Smart Kids Application to Develop the Potential Interests for Children

by

Guorui Li

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
May 5, 2018

Keywords: E-learning, Mobile Education,
Android Application, Web Development, Recommendation System

Approved by

Dr. Cheryl D. Seals, Chair, Associate Professor of Computer Science and Software Engineering
Dr. Richard Chapman, Associate Professor of Computer Science and Software Engineering
Dr. James Cross, Associate Professor of Computer Science and Software Engineering

Abstract

Children today are growing up in a rapidly developing world filled with quickly evolving technologies, such as the Internet, smartphones, and smart mobile devices, etc. As a result, more applications designed with children as the targeted population, and young students have more ways to obtain educational resources online. This work will discuss an educational app for children who want to find out more about their interests or help them to explore their interests in different topics such as science and health from a youth perspective.

The proposed project will be implemented as a mobile application and specifically an android application based on a web crawler system and a recommendation system. The system will collect data from specific organizations, which have educational articles aimed at children, such as extension.org. Also, we have collected requirements, designed and developed a user-friendly Android application for teenagers (i.e., based on user feedback) and students can utilize this app to access exciting topics (e.g., youth science). We aim to improve user experience in the area of STEM content access. Based on the student interaction with the application, we will refine the design of our recommendation system to analyze student interests and provide comparative data to support the targeted population in quickly finding information in their fields of interest.

Acknowledgements

From the second semester in Auburn University, based on the application demo IW2K which my team designed in the HCI class, I begin to build the idea to develop this project to improve my skill level and to strengthen myself to become a full stack engineer. When I talked about this to my advisor Dr. Seals, she set a high value on this idea and hiring me as her research assistant to give me a chance to let this project become true. I would love to extend my sincere gratitude to Dr. Seals for her instructive advice and useful suggestions during the process of this thesis. Also, I would like to express my genuine thanks to the other committee members Dr. Richard Chapman and Dr. James Cross. Thanks for your time and participation in this project.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

As the Internet technology is developing, huge resources and data become accessible to individuals, which includes news, videos, market goods and so on. Some new technologies may be inconspicuous to the common consumer, but in support of learning the development of new educational resources is critical. Educational resources and tools may not be as popular as the news and market goods, which are human's daily necessities, they are playing an essential role to the developing of human beings because they have a massive influence on the new generation, the kids. As a result, the usage of online educational resources is of critical importance, and it is a valuable thing for us to investigate.

As the popularization of mobile devices, such as the mobile smartphone, portable pad, smartwatch, people's lives become more convenient and efficient. By using these devices, we can request a taxi and pay service fees, we can order the goods we need from Amazon, and we can watch movies or have face to face video to our friends. All these events own the motility of the smart mobile devices. As a result, mobile learning(M-Learning) become a new field, which can help people learn anywhere and anytime.

The big data and data mining technology is one of the most popular techniques nowadays and is developing very quickly. It constructs an automatic factory, which provides the service of collecting users' data, analysis and produce an important result to improve the service. By using data mining, Amazon analysis users' purchase habits and provide a specific commodity. By using data mining, Netflix provides users with interesting video according to analysis users' viewing history. Similarly, we can use big data to M-Learning field to let the learning processing become more enjoyable and develop the potential interests of the young kids. The Internet provides the

enormous educational resource, smart mobile devices provides the efficient motility, and big data provides the technical solution, it is the exciting era to build the new educational platform, which giving enjoyment and efficiency learning process and develop the greatest extent of potential interests of the kids.

In this thesis, we integrate three technologies, which include the Internet, mobile devices, and big data into our system to build a new platform called Smart Kids. We collect the specific resources from the Internet, display them to the individual users and provide recommendation resource according to the data we receive from the client application. Chapter 2 consists of literature reviews, which includes special requirements and principles for the development of children-oriented software, the development process of software for children, the interaction between children and application and recommendation system used in the software system. In Chapter 3, we discuss the existing problem and proposed solution. Chapter 4 goes over the detailed implementation of the application. In Chapter 5, we cover the evaluation studies and the corresponding results. Chapter 6 describe the conclusion of the project along with the future work to come.

Chapter 2. Literature Review

2.1 Requirements and principles of developing child-oriented software

Mobile Learning as a Method of Ubiquitous Learning: Students' Attitudes, Readiness, and Possible Barriers to Implementation in Higher Education

Alhassan uses three methods, which are descriptive analytical methodology, questionnaire and personal interviews to explore the attitudes of readiness, the level of readiness and the possible barriers to implementing mobile learning. After analyzing the data collected from the methods, he concludes that students have high desire to learn using mobile technology outside the classroom; most students have enough technical and financial readiness to use the mobile learning services; students have only a little experience in mobile learning. There are five advantages, which are the availability of facilities, increasing motivation and interaction among student, preparing them for technical future, enhancing the effectiveness of learning, and the high reputation of the university for using this advanced learning method. There are some disadvantages: the current state of mobile technologies is not suitable for higher education; the interaction between students and teachers through those technologies do not seem natural or human such as that in the same room, and faculty members may use mobile learning technologies weekly. The potential barriers are costly, the students or faculty members' opposition towards the use of mobile learning style that the university may face, the lack of readiness and unsuitability of the existing infrastructure.

Mobile Learning Environment System (MLES): The Case of Android-based Learning Application on Undergraduates' Learning

Fahri and Samsudin take a survey to measure undergraduates' perception of mobile learning used throughout a semester long. They divide 56 students into two different groups. The first group uses the e-learning of the university and the second group uses their mobile phones, running to gain access to learning contents. The questionnaires are prepared in Google doc form, which includes four sections: respondents' demographic, android learning environment system, e-Learning environment provided by the university and the effectiveness of the android learning system. The result reveals that students used mobile learning system are more receptive. The Android mobile phones are quite affordable to won and regarding performance and mobile learning is much favored for this feature of a digital learning system. Finally, they conclude that mobile learning system costs efficiently and can quite easy to implement. Newer, better versions of Android OS continually update, giving better performance regarding processing and intuitive interface design. The participants of the study reported that the application was more fun, interactive and intuitive to use as a mobile learning system, which can help students manage self-paced learning.

Android Based Smart Learning and Attendance Management System

Joshi and Shete take a survey of different attendance tracking systems and find they have disadvantages, such as manual attendance system is inefficient, RFID with object counter is time-consuming and unsafe and Bluetooth based attendance system needs student's phone for attendance. They implement an Android-based smart learning and attendance management system which is less time to consume, safe and easy to use. By using this system, students can learn anywhere and anytime, timely updates can send to students and their parents, attendance marking, and report generation becomes easy because of fewer chances of malfunctioning.

The Usability of Mobile Applications for Pre-schoolers

Masood and Thigambaram point that the adult-oriented software UI design principles are different compared to the children. “Developers and designers usually create UI for application based on their experience, the guideline of the client or company and their intuition, which may be due to the lack of awareness or practice in usability, UCD or HCI field. The application fails because those children feel confused if the application design does not match their mental model.” To understand whether the mobile educational application (MEA) user interface design is usable for the children and identifying the design elements and principles to consider proposing children’s User Interface design guidelines to assist software developers and designers, the author uses Guilford's structure of the Intellect framework to collect data and analyze. The data includes observation, evaluation, and data from the SensoMotoric Instrument eye tracking glasses according to the performance of the activities in which groups of children involved. They analyze the intuitiveness and efficiency of Mobile Educational Application, and also the children’s user interface design proposed guidelines. Finally, they conclude that UCD knowledge and skills are needed to design and develop a useful and usable educational application, and the Education Software companies also need to create the urgency and awareness to integrate UCD methodology in the project lifecycle.

User-centered Interaction Design of a Mobile Learning Platform for the Generation 60 +

There is a prejudice that older people could not find accesses to sophisticated mobile applications. However, in this article, Norbert Reithinger, Aaron Russ, and Kinga Schumacher prove it is incorrect by using user-centered interaction design of a mobile learning platform for older people. They used a series of user tests for the interactive platform during the development process. 25

older persons from targeted user groups involved the first usability test, and 50 older users took part in the second test. Some of them did not have computer or smartphone experience, and some have MCI (mild cognitive impairment). During the iterative tests, researchers revised the style-guide and recorded the operations of these participants. According to these data, they concluded that if bringing design experience, methodology and tested style-guide during the process of design and development, older people would use complex application efficiently and there was a way to obtain the requirements of older people.

A study of the interface usability issues of mobile learning applications for smart phones from the users' perspective

A successful M-learning application should be easy to use, learnable, understandable and attractive. Enjoyable experience and meeting usability needs for users are also critical to this kind of app. There are little instructions or methods to evaluate the usability assessment of an M-learning application. Alrasheedi and his colleague developed a conceptual framework to achieve this goal. Also, a prototype application for smartphones was developed by using Java program and Android SDK as a guideline of their framework. They used questionnaire survey to analyze the collected data and concluded that their framework could be worked as a guideline to help design and develop M-learning applications. However, due to limited demographic during the research, the results cannot be used as universal.

2.2 Development process of software for children

Research and design issues concerning the development of educational software for children

Cynthia used three different types of software to engage children in using computers to find out knowledge. To evaluate the effectiveness of these three types of software, he examined three distinct aspects of the materials: comprehensibility, appeal, and usability. Cynthia used seven different types of measures, which included classroom observations, student and teacher interviews, written forms, login books, evaluation forms, teacher background information forms and the science and mathematics instruction. With these measures, he found quantitative and qualitative differences. Cynthia also found two factors that influenced the amount and the way software utilized in classrooms. He concluded that well-designed software could not only help teachers improve their teaching process, but also let students obtain friendly methods and materials to learn happily and efficiently.

JamMo: Developmentally designed software for children's mobile music-making

Paananen and Myllykoski described the features of JamMo software, gave some examples of the upcoming games, and tested with 3-12-year-old children. The study included usability tests and impacted analysis of the effects of JamMo on social inclusion of children. They listed the essential features that make JamMo an innovative mobile music pedagogic tool: user sensitivity, pedagogical structure, social sharing, mobile music classrooms, integration and research tool. Also, they concluded JamMo provided a new tool for cognitive and social developmental research tool in the domain of music.

Evaluating Usability, Fun and Learning in Educational Software for Children

Matthew Horton and his team members discussed the relationships between usability, fun and finding out educational software for children. He did some experiments to obtain four conclusions. There is the scholarly benefit of utilizing the software. Usability does matter to children, so getting it right should give priority to designers and manufacturers. Children's preference is for fun in software. Children as young as 7-8 were in a position to distinguish between concepts such as usability, fun, and potential for learning.

Developing Mobile Applications for Children

Color and text play an essential role to overcome a child's limitations when developing a mobile application for children. Colors can invoke powerful emotions and memories. When bright colors are used and put in contrast with each other children can see the difference between them, and this creates a more vivid world to explore. The text is vital that it is as effortless to read the documents used as possible and it is, therefore, a good idea to use sans serif fonts because of their more straightforward forms. Shapes and symbols: icons within a program are different from each other so that we can avoid confusions. It is better to use round rather than edged shapes because rounded ones come off as less imposing and sharp edges or sharp geometric pattern can cause intense discomfort to people with neurological disorders. The splash screen: children cannot understand loading page, and they do not have the patience to wait, so it is better to have some exciting things in the loading page, such as animations and music. If the setting menu is necessary, it does not obtain too significant an impact on the application. Sizing interactive elements: elements such as buttons, they must be accessible to found and large enough for children to hit. Advanced technical

features on the mobile devices, such as a camera, microphone, vibrating sensor, all these functions can be utilized to varying success.

Children in the information technology design process: A review of theories and their applications

According to the participatory design theory, users are the best qualified to determine how to improve their work and that their perceptions about technology are as important as technical specifications. Including children as a design partner, however, may have some difficulties. To improve the creativity, usability, and profitability, children can be involved as informants at salient points throughout the design process.

2.3 Interaction between children and computers

Search User Interface Design for Children: Challenges and Solutions

Based on Erickson's theory of psychosocial development, children require emotional support and a feeling of success. The idea here is to provide children with enough help to support their search process to avoid frustration, which can be achieved by proper guidance. The researchers designed a guidance avatar for emotional support, a treasure chest for memory support, a pie menu for language support and a cover flow to support the judgment of result relevancy.

How Interface Design and Search Strategy Influence Children's Search Performance and Evaluations

There is a trend in digital media for children to design digital products which are "cool" and "playful." The authors identified three aspects. The first is classical interface design with a classical interaction style and without playful graphics. The second is interface design with playful graphics

with a classical interaction style, and the third one is playful interface design with playful graphics and a playful interaction style. The researchers explored the effects of children's use of a search engine on children's search performance and effective responses. It turns out that the type of interface design did not have any effect on the performance when children are searching. However, it proves that the feelings of emotional valence and the evaluation of "goodness" were influenced. The researchers also find that children's search performance was much more effective and efficient when using the search engine.

A review on user interface design principles to increase software usability for users with less computer literacy

According to Ali Darejeh et al., several user-interface-design principles are extracted for users with less computer literacy. For children users, designers should consider the particular issues and derive solutions for those issues. Because of the limitation of understanding the abstract concepts and reading, the user interface should be easy to interact, and it is necessary to reduce the amount of or eliminate text and replace it with a simple picture. As the limitation for cognitive, it is better to provide a visual tutorial, and the software should have a clear and navigation structure.

Children aimed interfaces for android running devices

A developer should be aware of several particularities. First, an animated character is the main point of interest at the interface. Second, a text-to-speech engine is helpful for children to improve their learning process. Thirdly, a simple interface will enhance the usability of an application.

Design Principles for Children's Technology

According to Sonia Chiasson et al., children's development can be categorized into three main areas: cognitive, physical and social/emotion. Since children may have a problem with reading and spelling, help menu and related interface should not base on the text. Children expect to see the feedback of their actions immediately. Thus, visual feedback is helpful in the design. Children like tangible interfaces where they can physically touch and manipulate the devices. For educational software/application, occasionally entertaining diversions keep children engagement during the learning tasks. Animated pedagogical agents are useful for learning environments.

Websites for children - search strategies and interface design

This report is talking about ways to recognize and evaluation of search engines for children and the search engines interfaces design. The children's digital information search is to learn more about how children think and how digital search environments should design for children to support them search information efficiently. The design of informational interfaces for children is about exploring effects on performance and compelling experience.

Tangible User Interface for Children - An Overview

According to Diana Xu, most interfaces have relied on a limited variety of physical objects and a limited range of our abilities, so it blocks many of our natural abilities and forces complexity upon us. Tangible User Interfaces (TUIs) is something like "graspable" interfaces; it provides a physical form to digital information. Therefore, it assists a lot in Children's learning environment.

2.4 Privacy requirements of applications for children

Mobile Applications for Kids-Current Privacy Disclosures are Disappointing

To better understand and evaluate the emerging app market and the products and services it offers to children and to better protect children's privacy, Federal Trade Commission staff designed and surveyed the apps provided for children in the two most significant US app stores, the Android Market and the Apple App Store. Researchers searched the two app stores using keyword "kids," and put data collected into charts. The data includes the types, prices, data collection and sharing practices. The report highlights the lack of information available to parents before downloading mobile apps for their children, calls on industry to provide greater transparency about their data practices. Besides, parents should also be able to learn about the apps before downloading them.

Mobile Apps for Kids-Disclosures Still Not Making the Grade

Based on the survey about apps' children privacy disclosures, the Federal Trade Commission conducted a follow-up survey about whether the apps included interactive features or shared kids' information with third parties without disclosing these facts to parents. In the first survey, researchers used "kids" as the keyword to search apps in Apple App Store and Google Play Store. They randomly selected 200 from each store and carefully reviewed the apps' disclosures, looked at the disclosures and links on each app's promotion page, the app developer's website, and within the app itself. After that, researchers organized the data collected using statistic methods. The result indicated little or no improvement in the disclosures made and, worse, a significant discrepancy between the privacy disclosures and the actual practices of the surveyed apps.

Chapter 3. Project

3.1 Comparison of Previous Versions

This Android application is the extensional version of the previous four releases. The first three versions are all called IW2K, which based on the website extension.org. They are all android applications with static data which hardcoded in the local system. The first version was designed and implemented by Kalyan Bobbili which fulfilled the primary requirements. The second version was created by other two members and me when we took HCI class. We worked together designed a cartoon style app. Qian designed the third version. She integrated the first version and the second-cartoon style version.

3.1.1 Comparison with the 1st Versions

Kalyan Bobbili designed the second version based on Android system. He divided the category into four categories, which include science, health, responsibility and serve. And by using a sidebar, users could navigate each type. Also, there was a search bar to search the content from the website. The following figures (Figure 1 and figure 2) display the pages of this version.

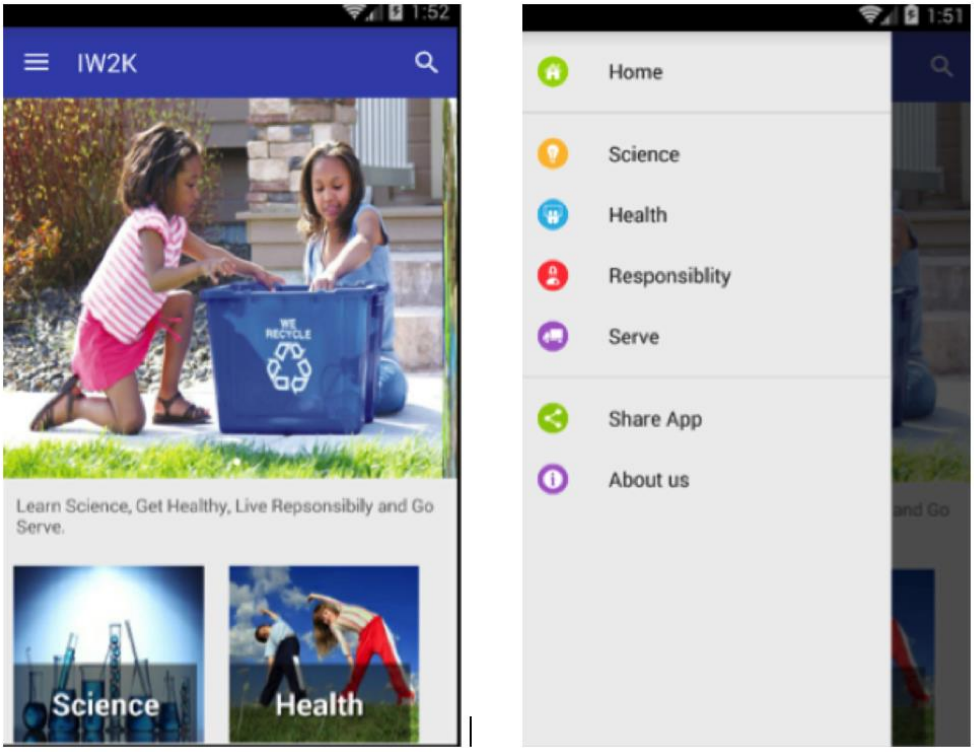


Figure 1 Version1 - Home Page and Sidebar

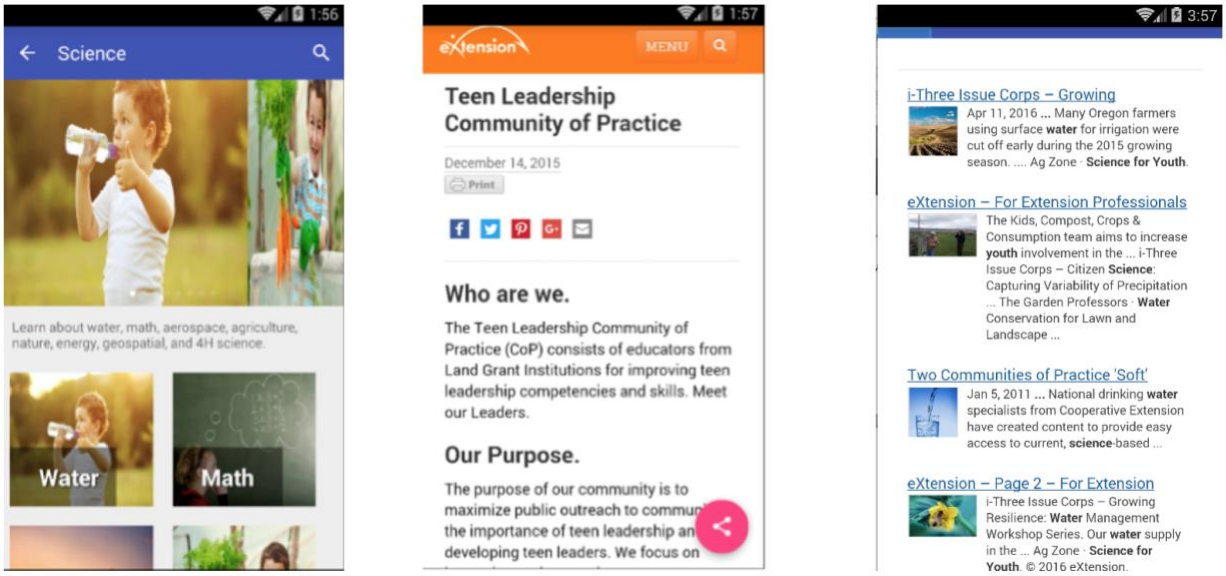


Figure 2 Version1 - Sub category, Article, and Search result

This version implemented most of the functional requirements. However, some features need to be improved. First, the article page does not provide any way for users to return to the previous

page. We analyzed the implementation details and found that though it used the WebView as a container to display the web page, it did not provide a friendly method for end users to return to the previous page (Figure 2 Article page). Secondly, the result page of search results has a toolbar overlapped by the phone's status bar. Users could not return to the previous page (Figure 2 Search result page). In the new version we designed, we would be able to address these weaknesses.

3.1.2 Comparison with the 2nd Versions

FYFL version 2 was created as an alternative view as a class project in Human-Computer Interaction (I was a member of that team). We considered that the end users would be young kids. Therefore, we designed this app with more user-friendly UIs based on the cartoon style. For each category, we created a cartoon-style icon to represent the content. We used a hamburger-style menu to navigate the main page. The figures below show this cartoon version.

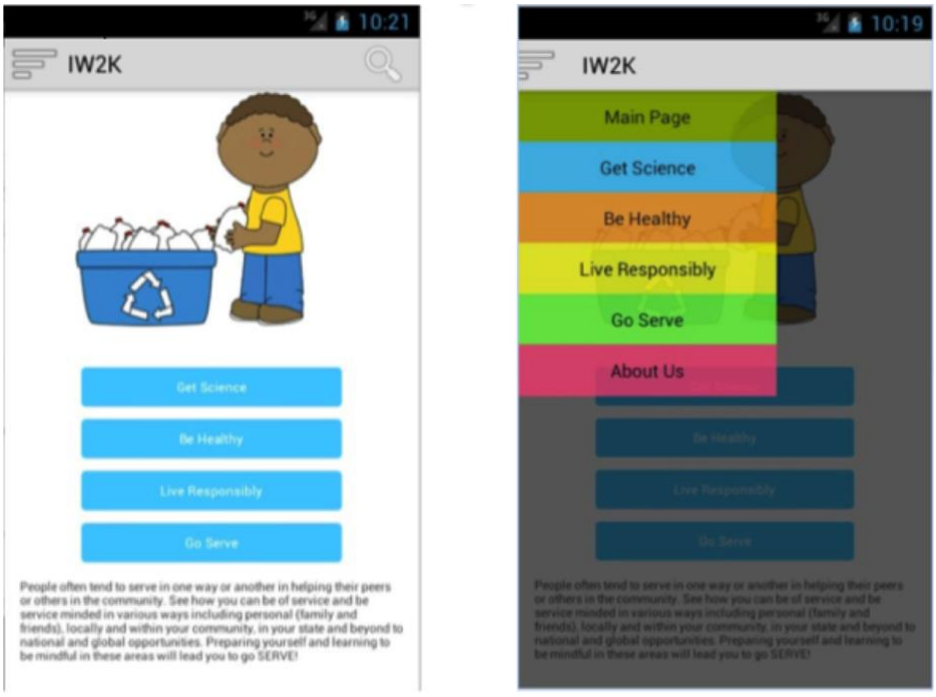


Figure 3 Version2 Home and menu page

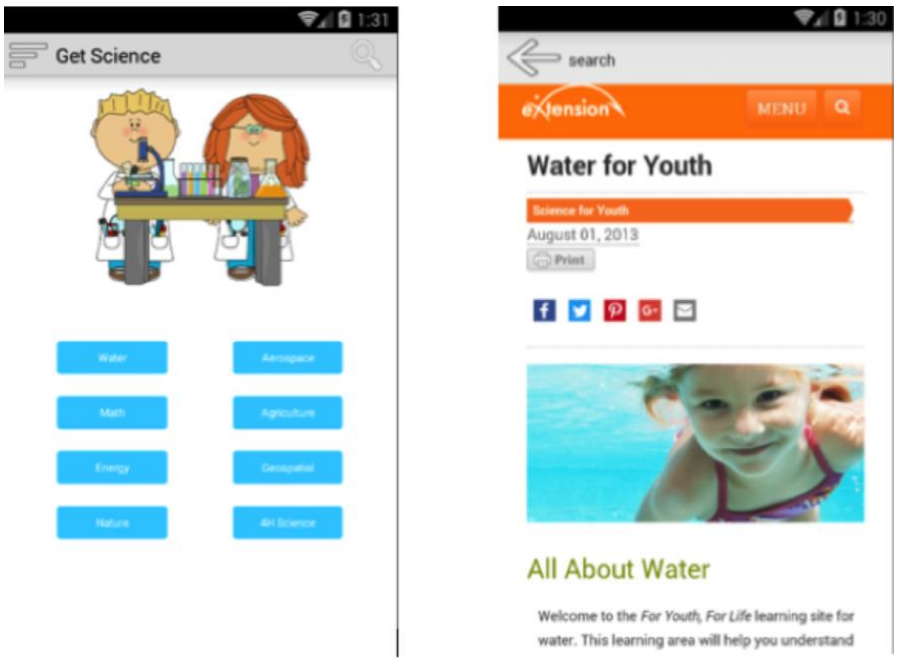


Figure 4 Version2 Sub category and article page

This version finished most functions as well. We considered more about the end users. The whole design was more user-friendly and more appealing to particular users like young kids. However, we had several defects in the application. First, when we tried to search keywords from the search bar, sometimes they did not respond appropriately. We analyzed the URL when this search function was executed and found some URLs did not have response content, which caused the blank page. Second, when users clicked on the details page and wanted to go back from the details, the back button did not respond. We hypothesized that we did not set the click listener when we implemented the function. It also reflected that we did not process a strict testing process about the application, based on the short development cycle. We would be able to pay more attention to the testing method to avoid these defects.

3.1.3 Comparison with the 3rd Versions

Based on the two Android versions, the client revised the requirements to create a version that supported both styles. Qian developed the fourth version, which had a setting button to change the theme of the application. Figures below show some of the pages.

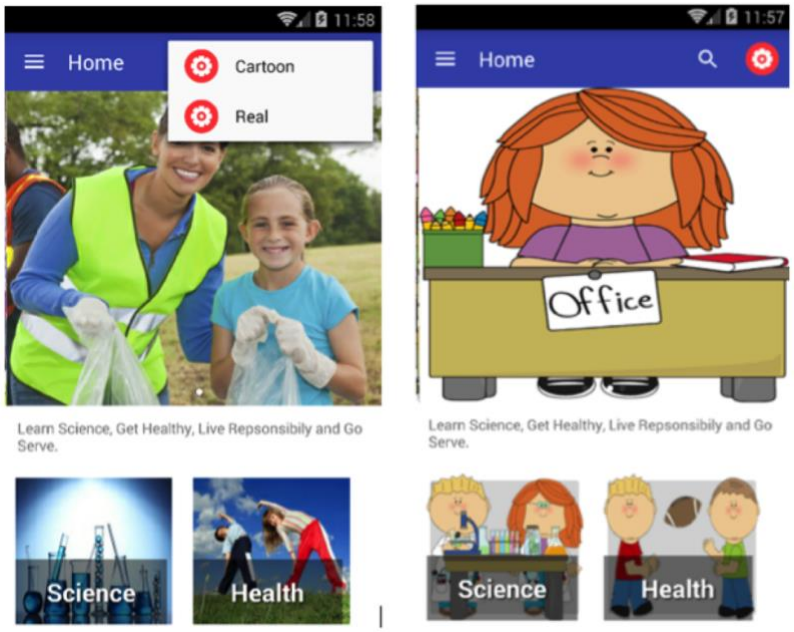


Figure 5 Version3 Real style and Cartoon style

Qian integrated two themes of the first two version into the third android app, which let the users have multiple choices. However, it was only an integration, and it was not a significant improvement, but just satisfied the user request. The three previous versions all separated the articles into four categories include science, health, responsibility and serve. There were also several subcategories in each category list with articles for users learning and reading. All the versions implemented a search function, which provides a search function for items from the website extension.org. However, the contents of the main pages are all static articles, which are hardcoded in the project. They are more like a static demo to clarify the requirements, but not robust dynamic application.

As an educational application for children, besides the user-friendly design, the app actually will work as a kind of information system, which needs dynamic contents to provide users with useful data. Also, we aim for this application to help children find their potential science, health, service and responsibility interests. So, the users' behaviors are significant to help us to analyze them and achieve this purpose. In this case, we need more strong control of the article resource, such as the dynamic article resources and diverse categories. We need more information about users, such as their basic information, which includes age, primary interests, and grade level. We also need the data of the users' behaviors, such as what articles they have read, what articles they mark as interested or collected. With the diversity of contents and necessary information of users, we need to obtain the behavior information based on the interaction between users and application and establish the relationship between the articles and users. Based on all the data above, we need to design and implement a recommendation system to provide users with useful articles to help them find their potential interests.

The previous three versions contained some unstable functions and defects, which showed that they did not follow a strict testing methodology when testing their applications and client changing requirements caused scheduling problems at the end of development cycles. We plan to give more attention to software process including test methods to keep our application high in quality and reliable in performance.

3.2 Hypotheses

We will evaluate our new application with four aspects which are aesthetics, effectiveness, ease of use, feature, satisfaction, and efficiency. We will compare our new design with the previous versions of this application at the end of this research study. We will provide usability questions

to process the usability. The application developing experts and test participants will take part in the process and test the hypotheses.

Aesthetics

The new application design will have a new clean layout and framework, which will give the users a feeling that the system is aesthetically pleasing.

Effectiveness

The new application will provide the effective user interactions for users. We will dynamically collect articles from the Internet and design a new backend system to provide dynamic data responses for the end users.

Efficiency

We will implement in the application based on Android 5.0 SDK, which has more features. We will use multiple test methods to test our whole system.

Ease of use

The new application will be easy to use. We will put all the usability functions in one frame which will help the UI looks user-friendly. Users could get touch the primary tasks in one step.

Features

Compared with the old versions, the new application will have dynamic data, which will be the most important new features for users. Dynamic data will be useful to help young kids to find their potential interests.

Satisfaction

For the aim of providing the potential interesting articles, our new application will have the log collection function, which will collect users' data, such as users' reading history and users favorites articles. Most of the end users will have the positive attitude to this function. They will be more acceptable to the new feature. Someone who cares about privacy may refuse it. However, with the process of using it, they may become more acceptable.

3.3 Requirements

For the whole project including three systems, we separate the requirements into three different categories.

3.3.1 Requirements of Crawler system

1. The crawler system should have the ability to collect useful data from specific websites. For the first version, we only collect valuable articles from extension.org. For the future version, it should be able to adapt to other sites in the same mechanism.

2. This crawler system should have the function to divide the essential data into specific categories. In this project, we define ten types which include art, biology, chemistry, education, geography, history, math, physics, psychology.

3. The crawler system should have the function to store the divided data into the database according to the category.

3.3.2 Requirements of Android application

1. The application should have the functions of user register, user login and sign out.

2. The application should have the function for the user updating specific profile.

3. The application should have the function presenting the common articles topics.

4. Users could select specific topics to read the articles.

5. Users could search specific articles according to the keywords.

6. The application should provide top five keywords for user to search.

7. The application should have the function to record the users reading history and user interaction such as users mark the interesting article.

8. The application should have the function to upload the log file which includes user's reading history and user's marking favorite articles.

9. The application should have the function to delete the reading history.

10. The application should have the function to unmark the articles they marked previously.

11. The application should have the function to present recommendation articles from server.

3.3.3 Requirements of Server system

1. The server system should provide APIs for user register, user login and sign out.

2. The server system should provide APIs for user updating specific profile.

3. The server system should have the function to receive the log file from the Android app.

4. The server system should have the function to analyze the data from the log file and provide the recommendation articles for specific users.

Chapter 4. Method

4.1 The Problem

4.1.1 How to collect the original educational resource?

Though there are lots of available resource from the Internet, it is a difficult task to deals with the thousands of websites because the different websites have their individual organized structures. Where could we find a more relevant useful original educational resource and how to use them?

4.1.2 How to display the processed data to users?

If we have collected the particular data, how could we display them to our users? By using the website or mobile applications? If we choose a mobile app, how could we design to adjust to users' habit and which is the most useful and efficient way to display the data to users? The same question with choosing web application.

4.1.3 How to collect users' data, analysis and response to users?

Based on the data, how to design and implement a particular recommendation system to adjust to our system? For there are many different experimental methods, such as offline experiment, user study and online experiment, which one should we choose to integrate into our system?

4.2 Proposed Solution

To solve the problems listed above, we designed the whole framework for our system, which includes a crawler layer, a server layer, and a front-end user layer. The crawler layer is responsible for collecting and processing data from different kinds of websites which could provide useful articles. The server layer has three primary purposes. Firstly, it provides the data we collect from the crawler layer; the data is formatted in the form of Restful API. Secondly, the server layer plays a role as a receiver, to receive the log data from front-end users which include the users' basic

information and users' interactions with the applications. We call these data like users' data. Thirdly, based on the users' data we also have a recommendation system in the server to analyze users' data and calculate the result which provides special users with special articles. The front-end applications are responsible for displaying processed data and recording and sending users' data to server layer. The figure below represents the roles of the three layers.

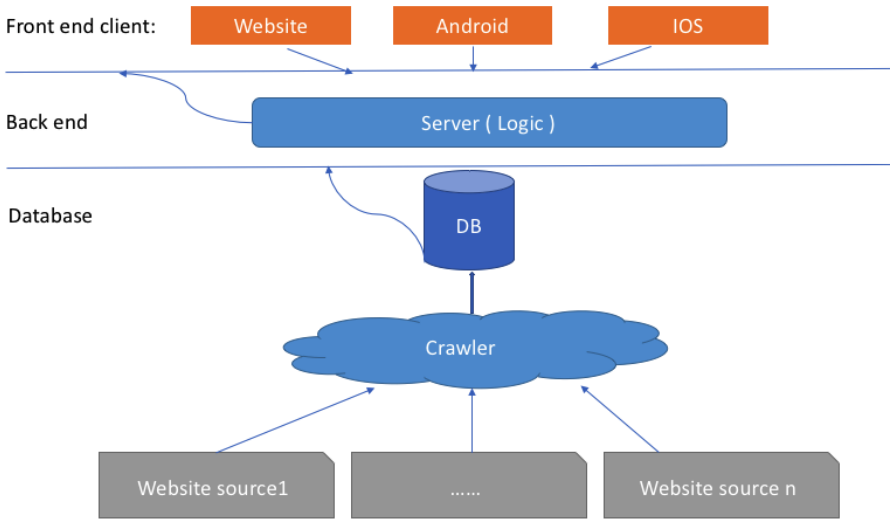


Figure 6 System Framework

4.3 Tools and Technologies

4.3.1 Crawler system

We used CasperJS, Java, and Jsoup in the crawler system. We write the function by using CasperJs to collect the specific data from the target website such as extension.org. The crawler function image below shows the principal function.

```
function collect() {
    var casper = require("casper").create({
        waitTimeout : 10000,
        stepTimeout : 10000,
        verbose : true,
        pageSettings : {
            webSecurityEnabled : false
        },
        onWaitTimeout : function() {
            this.echo('** Wait-TimeOut **');
        },
        onStepTimeout : function() {
            this.echo('** Step-TimeOut **');
        }
    });

    casper.start();
    casper.open(casper.cli.args[0]);
    casper.then(function() {
        fs.write(casper.cli.args[1], this.getPageContent(), 'w');
        console.log("Completed: ".concat(casper.cli.args[1]));
    });

    casper.run();
}
```

Figure 7 Crawler function

By using this third-party JS library, we also could simulate the actions such as click the button for next page and type in the information into the Text-Field in the HTML web pages and then log in to the website.

We wrote a java application to create bash file to run the CasperJs script file.

```
    /**
     * Create the bash file according to the keyword,
     * create a file named 'keyword.sh' in the sh_file directory.
     *
     * @param keyword
     */
    public static void createBashFile(String keyword) {
        File file = null;
        FileWriter fw = null;
        file = new File(PATH, keyword + ".sh");
        if (!file.exists()) {
            file.getParentFile().mkdirs();
            try {
                file.createNewFile();
                fw = new FileWriter(file);
                for (int i = 1; i < COUNT; i++) {
                    StringBuilder sb = new StringBuilder();
                    sb.append("casperjs load.js ").append(getUrlByKeyWord(keyword)).append(URL_POST).append(i)
                            .append("\"").append(" ").append(keyword).append(i).append(".html")
                            .append(System.getProperty("line.separator"));
                    System.out.println(sb.toString());
                    fw.write(sb.toString());
                }
                fw.flush();
            } catch (IOException e) {
                e.printStackTrace();
            } finally {
                try {
                    fw.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
```

Figure 8 Bash file generator

After collected the web pages we need to parser these HTML files to extract the particular data we need. We choose the Jsoup to achieve this objective. The primary function shows as below.

```
96    static void parserHtml(String filePath, int category, String source) {
97        File in = new File(filePath); Connection conn = null;
98        try {
99            conn = JDBCUtil.getConnection();
100           Document doc = Jsoup.parse(in, "UTF-8", "");
101           Elements elements = doc.select("table.gsc-table-result");
102           for (int i = 0; i < elements.size(); i++) {
103               Element e1 = elements.get(i).select("tbody").select("tr").select("td").get(0);
104               Element e2 = elements.get(i).select("tbody").select("tr").select("td").get(1);
105               String imgUrl = e1.select("div").select("a").select("img").attr("src");// img_url
106               String title = e2.select("div").select("a").text();// title
107               String articleUrl = e2.select("div").select("a").attr("href");// url
108               String articleDescriptoin = e2.select("div").get(2).text();// description
109               if (title != null && !title.isEmpty()) {
110                   Article article = new Article();
111                   article.setArticleTitle(title);
112                   article.setArticleImg(imgUrl);
113                   article.setArticleUrl(articleUrl);
114                   article.setArticleDescription(articleDescriptoin);
115                   article.setCategoryId(category);
116                   article.setSourceId(1);
117                   StringBuilder sb = new StringBuilder();
118                   sb.append(title).append(",").append(imgUrl).append(",").append(articleDescriptoin).append(",")
119                           .append(articleUrl).append(",").append(category).append(",").append(source);
120                   String sql = "insert into article (articleTitle, articleImg, articleDescription, articleUrl, category_id, source_id) values (?,?,?,?,?,?)";
121                   PreparedStatement ps = (PreparedStatement) conn.prepareCall(sql);
122                   ps.setString(1, title);
123                   ps.setString(2, imgUrl);
124                   ps.setString(3, articleDescriptoin);
125                   ps.setString(4, articleUrl);
126                   ps.setInt(5, category);
127                   ps.setInt(6, 1);
128                   ps.executeUpdate();
129               }
130           }
131       } catch (Exception e) {
132           e.printStackTrace();
133       } finally {
134           try {
135               if(conn != null) conn.close();
136           } catch (SQLException e) {
137               e.printStackTrace();
138           }
139       }
140   }
```

Figure 9 Parser function

4.3.2 Server system

As the purpose of the server layer, we divided the server system into two separate parts. The first part was to provide Restful APIs. The second part was to collect users' data and analysis. In this project, we designed and implemented the first part.

We studied the standard MVC framework and built the server system project with the instructions. The images below show the basic model and the frame structure of our project.

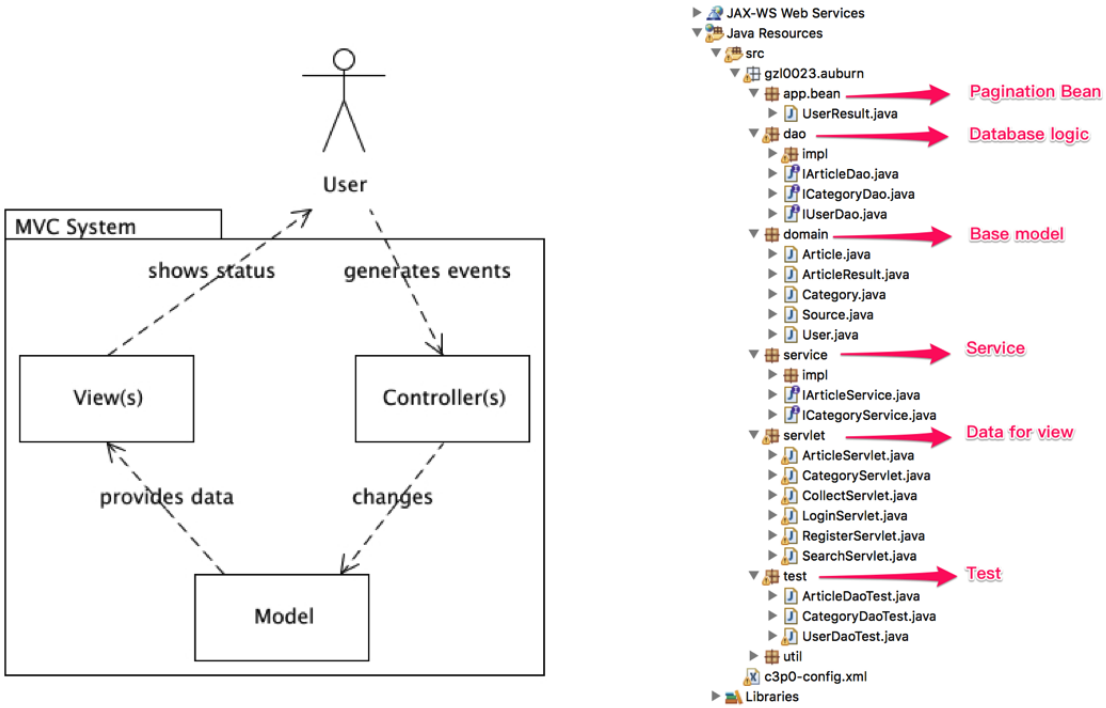


Figure 10 MVC structure and backend system structure

The package app.bean lists the pagination bean. The package dao contains the interface and the class implementation of operating database. The package domain lists the basic model in the system. The service package lists the interface and the class implementation for the basic logic API calls. The servlet package contains the implementation of HTTP response and request. The test package lists the Junit test cases.

As a sample code, we use the article, one of the most important functional parts, to demonstrate the project about our backend system.

Step1: We implemented a necessary model Article, based on the article table in the database. The following images show the database and the corresponding JavaBean.

```
+-----------------+--------------+------+-----+-------------------+----------------+
| Field           | Type         | Null | Key | Default           | Extra          |
+-----------------+--------------+------+-----+-------------------+----------------+
| articleId       | int(11)      | NO   | PRI | NULL              | auto_increment |
| articleDate     | datetime     | NO   |     | CURRENT_TIMESTAMP |                |
| articleTitle    | varchar(200) | YES  |     | NULL              |                |
| articleImg      | varchar(200) | YES  |     | NULL              |                |
| articleDescription | varchar(500) | YES  |     | NULL              |                |
| articleUrl      | varchar(200) | YES  |     | NULL              |                |
| articleCategory | varchar(50)  | YES  |     | NULL              |                |
| articleSource   | varchar(50)  | YES  |     | NULL              |                |
+-----------------+--------------+------+-----+-------------------+----------------+
8 rows in set (0.00 sec)
```

```java
public class Article {
    private int articleId;
    private long articleDate;
    private String articleTitle;
    private String articleImg;
    private String articleDescription;
    private String articleUrl;
    private int category_id;
    private int source_id;
```

Figure 11 Article and JavaBean

Step 2: With the primary Article model, we analyzed the essential logical functions and abstracted these functions as the interface. The following image shows the functions.

```java
import java.util.List;

/**
 * ArticleDao
 */
public interface IArticleDao {
    /**
     * Get article list by page and category ID
     * @param start
     * @param count
     * @param cId
     * @return
     */
    public List<Article> getArticleByPageAndCategory (int start, int count, int cId);
    /**
     * Get the total number of the articles.
     * @return
     */
    public int getTotalCount();
    /**
     * Get the total number of the certain category by CID
     * @param cId
     * @return
     */
    public int getTotalCountByCategory(int cId);
    /**
     * Get article list by key word
     * @param start
     * @param count
     * @param keywords
     * @return
     */
    public List<Article> getArticlesByKeyword(int start, int count, String keywords);
    /**
     * Get article list by the id array
     * @param ids
     * @return
     */
    public List<Article> getArticleByIds(Integer[] ids);
}
```

Figure 12 Article Dao Interface

Step 3: Based on the interface, we implemented each function. We only listed three functions as below to show the primary flow of our ideas.

```
15 public class ArticleDao implements IArticleDao {
16     @Override
17     public int getTotalCount() {
18         String sql = "select count(*) from article";
19         try {
20             QueryRunner qr = JdbcUtils.getQueryRunner();
21             Long count = qr.query(sql, new ScalarHandler<Long>());
22             return count.intValue();
23         } catch (Exception e) {
24             throw new RuntimeException(e);
25         }
26     }
27
28     @Override
29     public List<Article> getArticleByPageAndCategory(int start, int count, int cId) {
30         String sql = "select * from article where category_id = ? " + " limit ?, ?";
31         QueryRunner qr = JdbcUtils.getQueryRunner();
32         try {
33             List<Article> data = qr.query(sql, new BeanListHandler<>(Article.class), cId, start, count);
34             return data;
35         } catch (Exception e) {
36             e.printStackTrace();
37             throw new RuntimeException(e);
38         }
39     }
40
41     @Override
42     public int getTotalCountByCategory(int cId) {
43         String sql = "select count(*) from article where category_id = ?";
44         QueryRunner qr = JdbcUtils.getQueryRunner();
45         try {
46             Long count = qr.query(sql, new ScalarHandler<Long>(), cId);
47             return count.intValue();
48         } catch (Exception e) {
49             e.printStackTrace();
50             throw new RuntimeException(e);
51         }
52     }
53
```

Figure 13 Article Dao Implement

Step 4: We designed and implemented the service layer in the MVC structure. The image below shows the ArticleService implementation.

```java
public class ArticleService implements IArticleService {
    private IArticleDao dao = new ArticleDao();

    @Override
    public int getTotalCount() {
        return dao.getTotalCount();
    }

    @Override
    public List<Article> getArticleByPageAndCategory(int start, int count, int cId) {
        return dao.getArticleByPageAndCategory(start, count, cId);
    }

    @Override
    public int getTotalCountByCategory(int cId) {
        return dao.getTotalCountByCategory(cId);
    }

    @Override
    public List<Article> getArticlesByPageAndKeyWords(int start, int count, String kw) {
        return dao.getArticlesByKeyword(start, count, kw);
    }

    @Override
    public List<Article> getArticleByIds(Integer[] ids) {
        return dao.getArticleByIds(ids);
    }

}
```

Figure 14 Article Service Implement

Step 5: We implemented the ArticleServlet to provide the data by using the ArticleService.

```java
@WebServlet("/ArticleServlet")
public class ArticleServlet extends HttpServlet {
    private IArticleService service = null;
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        String start = request.getParameter("start");
        String count = request.getParameter("count");
        String categoryId = request.getParameter("categoryid");
        service = new ArticleService();
        List<Article> list = service.getArticleByPageAndCategory(Integer.parseInt(start),
                Integer.parseInt(count),
                Integer.parseInt(categoryId));
        Gson gson = new Gson();
        String jsonString = gson.toJson(list);
        response.getWriter().write(jsonString);
    }


    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        doGet(request, response);
    }

}
```

Figure 15 Article Servlet

When frontend application calls the API:

http://127.0.0.1:8080/iw2kserver/ArticleServlet?start=0&count=20&categoryid=1

It will send a request to the server and go to ArticleServlet with the parameters start = 0, count = 20 and categoryid = 1. The method called get ArticleByPageAndCategory of ArticleService will be executed to return a list of articles. The real operator is ArticleDao, who select the articles from the database and return the result. The result contains 20 articles whose index range is from 0 to 20, and their category id is 1. By using the MVC structure, we separate the functions to different levels.

We used Gson to transform the data with java class model to the data with JSON-format, which was very convenient to display the data to the front-end page. The image below shows the JSON-format data of the article list.

Figure 16 JSON format data

4.3.3 Android application

The three layers (i.e., crawler, backend server and frontend view) are separate; this gives them the ability to support different kinds of the front-end applications which include a website, Android

mobile application and IOS mobile application. In this project, we designed and implemented the Android mobile application to show our result.

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.4%
4.1.x	Jelly Bean	16	1.7%
4.2.x		17	2.6%
4.3		18	0.7%
4.4	KitKat	19	12.0%
5.0	Lollipop	21	5.4%
5.1		22	19.2%
6.0	Marshmallow	23	28.1%
7.0	Nougat	24	22.3%
7.1		25	6.2%
8.0	Oreo	26	0.8%
8.1		27	0.3%

Data collected during a 7-day period ending on February 5, 2018.
Any versions with less than 0.1% distribution are not shown.

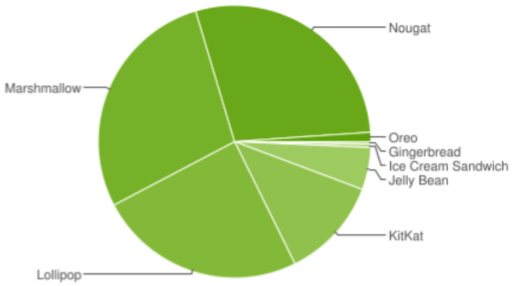


Figure 17 Android platform distribution

As the data are shown above which comes the official site of an android document, most Android devices run above the 5.0 Lollipop platform. The percentage is nearly 82.3%. This data indicates that if we develop the application based on Android 5.0, it will support most kinds of mobile devices with android system. At the same time, we are available to use the new features of the latest version of APIs. Therefore, we implemented our application based on Android SDK 5.0.

We implemented this application with a natural navigation bar at the bottom of the main page which includes five “tabs.” They are “For you,” “Favorites,” “Search,” “Saved” and “Setting.”

This was based on the user-friendly design for the users because they were able to reach each function directly and conveniently by using only one-touch action.

We implemented login page and create account page for users to enter into the system. We followed the standard flat design specification of Android 5.0, which let the visual effects look friendly. The images shown below demonstrate the main functions we implemented.

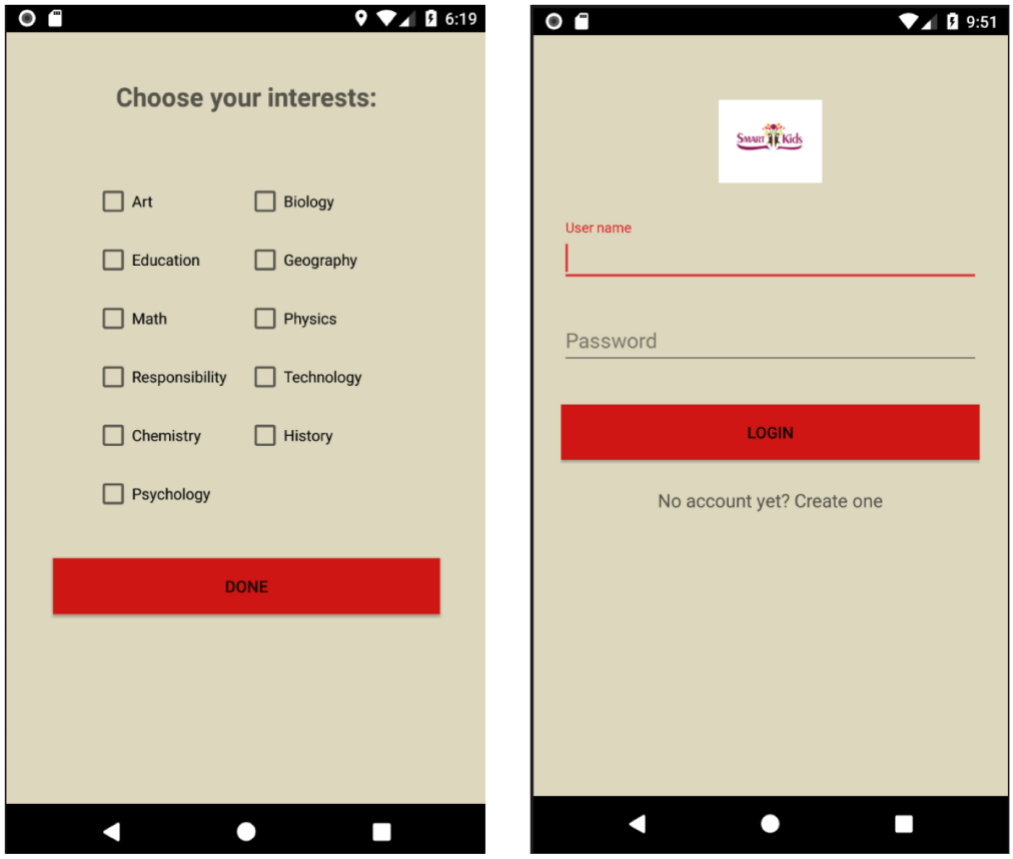


Figure 18 SmartKids - Interests and Login page

When the application was installed and opened by the users for the first time, it would lead them to a page called “Choose your interests.” In this page, there were eleven checkboxes which represented the eleven basic categories of the article we collected by our crawler system, which includes art, biology, education, geography, math, physics, responsibility technology, chemistry,

history, and psychology. These kinds of topics nearly covered all the subjects which involved both natural science and humanities. Users could select from one to eleven categories for their initial interests. These data would send to the server as the necessary user data for the recommendation system to calculate and provide the initial set of recommended articles for the specific users, which help them to select their categories of interests. When they finished this step, the application will bring them to the login page. In this page, users could log in by using an existing account or create a new account.

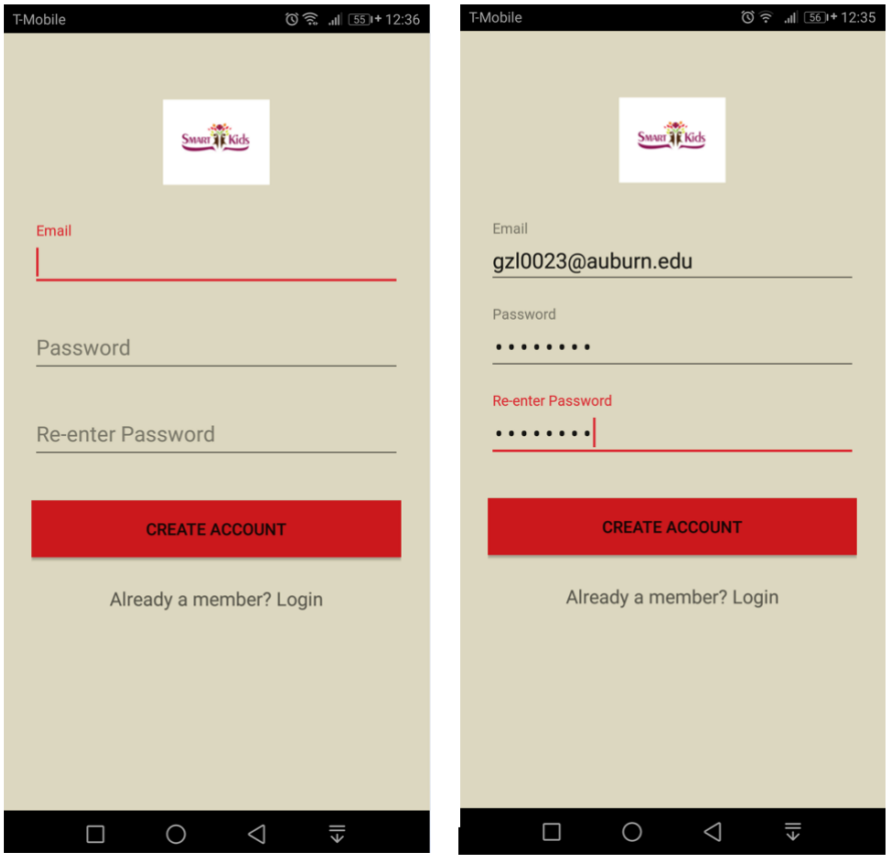


Figure 19 SmartKids - Register page

If the user did not have an account yet, they could create a new account on the page shown above. The name, email, and password were required items to fill in. If the user creates an account

successfully, the application will redirect to the main page. In the main page, there are five navigation tabs at the bottom, which were for you, favorites, search, bookmark, and setting. All the core functions and interactions present in these five tabs. The default tab is “For you.”

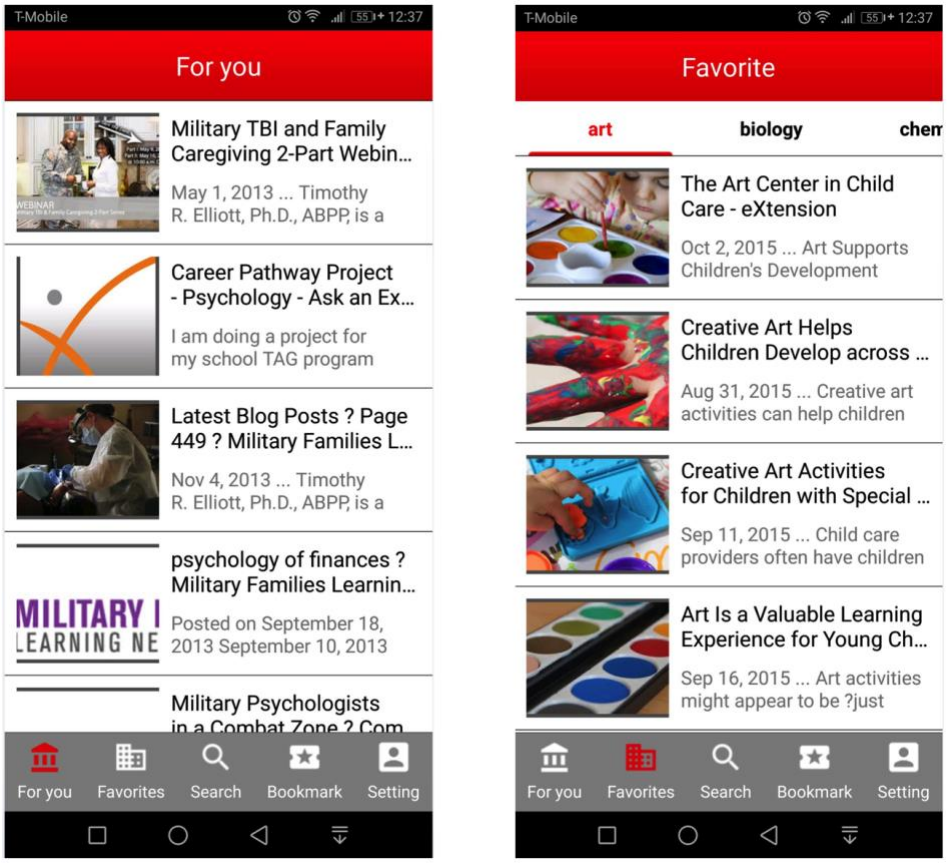


Figure 20 SmartKids - For you and Favorite page

The default page is “For you.” It is the first page when the user logs into the system. Based on the users’ basic information from the first page they selected, the “for you” page requested data from the server and displayed the recommended articles in the list. The data shown in this page is an example of data from the server. They were dynamic data based on the user’s information and the interactions with the application. We implemented a pagination function for all the pages with a shown list. On initial display, we sent a request to the server to get 20 articles in the list. When

user reached the bottom by scrolling the view, the application sends another request to the server to load the next 20 items.

In the “favorite” page, there were eleven categories in the middle. The list-view below showed the corresponding data for each group. Users could either click each category tag to review different article list or scroll right or left to switch the types. We also implemented the pagination function for each view-list.

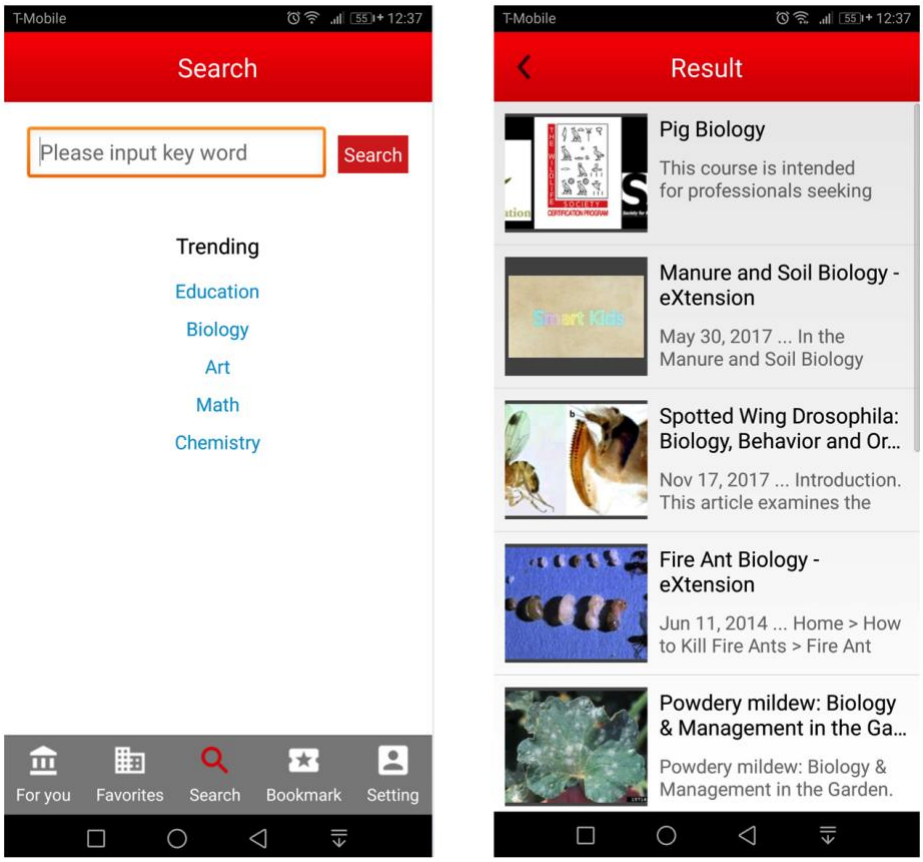


Figure 21 SmartKids – Search and Search result page

The third tab in the bottom was “Search” function. In this page, there was an edit text area for users to type in a keyword to search the articles they want to read. Below the search bar, there was a list showing the hot topics in this application. This function connected the users, and it could give the

users a suggestion and let them know what other users were searching. After the user searched a keyword, the request will be sent to the server and resend a list of data, which will list on the result page. The result page had a pagination function as well. If the user clicks each item, the application will show the details of the article on the details page. All the clicking article actions would open the details page. The image below shows the demonstration.

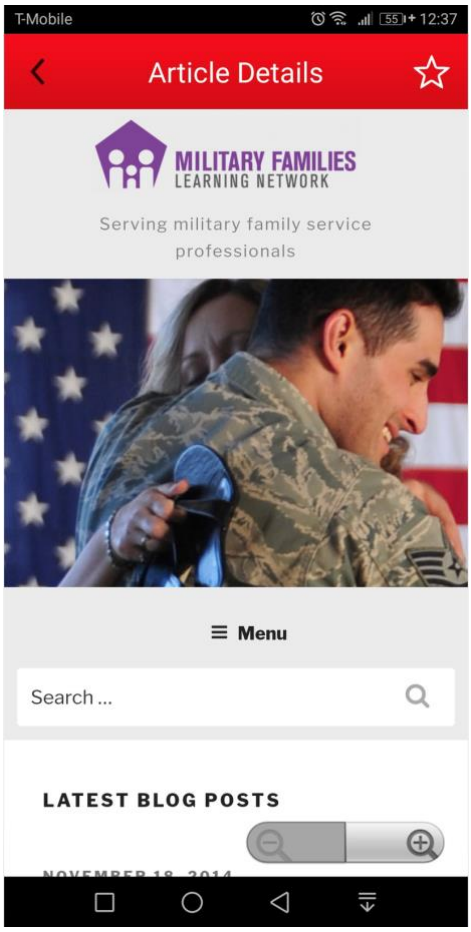


Figure 22 SmartKids - Article details page

In the middle of the "Article Details" page, there was the content of the article, which included the title, the time posted and the content. The content might be text-format, or a video combined with text and images. They were original HTML pages. We used the "WebView" component in the Android system to display the pages. The "WebView" can support all these formats. We

implemented a zoom in and zoom out function for this page. Users may click the buttons at the bottom to zoom in or zoom out to fix the text font size to let them easy to read. At the top of the page, there was a back button on the left and a star button on the right. If users clicked the back button, the application would return to the article list page. If users clicked the star-button, the star would become red color which represents the meaning that the user collected this article. Users could have the motivation of finding some interests in this article to collect it or would like to read it later. Users could click the star again to cancel collecting this article.

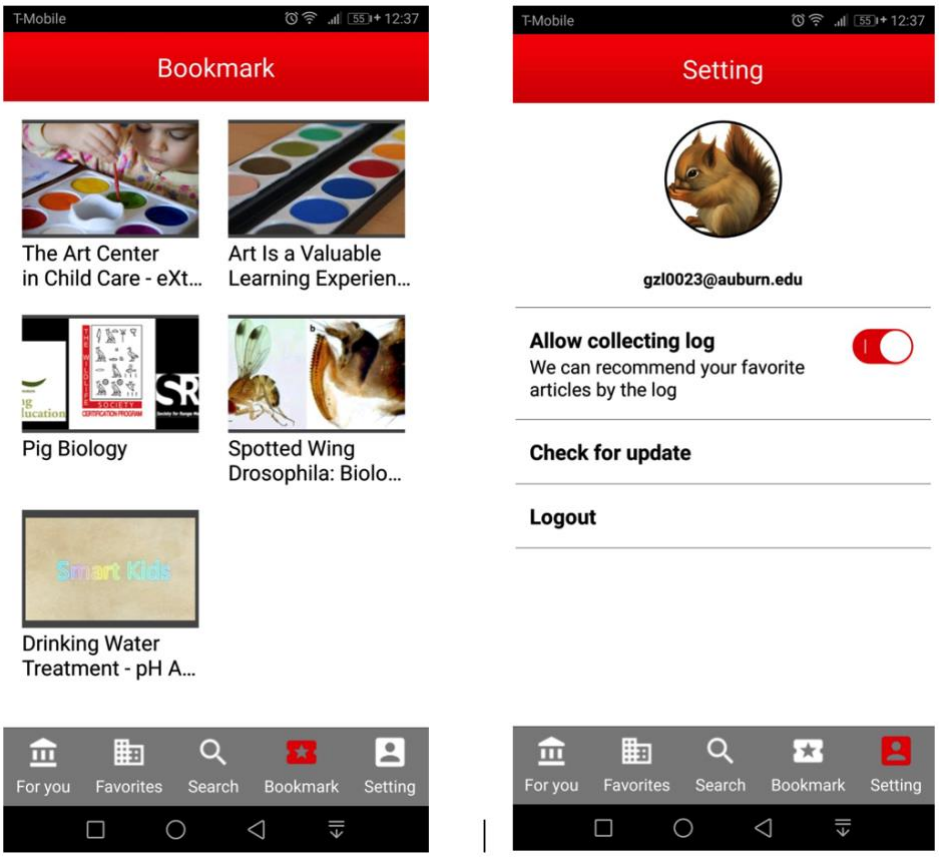


Figure 23 SmartKids – Bookmark and Setting page

The "Bookmark" page was a place to record all the articles the user collected. We used "GridView" component, which was different with the previous pages with "ListView" component to display

the article list, which let the articles have multiple displays. In this page, each row had two articles. Users could click them to review the articles they collected, and in each details page, they could cancel the collected articles so that they would not show in the "Saved" page again.

In the "Setting" page, we implemented three functions which included login or logout, updating the system and log collection controller. Users could log out to switch the account and click the update item to update the application. We considered the privacy right of the users and provided the switch option for the user. If they would like to let the system to analyze their data and calculate the recommend articles they might be interested in and provide the particular articles for them, they could switch on the button. Then the application would collect the data of their reading history and the collection history they marked or canceled in the log file at the specific time. The app would check the log file whenever the user launched the application. If the log file changed and the Internet was connected, it would send the log file to the server system to feed the recommendation system with the new users' data. If the user would not like to let the application record their data, they could switch off the button. Then the recommendation system only could provide the data based on the initial interests. However, they could change their mind whenever they would love to. The only thing they need to do was to switch on the button.

4.4 Testing

To increase system quality and performance, we analyzed the possible weakness of the software application. We tested our backend server system by using JUnit and tested our Android app by using Automated Monkey test.

4.4.1 Junit testing for backend system

The server API system mainly provided JSON-format data to feed the android application, so we designed and implemented the APIs mostly based on the requirements of the mobile app. The data form was clear and concise, and it was not the same as frontend system with different kinds of logic and interactions, so we use JUnit to test our backend server system to keep it stable and with secure and robust performance. We showed one of the test cases about the Article API as an example to discuss this process.

We inserted into the article table in the database with five rows of data.

articleId	articleDate	articleTitle	articleImg	articleDescription	articleUrl	category_id	source_id
1	2018-02-07 11:09:05	title one	http://192.168.0.24:8080/kids/article1.png	desc of first article	http://192.168.0.24:8080/kids/article1	1	1
2	2018-02-07 11:10:41	title two	http://192.168.0.24:8080/kids/article2.png	desc of the 2nd article	http://192.168.0.24:8080/kids/article2	1	1
3	2018-02-07 11:10:41	title three	http://192.168.0.24:8080/kids/article3.png	desc of the 3rd article	http://192.168.0.24:8080/kids/article3	2	1
4	2018-02-07 11:10:41	title four	http://192.168.0.24:8080/kids/article4.png	desc of the 4th article	http://192.168.0.24:8080/kids/article4	2	1
5	2018-02-07 11:10:43	title five	http://192.168.0.24:8080/kids/article5.png	desc of the 5th article	http://192.168.0.24:8080/kids/article5	3	1

5 rows in set (0.00 sec)

Figure 24 Article test data in the DB

We implemented the database logic functions in the project, and we wanted to test the functions in this API. We wrote JUnit test functions in the corresponding package in the project.

```
13 public class ArticleDaoTest {
14     IArticleDao dao = new ArticleDao();
15
16     @Test
17     public void testGetArticleByPageAndCategory(){
18         List<Article> list = dao.getArticleByPageAndCategory(0, 3, 2);
19         for(Article a: list)
20             System.out.println(a.getArticleId() + "," +a.getArticleTitle());
21     }
22     @Test
23     public void testGetTotalCount(){
24         int result = dao.getTotalCount();
25         System.out.println(result);
26     }
27     @Test
28     public void testGetTotalArticleByCategory(){
29         int result = dao.getTotalCountByCategory(1);
30         System.out.println(result);
31     }
32     @Test
33     public void testGetArticlesByKeywords(){
34         String key = "title";
35         List<Article> list = dao.getArticlesByKeyword(0,100,key);
36         for(Article a: list)
37             System.out.println(a.getArticleId() + "," +a.getArticleTitle());
38     }
39     @Test
40     public void testGetArticleByIds(){
41         Integer[] ids = {1,2,3,4};
42         List<Article> list = dao.getArticleByIds(ids);
43         for(Article a: list)
44             System.out.println(a.getArticleId() + "," +a.getArticleTitle());
45     }
46 }
```

Figure 25 JUnit test case

Take the first test case as an example.

As the definition of this API we designed previously, we knew that this function was to get the article list by using three parameters. The parameter “start” was the index of start position, the parameter “count” was the number of the articles we wanted, and the parameter “cId” was the id of the category.

```
public interface IArticleDao {
    /**
     * Get article list by page and category ID
     * @param start
     * @param count
     * @param cId
     * @return
     */
    public List<Article> getArticleByPageAndCategory (int start, int count, int cId);
```

Figure 26 Dao function

For this test case, testGetArticleByPageAndCategory(0, 3, 2), we wanted to get the articles from the category with the cId = 2, and we needed three articles. When we run this test case, the image below showed the result.

Figure 27 JUnit test result

The green bar in the left side of the IDE showed the test case run successfully, and as shown in the console, it printed the information we wanted because there were only two articles with the category id = 2.

4.4.2 Automated test (Monkey)

We analyzed the mobile applications in the Google store and app store and summarized a bunch of standard defects. They were connection timeout, crash, UI lag, force close, low-performance of user interaction. We write monkey instructions to test our final Android application. The Monkey is a piece of code which runs on the emulator or the mobile device which generates pseudo-random streams of user events, such as click, touch, and gesture. There are four option categories which include basic configuration options, operational constraints, event types and frequencies, and debugging options.

The basic syntax is written like the following instruction.

$ adb shell monkey [options] <event-count>

There are three stop conditions which are listed below.

1. If we have constrained the Monkey to run in one or more specific packages, it watches for attempts to navigate to any other packages and blocks them.

2. If the application crashes or receives any sort of unhandled exception, the Monkey program will stop and report the error.

3. If the application generates an ANR (application not responding) error, the Monkey program will stop and report the error.

For example:

$ adb shell monkey -p edu.auburn.smart -v 500

In our application, the package is included in the AndroidManifest.xml file, which is "edu.auburn.smart".

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.auburn.smart">
```

We use the instruction above to generates 500 random events to operate our application, and it runs well. The log below shows a part of the result when the instruction runs on the application.

```
:Sending Touch (ACTION_UP): 0:(600.8755,1625.323)
:Sending Touch (ACTION_DOWN): 0:(554.0,850.0)
:Sending Touch (ACTION_UP): 0:(554.18115,861.6126)
```

```
:Sending Touch (ACTION_DOWN): 0:(540.0,2236.0)
:Sending Touch (ACTION_UP): 0:(544.965,2252.6113)
:Sending Trackball (ACTION_MOVE): 0:(4.0,-1.0)
:Sending Trackball (ACTION_UP): 0:(0.0,0.0)
:Sending Touch (ACTION_DOWN): 0:(872.0,2152.0)
:Sending Touch (ACTION_UP): 0:(870.5852,2166.883)
:Sending Touch (ACTION_DOWN): 0:(1091.0,671.0)
:Sending Touch (ACTION_UP): 0:(1091.9294,653.93677)
:Sending Trackball (ACTION_MOVE): 0:(-5.0,0.0)
:Sending Touch (ACTION_DOWN): 0:(1285.0,616.0)
// Allowing start of Intent { cmp=edu.auburn.smart/.ui.WebActivity } in package
edu.auburn.smart
Events injected: 500
:Sending rotation degree=0, persist=false
:Dropped: keys=0 pointers=4 trackballs=0 flips=0 rotations=0
## Network stats: elapsed time=2788ms (0ms mobile, 0ms wifi, 2788ms not connected)
// Monkey finished
```

The instruction generated 500 events and injected into the application, such as ACTION_UP, ACTION_DOWN, and ACTION_MOVE.

Chapter 5. Results and Analysis

We conducted three task lists, and 22 users participated the functional and usability test. They finished the pre-survey and post-survey questionnaire after the test. The three task lists asked the user to test the three Android versions in a random order to ensure that there was no carry over effect of learning and reduce potential experimental bias. The result of the pre-survey and post-survey were used to evaluate our software application.

Our final SmartKids application was tested with undergraduate and graduate student from Auburn University Computer Science, and the many have a background in User Interface Design. They were asked to take a pre-survey questionnaire, which asked about general information of users, such as their age, gender and what operating system their mobile phone is. It asked users' preference on specific features, such as voice search vs. text search. It also contained open-ended questions such as "What are your two favorite apps." Moreover, we also asked about their experience in online learning and the acceptance of receiving the recommended information.

After that, we created three task lists for users to test the functionality and usability of the different android version of the application. The task list is a guide to help users through usability testing, and it covers all the functions of the application. They told the users details steps about the app which include navigating to the main page, use the search function and read the article pages and so on. We tested our application on the Huawei Mate SE mobile device.

We asked the participant to finish the post-survey after completing the whole list of tasks. The primary purpose of the post-survey was to investigate and evaluate the performance and the usability of the application. We asked the questions such as if the application is easy to use, they

love static or dynamic version more and if they think the recommend articles will help them learn.

We also asked them to list the most positive and negative aspects of the application.

5.1 Scenario task

Scenario #1

1. Open the first "IW2K" app (version 1).

2. Click "Get Science" button on the main page and read the details.

3. Click the back button in the top-left corner to open the menu view.

4. Click the "Get Science" item to navigate to the "Get Science" page.

5. Click the "search" button in the top-right corner.

6. Type in "water" as a keyword to find the result.

7. Click "Math" button to open the article page.

8. Click back button try to return the main page.

9. Click the physic back button on the phone to return the main page.

10. Close the application.

Scenario #2

1.Open the second "IW2K" app (version 2).

2.Click "Math" button on the main page and read the article.

3.Click the search button and type in the keyword "math" to search the result.

4.Click back button to return the home page.

4.Click the setting button in the top-right corner to switch the real style and cartoon style.

5.Click the menu button in the top-left corner.

6.Click the "Health" item to navigate to the Health page.

7.Click "Health Eating" button to open the article page.

10. Close the application.

Scenario #3

1. Open the "Smart" app (version 3).

2. Select the categories you are interested in and click "done" button.

3. Click "No account yet? Create one" link to open the create page.

4. Create a new account and navigate to the main page.

5. Click the bottom button "For you", "Favorites", "Search", "Bookmark", "Me" to review the whole view.

6. Click the "For you" button to navigate to this page and scroll down to loading more data.

7. Click "Favorites" at the bottom and navigate to this page.

8. Click "education" to select this category.

9. Click the second item to open the details page.

10. Click the star button in the top-right corner to mark this article as favorite.

11. Click the "+" and "-" to change the size of the text.

12. Click back button to return to the main page.

13. Click the "Bookmark" at the bottom to navigate to this page.

14. Click the article you marked and read it again.

15. Click the star button to cancel marked and click back button to return.

16. Click "Search button " at the bottom to navigate to this page.

17. Type in "water" as the keyword and search the result.

18. Click one item to read the details.

19. Click the back button to return to the result page.

20. Click the back button to return to the main page.

21. Click the first “Trending word" education as the keyword to search directly.

22. Click back button to return to the main page.

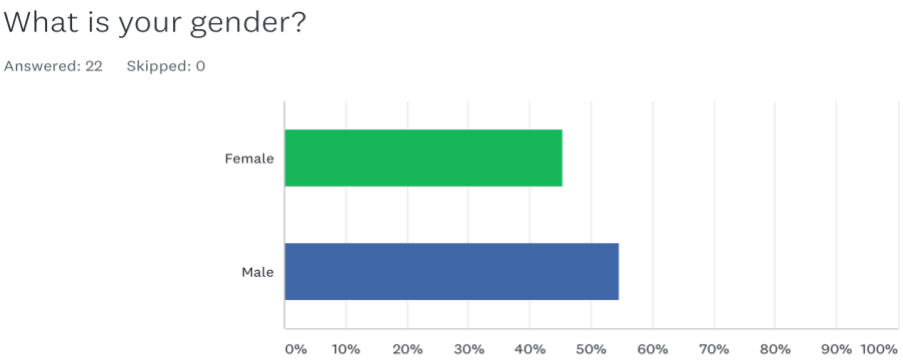
23. Click "Setting" button at the bottom to navigate to this page.

24. Switch the "Allow collecting log" button to turn on and turn off.

25. Close the application.

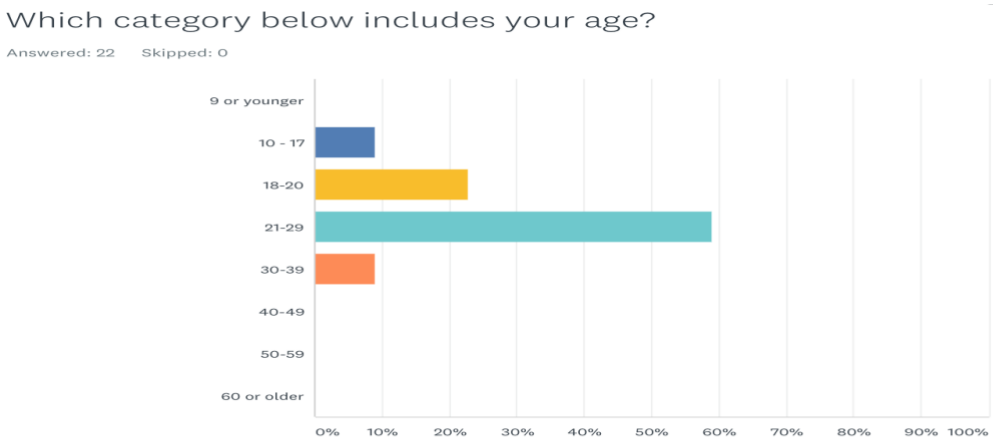
5.2 Pre-Survey

The pre-survey includes 18 questions, and 22 participants tested this application. They were all undergraduate and graduate students from Auburn University. Ten of the participants were females and 12 were male.



ANSWER CHOICES	RESPONSES	
Female	45.45%	10
Male	54.55%	12
TOTAL		22

Table 1 Pre-survey Q1



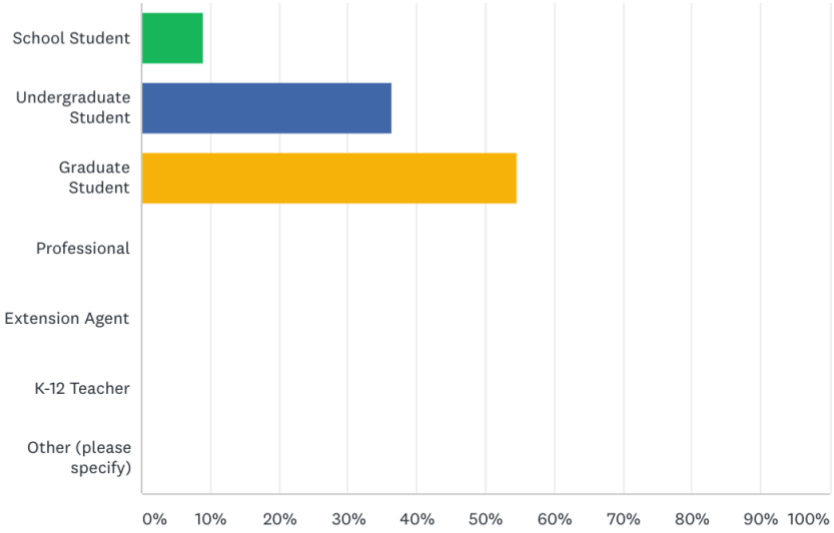
ANSWER CHOICES	RESPONSES	
9 or younger	0.00%	0
10 - 17	9.09%	2
18-20	22.73%	5
21-29	59.09%	13
30-39	9.09%	2
40-49	0.00%	0
50-59	0.00%	0
60 or older	0.00%	0
TOTAL		22

Table 2 Pre-survey Q2

The age range lists are 2 of 10 - 17, 5 of 18 - 20, 13 of 21 - 29 and 2 of 30 – 39. 2 of them are school student, 8 are undergraduate students, and 12 are the graduate student.

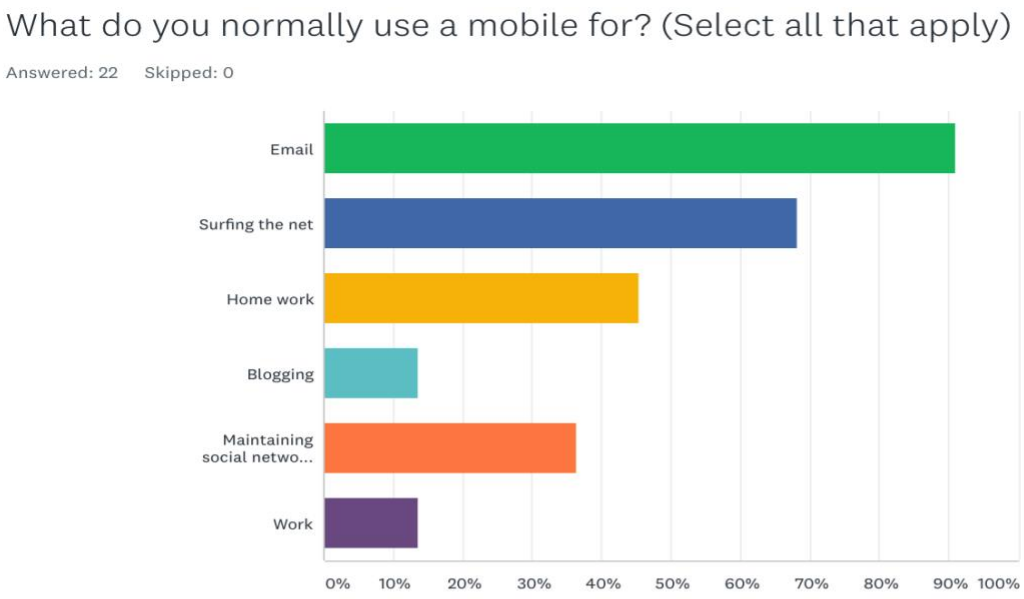
What is your classification?

Answered: 22 Skipped: 0



ANSWER CHOICES		RESPONSES	
School Student		9.09%	2
Undergraduate Student		36.36%	8
Graduate Student		54.55%	12
Professional		0.00%	0
Extension Agent		0.00%	0
K-12 Teacher		0.00%	0
Other (please specify)	Responses	0.00%	0
TOTAL			22

Table 3 Pre-survey Q3



ANSWER CHOICES	RESPONSES	
Email	90.91%	20
Surfing the net	68.18%	15
Home work	45.45%	10
Blogging	13.64%	3
Maintaining social network like Facebook, twitter etc	36.36%	8
Work	13.64%	3
Total Respondents: 22		

Comments (1)

RESPONSES (1) TEXT ANALYSIS TAGS (0)

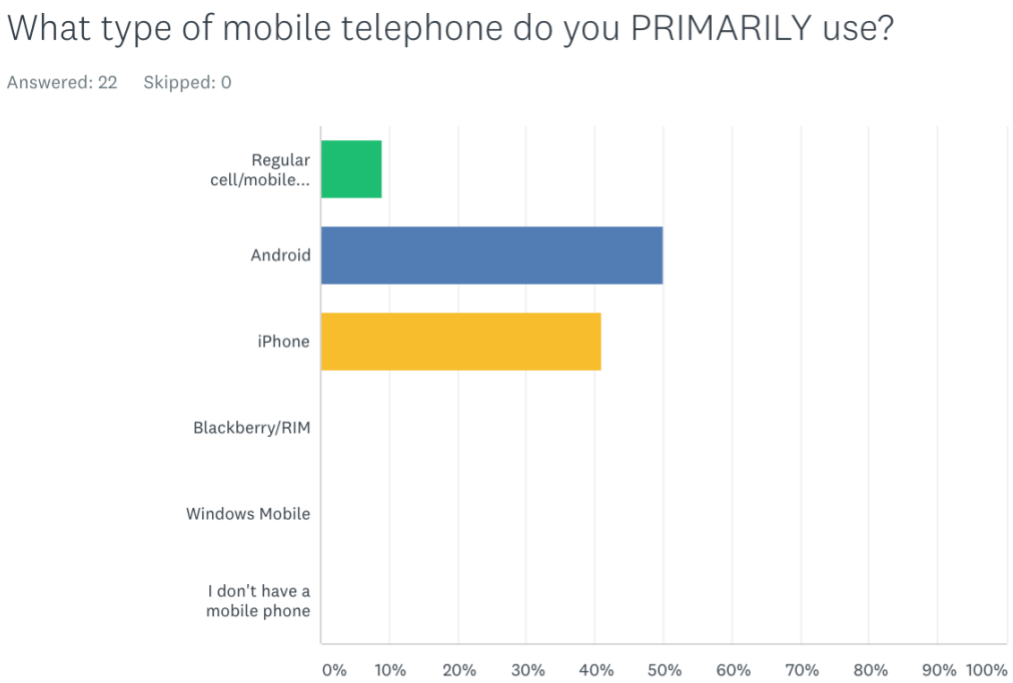
Add Tags Filter by Tag Search responses

Showing 1 response

youtube

Table 4 Pre-survey Q4

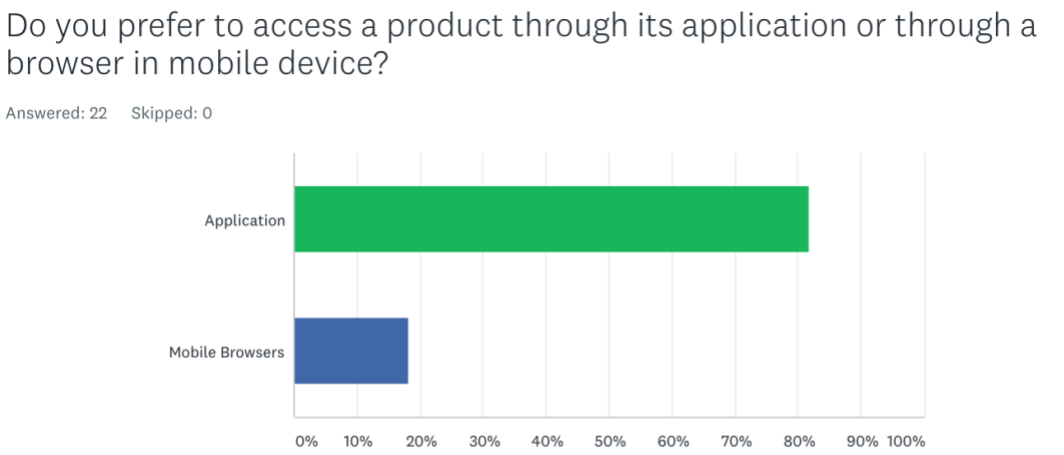
When they answered the question "What do you normally use mobile for?" That is a multiple choice, and 90.91% of them select email, 68.18% of them select the Internet, 45.45% of them choose the homework, 13.64% select blog, 36.36% select social network and 13.64% select work. There was an extra response, which is YouTube.



ANSWER CHOICES	RESPONSES	
Regular cell/mobile phone (not a smartphone)	9.09%	2
Android	50.00%	11
iPhone	40.91%	9
Blackberry/RIM	0.00%	0
Windows Mobile	0.00%	0
I don't have a mobile phone	0.00%	0
TOTAL		22

Table 5 Pre-survey Q5

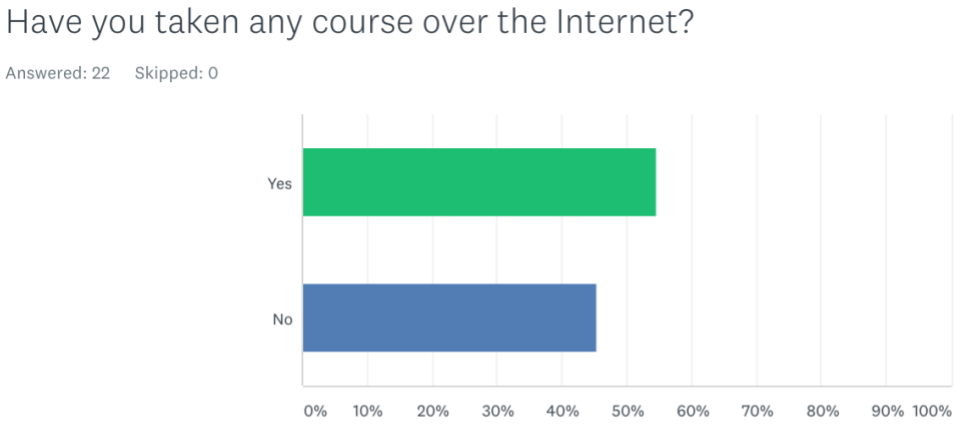
When they answered the question "What type of mobile phone do you primarily use?" 9.09% of them choose the regular cell/mobile phone (not a smartphone), 50% of them prefer the Android, 40.91% of them like the iPhone. Nobody uses blackberry or windows mobile phone.



ANSWER CHOICES	RESPONSES	
Application	81.82%	18
Mobile Browsers	18.18%	4
TOTAL		22

Table 6 Pre-survey Q6

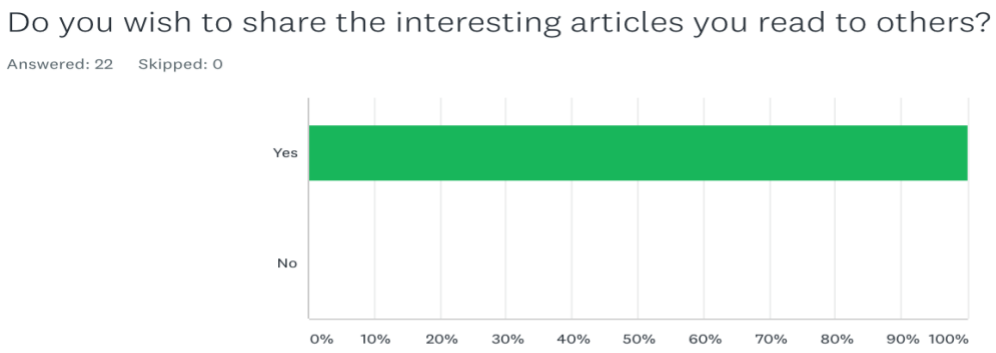
81.82% of them prefer to access a product through its application, and 18.18% of them prefer the browser in a mobile device.



ANSWER CHOICES	RESPONSES	
Yes	54.55%	12
No	45.45%	10
TOTAL		22

Table 7 Pre-survey Q7

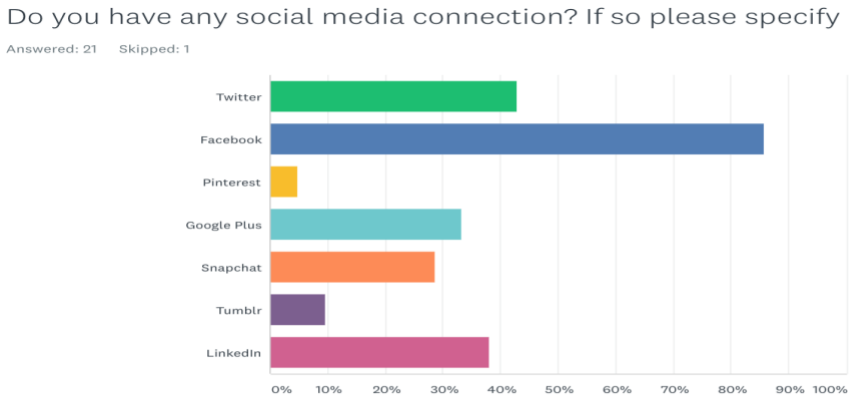
54.55% of them have the experience of taking the course over the Internet. 45.45% of them do not have this kind of experience.



ANSWER CHOICES	RESPONSES	
Yes	100.00%	22
No	0.00%	0
TOTAL		22

Table 8 Pre-survey Q8

All of them would like to share the fascinating articles to others.

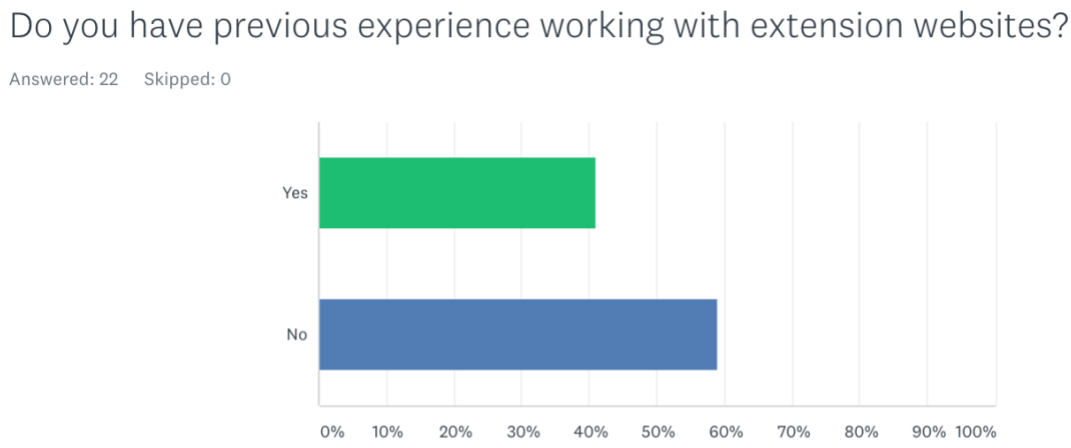


ANSWER CHOICES	RESPONSES	
Twitter	42.86%	9
Facebook	85.71%	18
Pinterest	4.76%	1
Google Plus	33.33%	7
Snapchat	28.57%	6
Tumblr	9.52%	2
LinkedIn	38.10%	8
Total Respondents: 21		

Comments (1)

Table 9 Pre-survey Q9

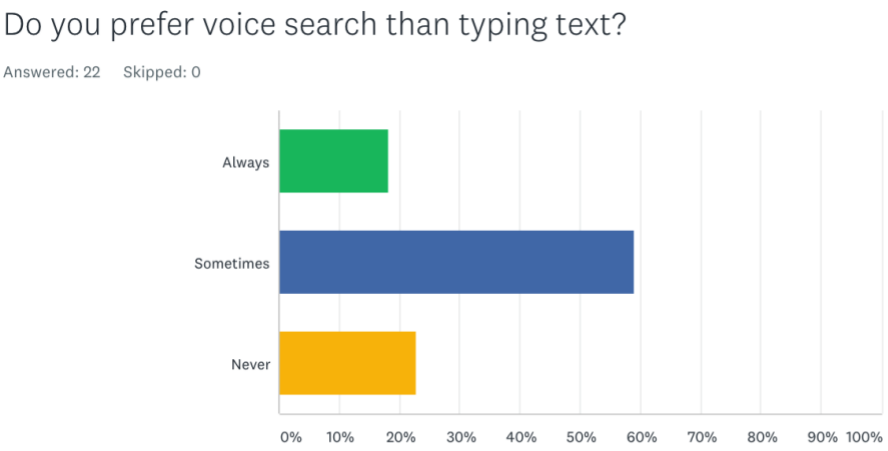
When they answered the question “Select the social media connection you have.” That is a multiple choice. 42.86% selected Twitter, 85.71% select Facebook, 4.76% selected Pinterest, 33.33% selected Google Plus, 28.57% selected Snapchat, 9.52% selected Tumblr and 38.10% selected LinkedIn.



ANSWER CHOICES	RESPONSES	
Yes	40.91%	9
No	59.09%	13
TOTAL		22

Table 10 Pre-survey Q10

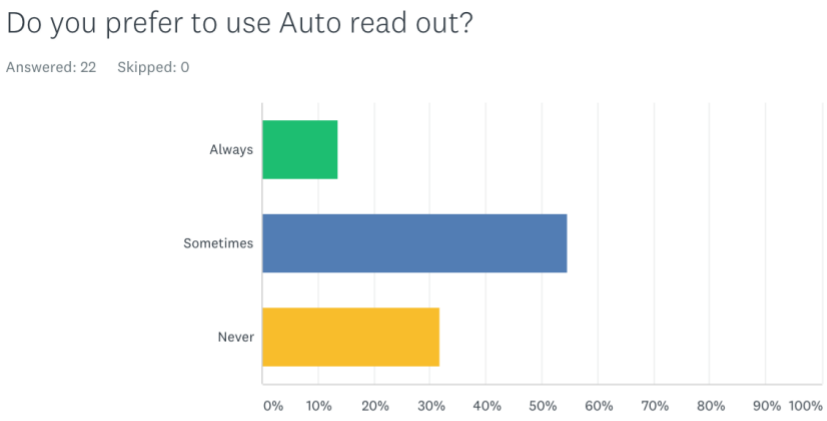
40.91% of them have previous experience working with extension websites. 59.09% of them do not have.



ANSWER CHOICES	RESPONSES	
Always	18.18%	4
Sometimes	59.09%	13
Never	22.73%	5
TOTAL		22

Table 11 Pre-survey Q11

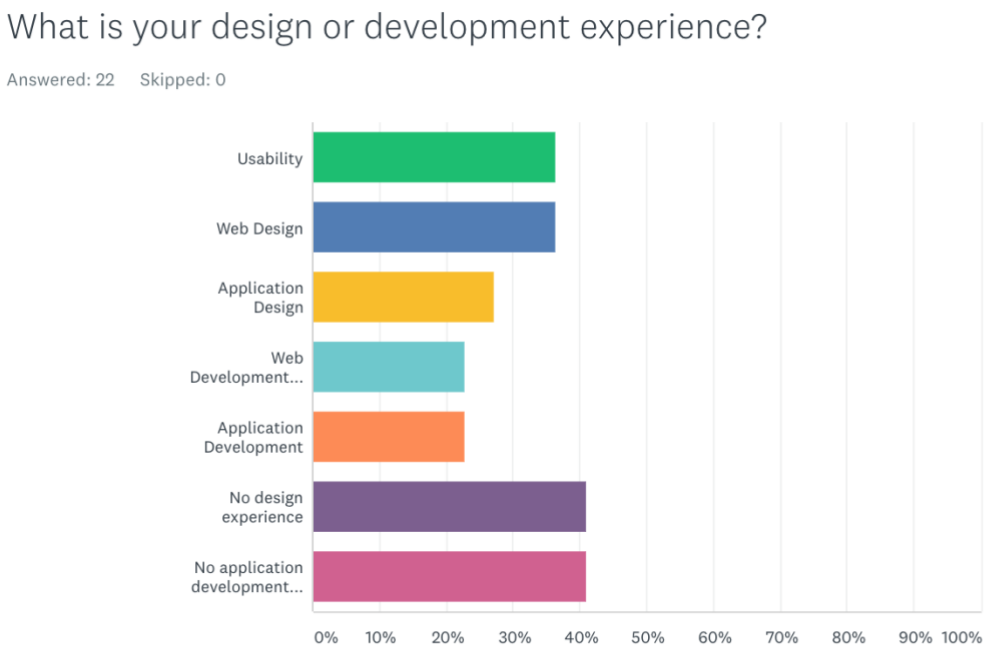
When they answered the question "Do you prefer voice search than typing text?" 18.18% selected "always," 59.09% selected "sometimes" and 22.73% selected "never."



ANSWER CHOICES	RESPONSES	
Always	13.64%	3
Sometimes	54.55%	12
Never	31.82%	7
TOTAL		22

Table 12 Pre-survey Q12

When they answered the question “Do you prefer to use Auto read out?” 13.64% selected “always,” 54.55% selected “sometimes” and 31.82% selected “never.”



ANSWER CHOICES	RESPONSES	
Usability	36.36%	8
Web Design	36.36%	8
Application Design	27.27%	6
Web Development Development	22.73%	5
Application Development	22.73%	5
No design experience	40.91%	9
No application development experience	40.91%	9
Total Respondents: 22		

Table 13 Pre-survey Q13

When they answered the question “What is your design or development experience?” That is a multiple choice. 36.36% selected usability, 36.36% selected web design, 27.27% selected application design, 22.73% selected web development, 22.73% selected application development, 40.91% selected no design experience and 40.91% selected no application development experience.

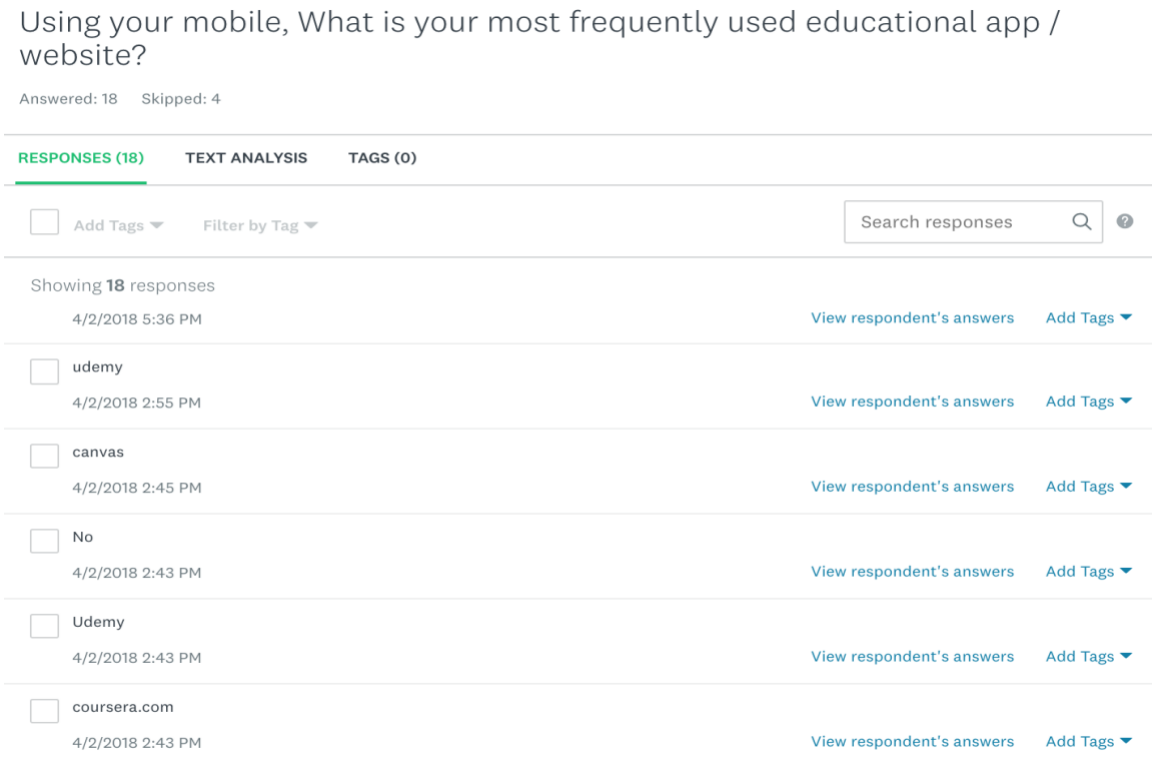


Table 14 Pre-survey Q14

When they answered the question "Using your mobile phone, what is your most frequently used educational application or website?" The answers we collected are Udemy, Canvas, Coursera, Lynda and Google Translate.

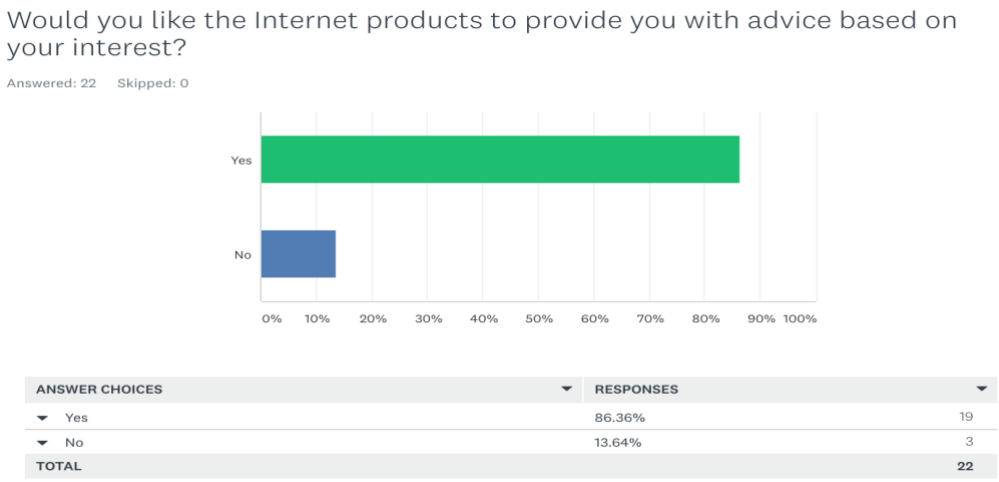
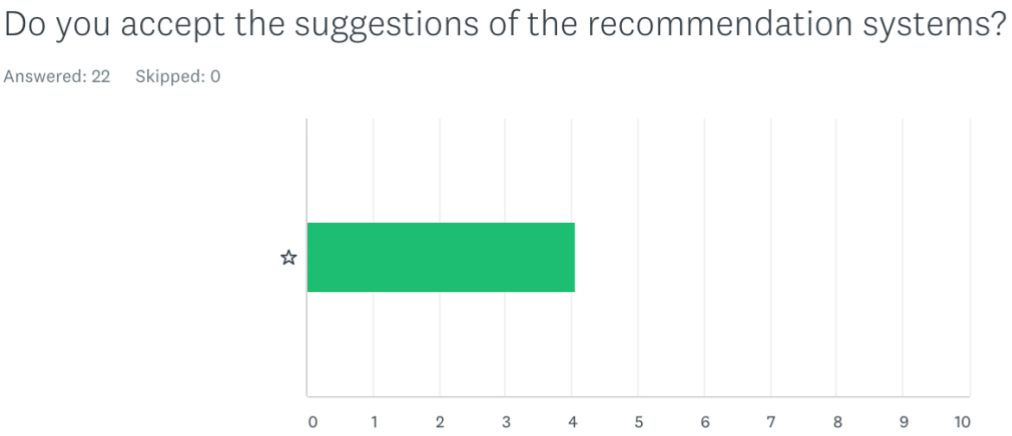


Table 15 Pre-survey Q15

86.36% of them would like the Internet products to provide with advice based on their interest.



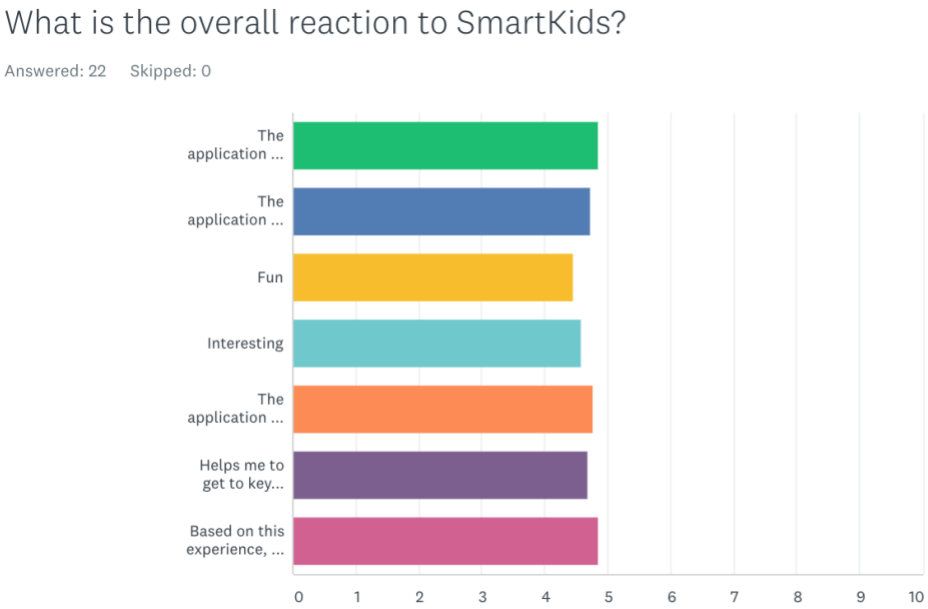
	I DO NOT ACCEPT.	I DO NOT CARE.	ACCEPT.	ACCEPT A LITTLE.	ACCEPT STRONGLY.	TOTAL	WEIGHTED AVERAGE
☆	4.55% 1	13.64% 3	4.55% 1	27.27% 6	50.00% 11	22	4.05

Table 16 Pre-survey Q16

When they answered the question "Do you accept the suggestions of the recommendation system?" 4.55 selected "I do not accept," 13.64% selected "I do not care," 4.55% selected "accept," 27.27% selected "accept a little," 50% selected "accept strongly."

5.3 Post-Survey

By collecting the results of the post-survey, we got the participant's perception of the application from the usability perspective. The average response to all the usability questions indicated a positive reaction to the app from the usability point of view. Data were collected on a scale of the 1-5 scale, 'strongly disagree' being the lowest and 'strongly agree' being the highest.



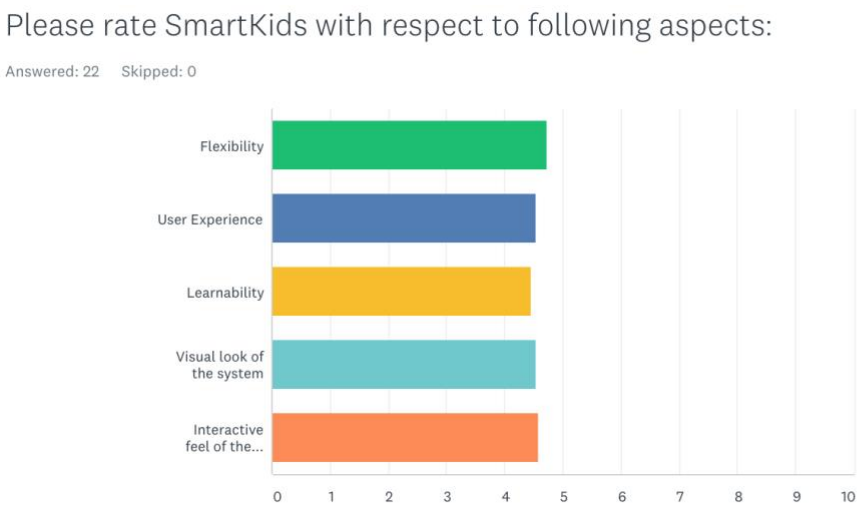
	STRONGLY AGREE	AGREE	NEUTRAL	DISAGREE	STRONGLY DISAGREE	TOTAL	WEIGHTED AVERAGE
The application is attractive	90.91% 20	4.55% 1	4.55% 1	0.00% 0	0.00% 0	22	4.86
The application is easy to use	77.27% 17	18.18% 4	4.55% 1	0.00% 0	0.00% 0	22	4.73
Fun	45.45% 10	54.55% 12	0.00% 0	0.00% 0	0.00% 0	22	4.45
Interesting	59.09% 13	40.91% 9	0.00% 0	0.00% 0	0.00% 0	22	4.59
The application is simple to navigate	77.27% 17	22.73% 5	0.00% 0	0.00% 0	0.00% 0	22	4.77
Helps me to get to key information quickly	72.73% 16	22.73% 5	4.55% 1	0.00% 0	0.00% 0	22	4.68
Based on this experience, I will use this application again	86.36% 19	13.64% 3	0.00% 0	0.00% 0	0.00% 0	22	4.86

Table 17 Post survey Q1

For the question “What is the overall reaction to SmartKids application?” We give seven aspects for them to answer. For the aspect “the application is attractive,” 90.91% selected strongly agree, 4.55% selected agree and 4.55% selected neutral. For the aspect “the application is easy to use,” 77.27% selected strongly agree, 18.18% selected agree and 4.55% selected neutral. For the aspect “the application is fun,” 45.45% selected strongly agree and 54.55% selected agree. For the aspect

"the application is interesting," 59.09% selected strongly agree and 40.91% selected agree. For the aspect "the application is simple to navigate," 77.27% selected strongly agree, 22.73% selected agree. For the aspect "the application helps me to get the key information quickly," 77.73% selected strongly agree, 22.73% selected agree and 4.55% selected neutral. For the aspect "I will use this application again," 86.36% selected strongly agree, 13.64% selected agree.

Hypotheses "Aesthetics," "Effectiveness," "ease of use" and "efficiency" are supported by the above survey results as most participants feel that the application is fun, exciting and easy to use and they could get the critical information quickly.

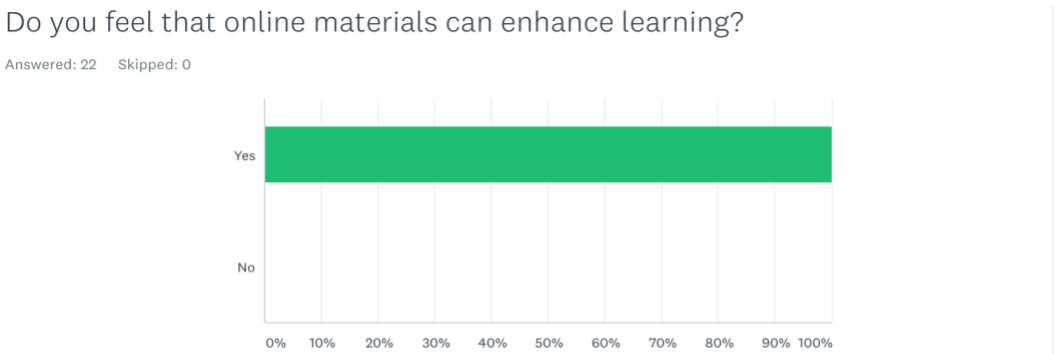


	VERY HIGH	HIGH	MODERATE	LOW	VERY LOW	TOTAL	WEIGHTED AVERAGE
Flexibility	72.73% 16	27.27% 6	0.00% 0	0.00% 0	0.00% 0	22	4.73
User Experience	63.64% 14	27.27% 6	9.09% 2	0.00% 0	0.00% 0	22	4.55
Learnability	45.45% 10	54.55% 12	0.00% 0	0.00% 0	0.00% 0	22	4.45
Visual look of the system	63.64% 14	27.27% 6	9.09% 2	0.00% 0	0.00% 0	22	4.55
Interactive feel of the system	63.64% 14	31.82% 7	4.55% 1	0.00% 0	0.00% 0	22	4.59

Table 18 Post survey Q2

For the question “Please rate SmartKids with respect to following aspects.” We give five aspects for them to rate. For the “Flexibility,” 72.73% selected very high and 27.27% selected high. For the “User Experience,” 63.64% selected very high, 27.27% selected high and 9.09% selected moderate. For the “Learnability,” 45.45% selected very high and 54.55% selected high. For the “Visual look of the system,” 63.64% selected very high, 27.27% selected high and 9.09% selected moderate. For the “Interactive feel of the system,” 63.64% selected very high, 31.82% selected high and 4.55% selected moderate.

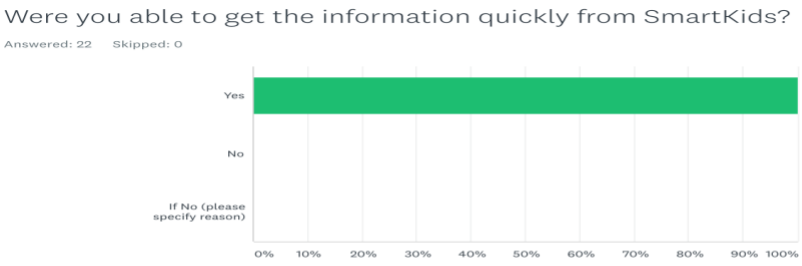
Hypotheses “Aesthetics,” “Effectiveness,” “satisfaction” and “efficiency” are supported by the above survey results because most participants give a high value of the flexibility, user experience, visual look and interact feeling of the application. We make the application with dynamic data. When users read articles, the app will send a request to the server to retrieve data. That will take response time which may lead to a delay of UI responding. To support efficiency, we use the third-party library OKHttp to load data from server and GSon to purser data to render the view in the app. These guaranteed the users experience of the application.



ANSWER CHOICES	RESPONSES	
Yes	100.00%	22
No	0.00%	0
TOTAL		22

Table 19 Post survey Q3

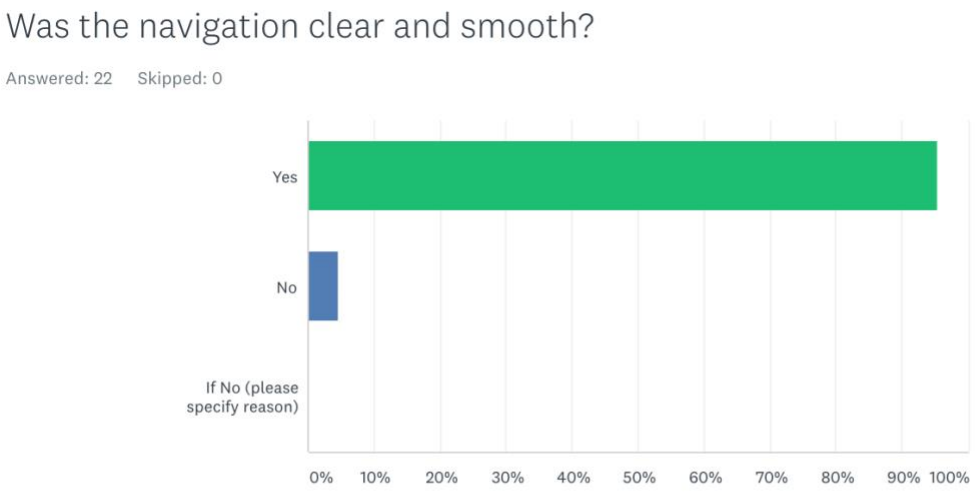
100% of them feel that online materials can enhance learning, which supports our hypothesis of effectiveness.



ANSWER CHOICES		RESPONSES	
Yes		100.00%	22
No		0.00%	0
If No (please specify reason)	Responses	0.00%	0
TOTAL			22

Table 20 Post survey Q4

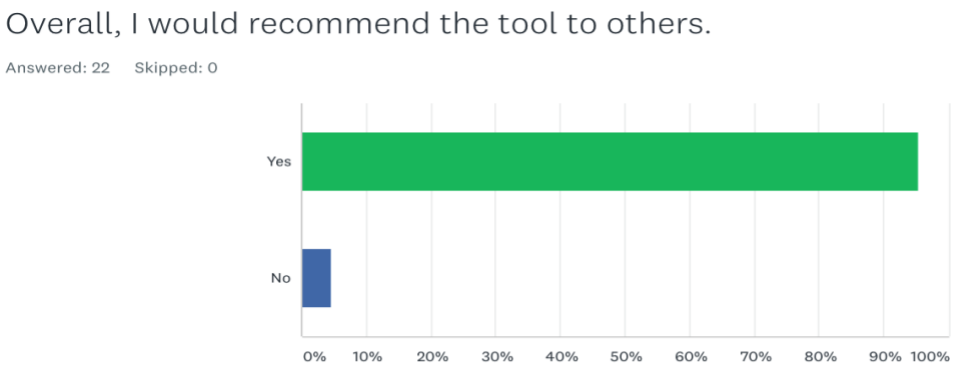
100% of them can get the information quickly by using the application, which supports the hypothesis of feature that the new version could help users to find their potential interests.



ANSWER CHOICES		RESPONSES	
Yes		95.45%	21
No		4.55%	1
If No (please specify reason)	Responses	0.00%	0
TOTAL			22

Table 21 Post survey Q5

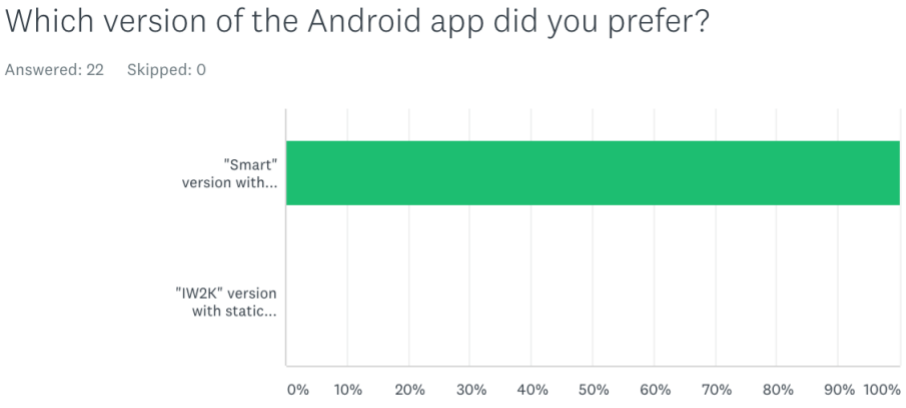
95.45% of them believe the navigation is clear and smooth and 4.55% do not. This result supports the hypothesis of aesthetics that the new version app will have a new clean layout and framework which give the users a feeling that the system is aesthetically pleasing.



ANSWER CHOICES	RESPONSES	
Yes	95.45%	21
No	4.55%	1
TOTAL		22

Table 22 Post survey Q6

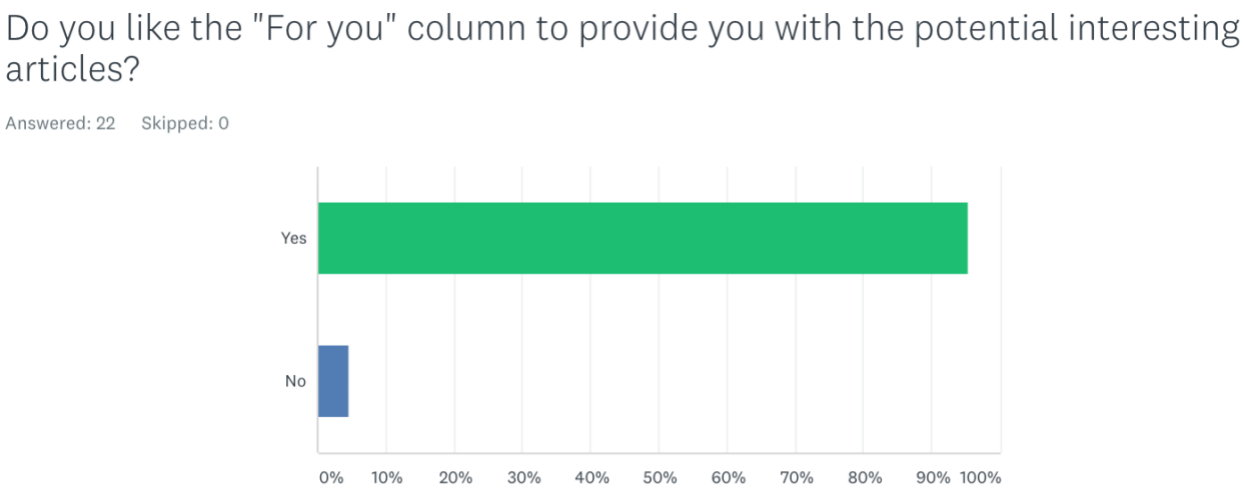
95.45% of them would like to recommend the tool to others and 4.55% would not.



ANSWER CHOICES	RESPONSES	
"Smart" version with dynamic data.	100.00%	22
"IW2K" version with static data.	0.00%	0
TOTAL		22

Table 23 Post survey Q7

Compared to the IW2K version with static data, 100% of them prefer the SmartKids version with dynamic data. This result proves our hypothesis of satisfaction that most users will have a positive attitude to the recommend function.

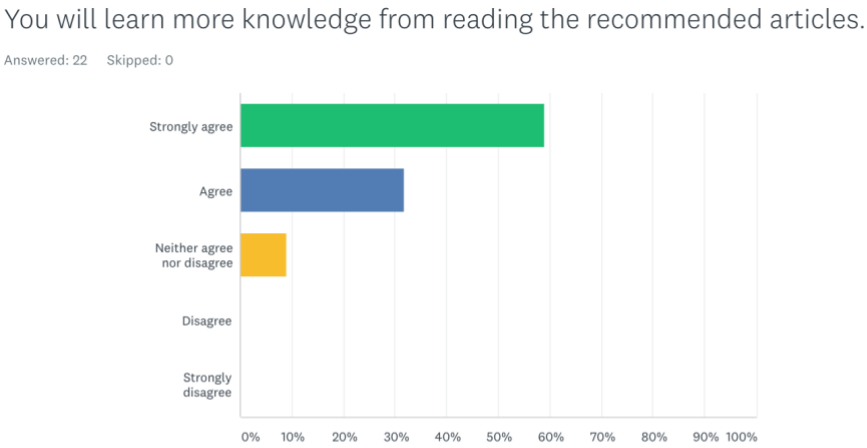


ANSWER CHOICES	RESPONSES	
Yes	95.45%	21
No	4.55%	1
TOTAL		22

Table 24 Post survey Q8

When they answered the question “Do you like the ‘For you’ column to provide you with the potential interesting articles?” 95.45% of them selected yes, and 4.55% selected no. This result proves the hypothesis of effectiveness and satisfaction. Most of the end users prefer to receive the

recommended articles than static data. They confirmed that the new version could help them to get more interesting articles.



ANSWER CHOICES	RESPONSES	
Strongly agree	59.09%	13
Agree	31.82%	7
Neither agree nor disagree	9.09%	2
Disagree	0.00%	0
Strongly disagree	0.00%	0
TOTAL		22

Table 25 Post survey Q9

When they answered the question “You will learn more knowledge from reading the recommended articles.” 59.09% of them selected strongly agree, 31.82% selected agree and 9.09% selected neither agree nor disagree. This result also proves the hypothesis of effectiveness and satisfaction. Most of the end users agree or strongly agree the recommend articles help them learn more useful knowledge and their attitude to the recommendation system is positive.

We asked the participants to list the positive aspects of the application SmartKids they observed. Seven of them like the "Bookmark" column. Four of them like the "For you" column. Five of them

make comments of this application which include "it is easy to use," "the logic is clean," and "it is attractive." Seven of them would like to receive more recommend articles.

We asked the participants to list the most negative aspects of the area of improvement. Six of them gave the answers. Four participants reported that the system is slow when the articles loading and two of them reported the user interface could be more attractive.

Chapter 6. Future Work and Conclusions

6.1 Conclusion

In this research project, we designed and implemented web crawler system to collect the articles from extension.org. We stored these articles into the MySQL database on our server. We developed the backend system to provide the Restful API for the front-end system. We used the Android version Smartkids application as the front-end system to display the articles. We collected users' reading history and primary interests and send the data to the backend system.

We asked 22 participants to take the pre-survey and then asked them to take three task scenarios in a random order to process the usability and functional test. After that, they finished the post-survey based on the user experience of the three different versions of the application.

Based on the result of the survey, more than 95% users agree or strongly agree the application is fun and interesting. They believe it is easy to use and it helps them get the key information quickly. More than 90% users give a high or very high value to the flexibility and user experience of the application. All of the users confirmed they could obtain the information quickly by using SmartKids. Compared to the previous versions with static data, all of them prefer the new version with dynamic data. These support our hypotheses "aesthetics," "effectiveness" and "ease of use." The innovation of this version is that we make the application with dynamic data. When users read articles, the application will send a request to the server to retrieve data, which will take response time which may lead to the lag of UI responding. To support efficiency, we use the third-party library OKHttp to load data from server and GSon to parse the data to render the view in the app. These third-party libraries guaranteed the speed and quality of both user experience and developing process. According to the result of the post-survey, more than 95% users like the "for you" column

to provide the potential interesting articles and showed the interests about the recommend articles, which support the hypotheses of "features" and "satisfaction."

6.2 Future Work

Firstly, we need to extend our crawler system to adapt it to other websites, so that we cans obtain more useful articles. Secondly, as the end users give the positive attitude to the recommendation system, we need to implement the recommendation system based on the users' data and article resource and integrate it into the whole project. Thirdly, we need to deploy the Android application to the Google Store. Finally, there are nearly half of users who are using iPhone devices, so we need to design and implement the IOS version and deploy it into the App Store.

Bibliography

[1] Alhassan, R. (2016). Mobile Learning as a Method of Ubiquitous Learning: Students' Attitudes, Readiness, and Possible Barriers to Implementation in Higher Education. Journal of Education and Learning, 5(1), 176. doi:10.5539/jel.v5n1p176

[2] Fahri, H., & Samsudin, K. (2012). Mobile Learning Environment System (MLES): The Case of Android-based Learning Application on Undergraduates' Learning. International Journal of Advanced Computer Science and Applications, 3(3). doi:10.14569/ijacsa.2012.030311

[3] Masood, M., & Thigambaram, M. (2015). The Usability of Mobile Applications for Pre-schoolers. Procedia - Social and Behavioral Sciences, 197, 1818-1826. doi:10.1016/j.sbspro.2015.07.241

[4] Reithinger, N., Russ, A., & Schumacher, K. (2015). User-centered Interaction Design of a Mobile Learning Platform for the Generation 60. Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct - MobileHCI 15. doi:10.1145/2786567.2794305

[5] Ali, A., Alrasheedi, M., Ouda, A., & Capretz, L. F. (2014). A Study of The Interface Usability Issues of Mobile Learning Applications for Smart Phones from the User's Perspective. International Journal on Integrating Technology in Education, 3(4), 1-16. doi:10.5121/ijite.2014.3401

[6] Gan, Jason. "Developing Smart Client Mobile Applications." Mobile Computing, pp. 383–387., doi:10.4018/978-1-60566-054-7.ch033.

[7] Nesset, V. “Children in the Information Technology Design Process: A Review of Theories and Their Applications.” Library & Information Science Research, vol. 26, no. 2, 2004, pp. 140–161., doi:10.1016/s0740-8188(04)00023-4.

[8] Gossen, Tatiana. “Search User Interface Design for Children.” Search Engines for Children, 2015, pp. 115–139., doi:10.1007/978-3-658-12069-6_7.

[9] Jochmann-Mannak, Hanna, et al. “How Interface Design and Search Strategy Influence Childrens Search Performance and Evaluation.” Web Design and Development, pp. 1332–1379., doi:10.4018/978-1-4666-8619-9.ch061.

[10] Darejeh. “A Review On User Interface Design Principles To Increase Software Usability For Users With Less Computer Literacy.” Journal of Computer Science, vol. 9, no. 11, Jan. 2013, pp. 1443–1450., doi:10.3844/jcssp.2013.1443.1450.

[11] Joshi, R., Shete, V. V., & Somani, S. B. (2015). Android Based Smart Learning and Attendance Management System. International Journal of Advanced Research in Computer and Communication Engineering, 4(6), 256-60.

[12] Char, C. A. (1983). Research and design issues concerning the development of educational software for children. US Department of Education, Office of Educational Research and Improvement, Educational Resources Information Center.

[13] Paananen, P., & Myllykoski, M. (2009). JamMo-Developmentally designed software for children’s mobile music-making. In ESCOM 2009: 7th Triennial Conference of European Society for the Cognitive Sciences of Music.

[14] Sim, G., MacFarlane, S., & Horton, M. (2005, June). Evaluating usability, fun and learning in educational software for children. In EdMedia: World Conference on Educational Media

and Technology (pp. 1180-1187). Association for the Advancement of Computing in Education (AACE).

[15] Cristina, T. G. (2011). Children Aimed Interfaces For Android Running Devices. Journal of Information Systems & Operations Management, 5(1), 161-174.

[16] Chiasson, S., & Gutwin, C. (2005). Design principles for children's technology. interfaces, 7, 28.

[17] Jochmann-Mannak, H. (2014). Websites for children: Search strategies and interface design-Three studies on children's search performance and evaluation.

[18] Xu, D. (2005). Tangible user interface for children-an overview. In Proc. of the UCLAN Department of Computing Conference.

[19] Federal Trade Commission. (2012). Mobile apps for kids: Current privacy disclosures are disappointing. Federal Trade Commission, Washington, DC. [Google Scholar].

[20] Federal Trade Commission. (2012). Mobile apps for kids: Disclosures still not making the grade.

[21] "MySQL," [Online]. Available: https://en.wikipedia.org/wiki/MySQL.

[22] "Android," [Online]. Available: https://developer.android.com/index.html.

[23] "Jsoup," [Online]. Available: https://jsoup.org/apidocs/overview-summary.html.

[24] "Casper," [Online]. Available: http://docs.casperjs.org/en/latest/quickstart.html.

[25] Li, G. and Seals, C. (2017). IW2K Android application for young kids. 2017 This is Research: Student Symposium. Auburn University. Retrieved from https://cws.auburn.edu/shared/files?id=159&filename=Full%20Program%20Abstracts.pdf [Online]. Available: https://jsoup.org/apidocs/overview-summary.html.