manager can exchange the role from the current Scrum Master and resolve conflicts. Furthermore, since this sprint project is to stimulate an actual project team, the roles will be interchanged for each sprint to provide contrasting experience in the different responsibilities of each assigned roles.

A. Scrum Master

The Scrum Master of this team is to organize group meetings and helping to keep meetings on topic. In other words, the person must organize how activities during the iterations are completed. Ionwyn was the one who organized group meetings and helped to keep them on topic. He decided how activities during each iteration are completed as well as updated uses cases so team members will not get distracted by the process.

B. Product Owner

Jagrajan played the role of Product Owner in this project since he communicated with instructor and TA about details of each iteration, took care of source code submissions and initiate the idea of making the application. He contributed to the build of main features from Iteration 1 and continued to expand functionality in further development phases.

C. Repository Manager

Jason and Eric was in mainly in charge of setting up GitLab repository and helping others with Git issues arisen from branching workflow and merging conflicts. In addition, they gave feedback to team members on how to avoid future Git problems and cleaned up the branch before merging. Other responsibilities are code review and debugged and fixed LaTex renderer.

D. Team Member

Everyone in our group in also a team member. Being a teammate is actually the most difficult since the description of this role is extremely general. a team member's duties include implementing required features, contributing to discussions, communicating with other team members, making decisions with the team, and many more responsibilities. For example, during sprint 3, Eric is the repository manager in charged of our gitlab while being a team member that contributes to discussions. Another example is that even though Ionwyn is the Scrum Master, he still contributed as team member as he has uploaded numerous files and implementing extra features to our application. From sprint 1 to sprint 3, everyone on the team knows what their jobs are and has fulfilled the duties as perfect as they possible can.

IV. FEATURES

In this application, we provide a wide range of functionality focusing on different types of users such as students and teachers (especially non-mathematicians) and want to give them the most satisfactory experience while using our app. Main features include lessons and quizzes and progress tracker, Other features include questions made by community and random inspirational quotes from famous people and figures.

A. Overview of lessons

Each of the main topics consists of several smaller sub-topics. Within the small sub-topics include lessons associated with the topics. We believe AccelMath is an application that has a learning curve that build users' confidence through introductory questions at first before releasing intermediate problems which requires more thinking. Moreover, we feel that somewhat challenging questions is the key for lesson-based applications. For instance, users will easily become dull if we made our application questions too easy and simple while users will become frustrated and lose hope and confidence if we made our problems too difficult. Even though it is extremely difficult to make an application that is both somewhat challenging and academically helpful, our team have accomplished and determined the questions through numerous thoughtful discussions and debate with each lesson and associated topics. Ultimately, in order for the application to actually be educational and beneficial to our users and their grades, our lessons start from easy and simple into more in depth and more complex.

B. Randomly selected quiz

To prevent memorization in order to pass the quizzes, we have specifically designed our system so that every time there will be a different quiz as possible. As a student, we know that with pure memorization, especially just memorizing the answer key, does not help with learning. In order for our users to learn, we aim develop their critical thinking instead of strengthening their short-term memory. During sprint 2, we have had several close friends testing our prototype. However, there weren't much learning involved since all they did was memorize the answer key. As a result, we have made the quiz that is randomly selected to make answer key memorizing more difficult.

C. User's Progress Tracking

AccelMath has create a system which tracks users' learning progress. Because the the progress is stored in the database, users are able to continue where they were left off instead of having to relearn everything again. Users are only allowed to save their progress once they have successfully completed their quiz. Moreover, Progress tracking can show users that whether or not they are close to the end of each section which builds them a sense of accomplishment.

D. Community-made questions

Due to the reason that none of us is professional at designing math related questions, we have to outsource sample questions online or from textbooks. However, to prevent from getting involved in copyright issues, we eventually change how the questions are phrased, alter the number and answers given, and basically do what we can to avoid getting sued. This is why our team decides to introduce community-made questions into our app. The community based questions can also be the solution to our lack of questions. we intended to make the question database similar to how Wikipedia works. The application does not require high cost at creating the problems as most of them will be generated from our users. In addition, we are able to gain more perspective and point-of-view on how questions are created and solved when there are numerous unique individuals generating these questions.

E. Random quotes

In our application, AccelMath, we have included numerous famous quotes from all kinds of famous individuals. we figure that showing these inspiring quotes from time to time can be a good motivation for our users. There is a coaching factor involved in every quote because mankind are naturally aspirational. Just like how it is the nature for humans to bond with each other, individuals tend to look up to role models. The ideas and quotes from leaders, famous people, or powerful individuals can affect us on a primal level. As a result, we believe these random quotes can help our users learn.

V. ARCHITECTURAL & DATABASE DESIGN

The application follows a model-view controller design pattern. The model layer is made up of the *ScoresHandler* and *DatabaseHandler*. Both of these classes follow the singleton design pattern, where is a single instance is shared across all object. The reason for this approach was to avoid unnecessary complexities that can arise when trying to maintain data consistency across multiple instances. Knowing that all the data was going to be consistent throughout the app allowed us to remain confident that the model layer would not provide any trouble throughout the course of development. The *ScoresHandler* and *DatabaseHandler* work very similarly. As mentioned before, they are both singletons. But beyond that, they both utilize JSON files to keep track of data in order to maintain information when the application gets terminated. Furthermore, they both provide an ample amount of methods for developers to implement very complex features. The *DatabaseHandler* also has the ability to fetch files from URLs by utilizing Android's ability to have AsyncTasks, which allows developers to have their program run computationally expensive or long tasks in the background, without stalling the UI or the user experience. Each database is stored with a 'db' prefix so that the handler knows which files to read. Each score file has a 'score' prefix, so that the *ScoresHandler* knows which files to read. Outside of the prefix, these files have identical names to keep track of both the databases and their corresponding scores. By taking care of the model layer early, this let us focus our efforts on what the user cares about most, the view and controller layers.

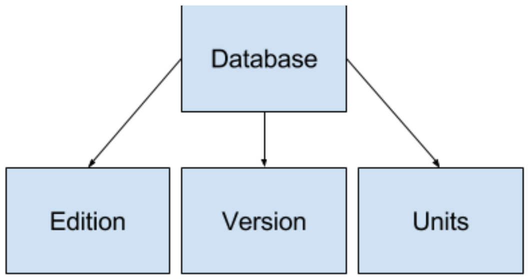
The controller layer was where most of our efforts were put into. We decided early on to use fragments as frequently as possible. The reasoning behind this is it mean that we could reuse them in multiple activities. Although the need never arose, its does give us flexibility in the future. At times, the extra effort to create a Bundle to pass Intent data to a fragment felt like a step in the wrong direction, however it allows us to feel comfortable that this will make it much easier to expand these fragments into other activities and create complex layouts for larger devices. Since Android activities have the ability to show multiple fragments at once, this can prove to be very useful if we wish to make a tablet-friendly design.

Continuing on with the controller layer, our intents passed around many 'Extras' to help display data to the user. At times, intents would have to pass 5 extras to keep track of progress in a quiz. However, since we kept our code clean and very easy to follow in the controller layer, and it was a rather painless process, but time consuming nonetheless. We tried to keep the naming scheme of intents consistent throughout the whole project. The motivation behind this was to reduce any potential problems that can occur when two intents from different applications have the same name. Furthermore, we found it to be a lot more professional if we stayed consistent.

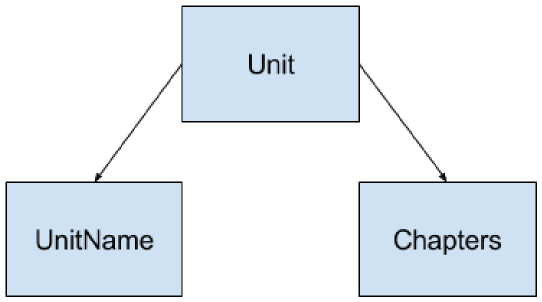
Also on the controller layer, we used the RecyclerView throughout our app, as it presented an efficient way to present similar data in a linear layout. The RecyclerView saves memory as it reuses views known as 'holders'. Once a holder goes off the screen, it is brought back onto the screen, but this time it has new data. By using the RecyclerView, we were able to avoid the effects that a very long linear layout has on memory and overall performance.

The view layer also got a lot of attention as well. We wanted to make our views reusable, while at same time making them tailored towards their needs. We kept our naming very consistent as well. Layouts that were used by activities and the activity prefix. Layouts that were used by fragments had a fragment prefix. And layouts that were used by recycler view holders had a list prefix. This made it very easy to find the corresponding view whenever we had to make changes. Our xml files also contained most of the layout pre-done when inflated by a controller. This made it easier to simply integrate on the controller layer. We tried to keep view ids consistent, however this proved to be challenging, as ids tended to get very long, and we often found ourselves taking shortcuts.

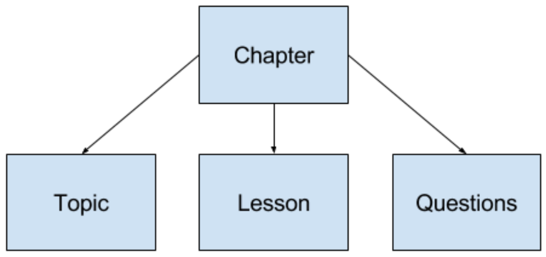
We kept our database design simple and easy to follow. Since we knew we were going to use JSON, we did not use relationship models, and instead stick to hierarchy trees to illustrate how our databases were structured.



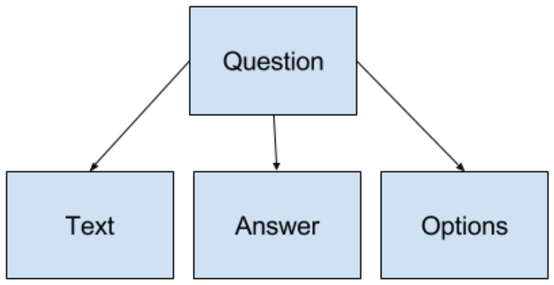
Each database has an edition, version and units. The edition is the unique identifier that is used to differentiate between two databases in the database selector menu. The version allows the app to indicate whether a database is outdated and needs an update. The units lists contains all the units that the user can learn about.



Each unit has a name and a list of chapters. The chapters split into several subtopics of the main topic to make the data more organized and easier to digest.



Each chapter has three main components. The topic which is the title of the chapter, and can give users a small of idea of what to expect. The lesson gives the user some information they can use when answering questions. The list of questions relate to the topic and can come in a multitude of forms.



Each question has text, which is question that the user has to answer. It has several options from which the user can choose from. And finally, it has the correct answer.

A. About LaTex

For AccelMath, we have decided to use LaTeX system to document our lessons, questions, and solutions. Latex is used all over the world by instructors, students, and professionals in almost every field in the world. Latex renders it's unique syntax in to beautiful document, taking away the painful process of having to manually do everything by hand which can take forever to do at times.

B. JSON and LoganSquares

Our app used JSON to maintain databases, and LoganSquares played a critical role as well. We chose JSON as it is very easy to transfer over a network. JSON objects can be serialized into strings, transferred to another device, and that device can parse and use it. When considering our app is community powered, this makes it very easy to quickly upload databases to the web, and quickly share with someone else.

LoganSquares builds the bridge between JSON and Java. LoganSquares takes the JSON files, and converts them into Java objects so that we can use it in our app. LoganSquares also does the reverse, which plays a critical role in scores tracking.

C. MathView and MathJax/Katex

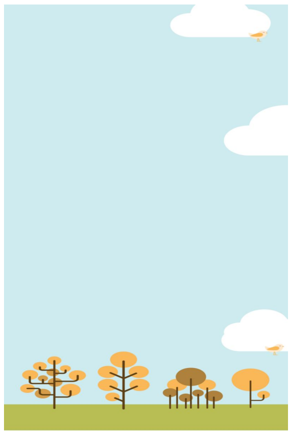
Our app stores native latex code in its databases. This is to save space, as saving each equation as an image would result in a the app taking up a lot of storage. However, if it is stored as raw latex, that would make it unpleasant for the user trying to use the app. This is where MathView comes into the fold. MathView allows us to dynamically generate latex on-the-fly, which saves lots of precious space on the user's device.

MathView can work with two different engines, MathJax and Katex. Although both of these are very popular, they have trade-offs. MathJax renders far more crisp and elegant text and symbols, while sacrificing speed and rendering time. On the other hand, Katex renders noticeably faster, but does not render as many symbols as MathJax, and the rendered output loses some elegance. In the end, we decided to use Katex for the longer lesson, that would take some time to render under MathJax. We decided to use MathJax for questions, as the difference in render times was for less noticeable due to less content being rendered.

VI. USER INTERFACE

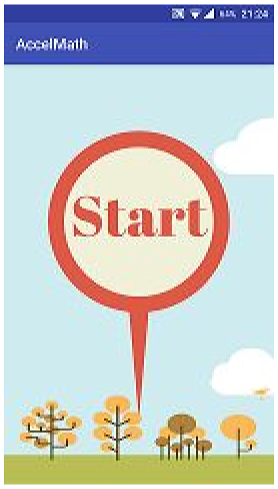
A. Design

First and foremost, we do not accept unoriginal work. We take it seriously even for the background design. To achieve that, we never consulted Google Image for our application background or any other images users can see on the system. Rather, we use Adobe Illustrator in the completion of designing our background image and all the other images within the application. All the work that has been done are made from scratch. Not a single pixel is taken elsewhere. We've also taken into consideration on how user will experience the app, depending on the background. After the completion of the second sprint, we have noticed that some users are experiencing trouble in reading texts on the application due to the blending colours of the application background and the text itself. So we changed it so that users will experience no trouble reading at all. We came up with a background image as below.



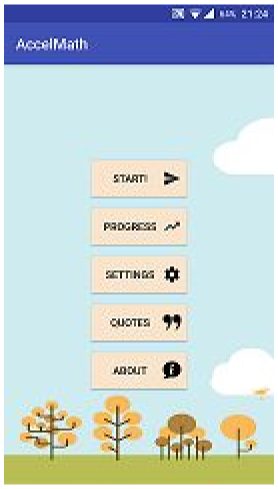
B. Splash Screen

Once the users tap on the app logo appeared on Android menu screen, the programs will immediately start up in less than one second and a welcome screen will pop up for 5 seconds with background music..



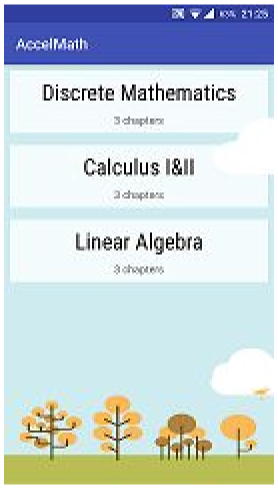
C. Main Menu

After the Start button was clicked, the Main Menu screen will be showed for users to navigate. All main features will be included in this menu as a list. Simply tap on your phone to explore our app.



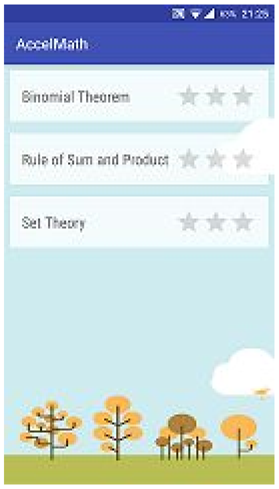
D. List of Chapters

Users can choose which chapter they want to practice after enter the main feature. It will include three main topics as discussed in the Domain section above. Users can easily tap/click to any item on the list to start the lesson and the quiz.



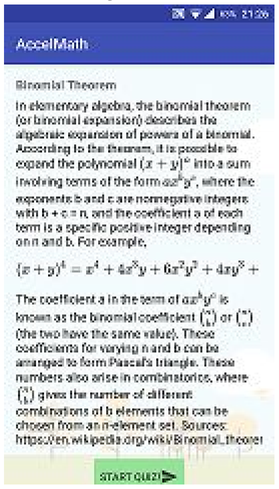
E. List of Subtopics

Inside each chapter there will be several subtopics for users to explore. Details of them can also be found in the Domain section written above. Furthermore, a progress bar will be shown right beside each subtopic so that users can keep track of their improvement.



F. Lessons

Lessons provide a summary of each topic taken from Wikipedia article. Users can use this as guidelines on how to study the subject more efficiently and find more useful information through the link given in the end of each lesson.


AccelMath

Binomial Theorem

In elementary algebra, the binomial theorem (or binomial expansion) describes the algebraic expansion of powers of a binomial. According to the theorem, it is possible to expand the polynomial $(x+y)^n$ into a sum involving terms of the form ax^by^c, where the exponents b and c are nonnegative integers with b + c = n, and the coefficient a of each term is a specific positive integer depending on n and b. For example,

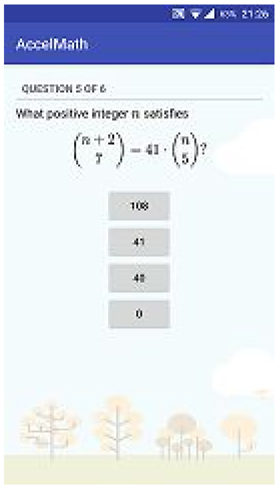
$(x+y)^4 = x^4 + 4x^3y + 6x^2y^2 + 4xy^3 +$

The coefficient a in the term of ax^by^c is known as the binomial coefficient $\binom{n}{b}$ or $\binom{n}{c}$ (the two have the same value). These coefficients for varying n and b can be arranged to form Pascal's triangle. These numbers also arise in combinatorics, where $\binom{n}{b}$ gives the number of different combinations of b elements that can be chosen from an n-element set. Sources: https://en.wikipedia.org/wiki/Binomial_theorem

START QUIZ!

G. Quiz

This is the main feature of this application. Each quiz will consist of total six questions which are randomly selected from the database. They will all be shown in proper mathematical format (the same as in every exam in university). Users answer by tapping on the multiple choice option button or typing integers number in a text field provided (only one submission will be accepted, so be careful!). The score of the quiz will be based on number of questions correctly answered (maximum is 6 = 3 stars, 4 = 2 stars and 0 = 0 stars). After each quiz is finished, users can choose whether to save the current progress or go back to lesson page.


AccelMath

QUESTION 5 OF 6

What positive integer n satisfies

$$\binom{n+2}{7} = 41 \cdot \binom{n}{5}?$$

108

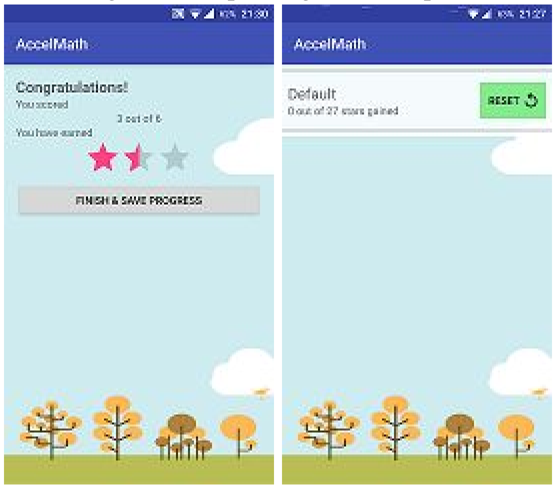
41

48

0

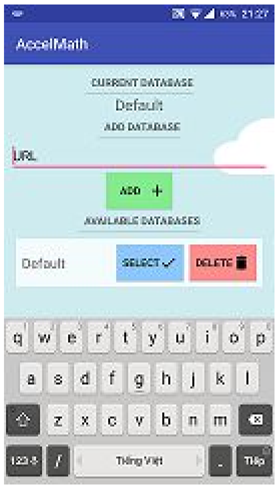
H. Progress Tracking

Access via Main Menu. Keep track of your current progress. Able to reset high score depending on user's preference.



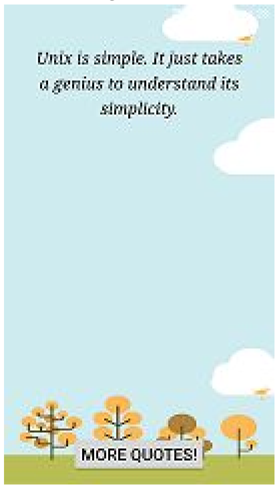
I. Settings

Users now have the possibility to add/edit questions in the database.



J. Quotes

Random inspirational quotes will be generated and shown on the screen to help players relax after long hour of work or study. This feature was added in Iteration 3 to make the app more friendly and entertaining to users (mostly CS students).


Unix is simple. It just takes a genius to understand its simplicity.
MORE QUOTES!

VII. LIMITATIONS

Our applications are limited under time constraints and budget. As with any other development problems, demand is driving every enterprise down the mobile route whether they like it or not. Issues in our development also include that our application is the development environment where we still do not support multiple platforms. Although we have a simple backend integrations, we have yet to exploit and discover more database development in a sense that we need faster and optimized rendering. As users might have noticed, it takes time for the LaTeX typeset to be rendered. It takes enough time that users do not get the full experience of using the application. This leads to another issue due to the fact that throughout the development, we used Java as a programming language. Java is not necessarily built for sophisticated memory management like C++ is. Even if we want to, optimisation is a time consuming and complex process. The renderers for LaTeX are either MathJax or KaTeX. The latter offers faster rendering while limiting the symbols that we may use. Due to fragmenting activities, we decide to adopt MathJax for stability.

It is not easy to educate the mass about Mathematics. Not only that only a few percentage of the population would be interested, but even fewer of that percentage is even bothered to touch an educational mobile application due to the size of the application's presentations. Most people would prefer learning on a "bigger screen" with their computers. This is precisely the condition to bring the need for a web-application alongside the mobile application.

To make matters worse, as demand for more sophisticated features such as randomized values in questions and demand for more topics grow, our group will continue to have a challenge to bring more developers and Mathematicians. Not only that it is hard to find, they will also be increasingly expensive to recruit. Given project constraints of having four members only, this is an impossible task. Had we had an extra member for the project, we would have accomplished much more.

VIII. CONCLUSION AND FUTURE WORKS

We have led ourselves to the conclusion that we take this project as a wonderful experience as not only developers, but to also experience what the industry ahead of us is like. Having learned much general knowledge in software development and its practices, we may apply our knowledge to potential companies in front of us.

We have also learned, collectively, that the most critical stage in any development is to familiarize ourselves with the project. "How much does the development cost?", or "is it worth the time?". or even "has anyone ever done this? if so, can I do a better job?".

In the future we plan to use user studies. We hope to further determine its usefulness to the community and to what degree the application will be beneficial to them. This is market research, after the main problem is analyzed.

To talk about the future of the system itself, it depends on the society. As mentioned earlier, AccelMath is a community driven system in a sense that we allow the community to design questions for others. This is obviously a double-edged sword model. One can use it for their own benefit to ask questions. The questions asked may or may not meet our standard, or follow improper formatting.

A versioned release of the system is already available for the public on Google PlayStore. A part of our plan, should we choose to continue this project beyond the academic scope, is that we would look at the application's analytics to determine whether we have enough users to continue development, or based on whether we would have time to do so or not. One can only do so much.

ACKNOWLEDGMENT

This project is supported by Faculty of Applied Science at Simon Fraser University. A large portion of Java code and GitLab workflow derived from lecture notes and video tutorials of Dr. Brian Fraser. We would like to show our gratitude to librarians and the staff of Faculty of Applied Science for giving us a good guideline for assignment throughout numerous consultations. We would also like to expand our deepest gratitude to all those who have directly and indirectly guided us in doing this assignment. In addition, a thank you to Professor Naz, who introduced us to the Methods in Software Development, and whose interests for teaching had lasting effect. Many people, especially our classmates and team members themselves, have made valuable comment suggestions on this proposal which gave us an inspiration to improve our assignment. We thank all the people for their help directly and indirectly in the completion of our assignment.

REFERENCES

I. Sommerville, *Software Engineering 9th edition*, Pearson publisher, 2010, ISBN:0137035152.

C. Delessio, L. Darcey, S. Conder, *Android Application Development in 24 Hours, Sams Teach Yourself*, 3th edition, Sams Publishing, 2013, ISBN: 0672334445