**MQ Algebra**

1. **Introduction**

This app will connect students with one another in small online groups and allow them to solve, discuss and study algebraic questions together.The app can generate a completely random question in algebra of a chosen topic and allow students attempt the question or see the step-by-step solution. Students will also have the ability to input specific questions to solve and discuss it with their peers.

* 1. **Application Vision**

It will allow student around the world to learn, help, and solve questions together in algebra. It is designed for specificity when seeking solutions to certain questions, and provide another medium through practical learning besides online videos and text books.

* 1. **Scope**

Devices should be able to connect to each other and users will be able to communicate with each other with instant messaging. The mathematical concepts are out of scope since it doesn’t really have much to do with android however it will contain at least 2 or more concepts.

* 1. **Document version History**

08-9-15 – Initial one-page proposal

09-8-15 – Add extensions for features to include online services

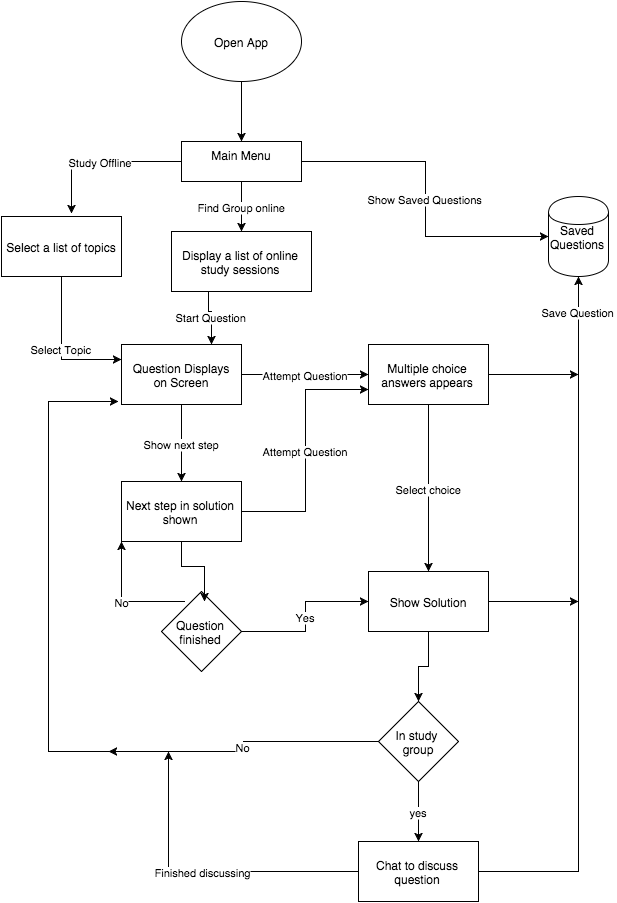
16-9-15 – Convert proposal to this type of document

23-9-15 –Finish draft of document

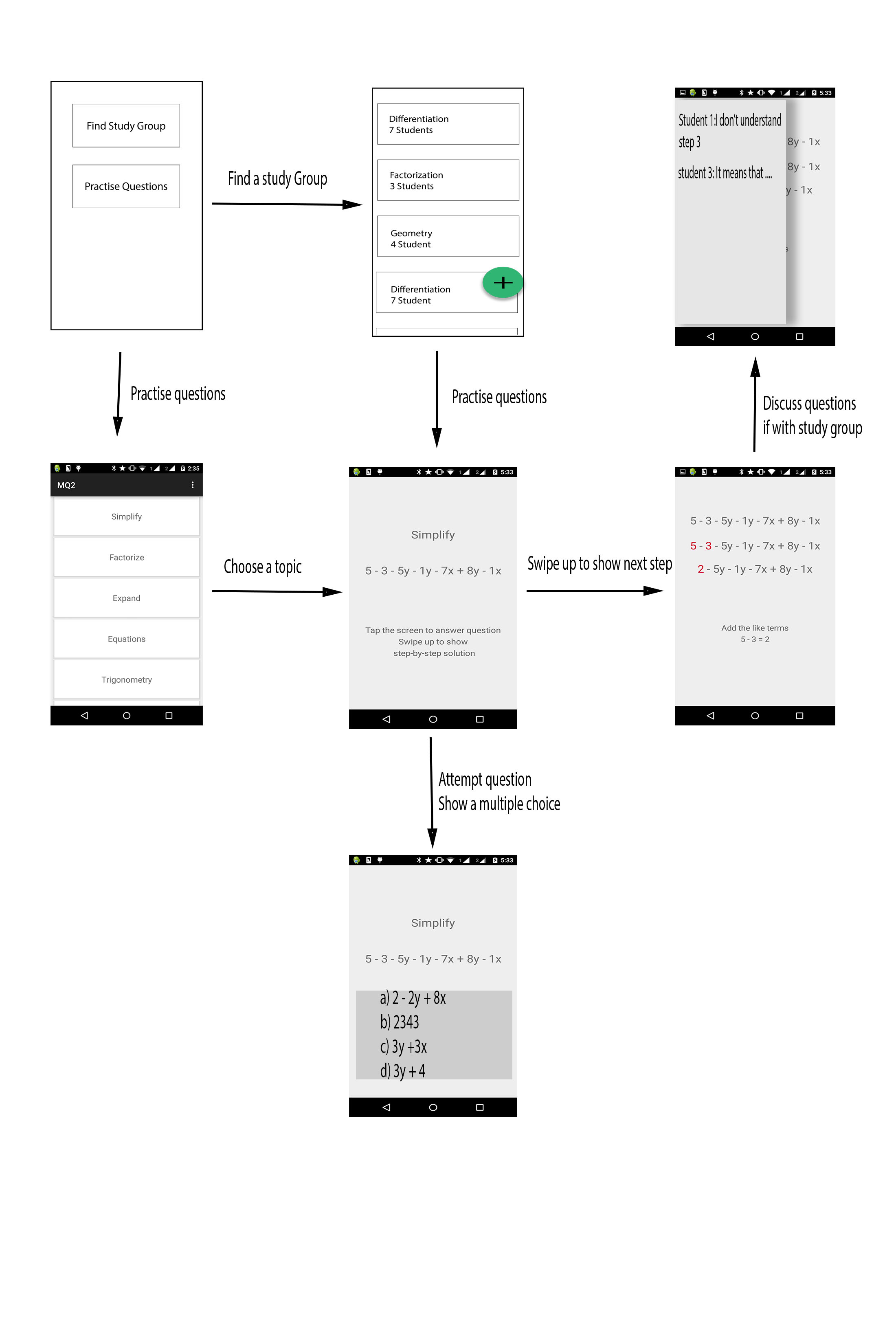
**2.1 Feature Summary**

* Connect and solve questions with other students
* Create a group for students to join and study
* Message and discuss questions with other students
* Store solutions and discussions from questions solved
* Input specific questions
* Add profile pictures and screen names
* Ratings for each student and ability to rate other students
* Generate random questions
* Solve random questions
* Give multiply choice answers
* Show explanation of each step of each solution

**2.3 Application Flow Diagram**



1. **Design**
   1. **Overall components overview**
      * The account management and instant messaging will be managed by parse. This will be the cloud for the app to communicate with other devices.
      * A service to time the user and a broadcast whether the time to solve the question is up.
      * A question activity that contains a view pager, this will show a fragment showing a type of question, and provide swiping functions for generating questions.
      * Settings functions that enables modifying the randomization of questions.
      * Database storage to store specific questions and solutions.
      * Java classes that generate and solve different types of questions.
   2. **Mockups**



* 1. **Used Application Programming Interfaces**

**Butterknife View Injection library –**

View binding to create simpler and readable code. Used to create cleaner view holders in adapters.

https://github.com/JakeWharton/butterknife

**Awesome-android-UI** –

For creating material design layouts

https://github.com/wasabeef/awesome-android-ui

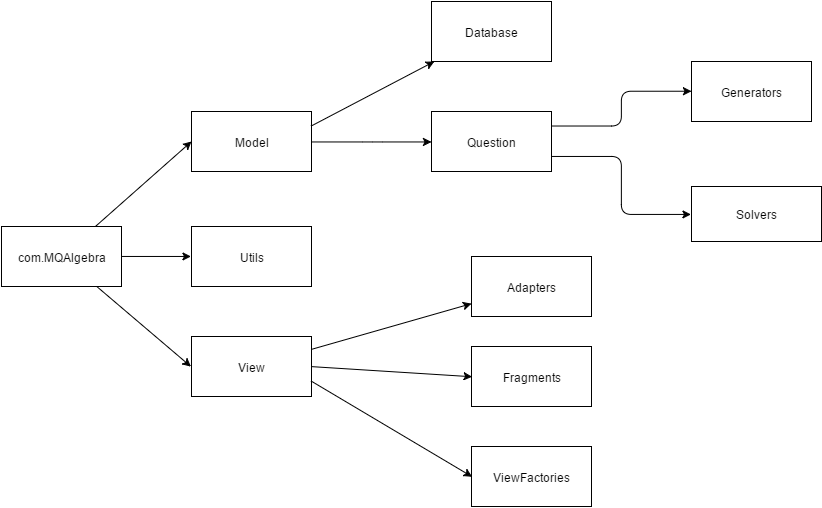
**Google Cloud Messaging** – Cloud service for enabling

https://github.com/google/gcm

**Parse –** To enable user accounts for logging in

https://github.com/ParsePlatform

1. **Application Structure**
   1. **Packages**



**Model** – contains the data of the question and solutions

**Database** – where certain questions are stored

**Question** – interface for questions

**Generators** – generates a type of question related to a topic

**Solvers** – Solves a question and outputs a solution

**Utils** – Contains constants, static methods and classes

**View** – Contains the folders adapters, fragments, and view factories

**Adapters** – populates the listview, and chatbox

**Fragments** – contains UI for specific types of topics

**ViewFactories** – Enables the view switching for TextSwitchers and ViewPagers

* 1. **Activities**

**QuestionActivity** – contains a view pager which displays different fragments according to topic selected

**TopicsActivity** – Shows a list of topic available

**LoginActivity** – allows the user to log into an account

**RegisterActivity** – Create a new account for the user

**SettingsActivity** – Allows users to change settings of their questions

**SessionActivity** – finds online groups with other students

* 1. **Services**

**StopWatchActivity –** Allows time to be display to solve a question

* 1. **Others**

**Question –** Interface for each question type, contains generate and solve method

1. **Data Structure**

The student accounts and session will be stored in parse, the information include will be:

* Username
* Password

The for questions saved, the table will include columns

* \_id
* questions
* comments\_id

The chat messages for questions will be saved as

* \_id
* Username
* Comment
* Order

1. **Testing**

**Save a question**

Pre-condition: user stores a question

Post-condition: user can view questions stored in database

**Join online study group**

Pre-condition: select a study group

Post-condition: same questions display on all devices

**Registration**

Pre-condition: user has no account

Post-condition: user can log in

**Instant Messaging**

Pre-condition: user types in message

Post-condition: message displays in all devices

1. **Conclusion**
2. **References**