TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The popularity of smart mobile devices is growing fast. These digital devices represent a new generation of technological tools that offer remarkable access to content as well as opportunities for creative use even by young children. Most the best-selling paid apps in the education category are targeted towards children. At the same time, the educational value of those applications is difficult to be determined. Parents and educators, who are turning to those devices for the potential educational benefits they expect for their children and/or their students, have a limited number of tools with which to evaluate these apps. With regard to the literature review, we present the latest findings related to the real educational value of these ‘self-proclaimed’ educational apps. Our analysis concludes that while there are thousands of apps available today, choosing the most appropriate educational ones for children is difficult and problematic for both teachers and educators.

CHAPTER-1

1.1 INTRODUCTION

Children are in the midst of a vast, unplanned experiment, surrounded by digital technologies that were not available but 5 years ago (Hirsh-Pasek et al., 2015). Mobile touchscreen technologies also referred to as tablet technologies (Goodwin, 2012) are revolutionising the interactive digital experiences of young children (Chiong and Shuler, 2010; Papadakis and Kalogiannakis, 2010; Zaranis et al., 2013; Hung et al., 2014a; Hung et al., 2014b; Chen et al., 2014; Hwang et al., 2015). Owing to the growing inclusion of touchscreens, 'the days of the mouse keyboard and desktop graphical interface are numbered' (Waters, 2009). Young children explore and learn with mobile devices in ways that are natural to them (touch, repeat, trial and error) (Cohen et al., 2011, p.3). The reason is that touchscreen devices (tablets) are designed in such a way that even very young users can use them easily (Papadakis et al., 2017). Early research findings show that children younger than 2 years old can play and learn using mobile devices and/or multitouch displays (Michael Cohen Group LLC, 2011). Other studies revealed that children as young as two will naturally interact with a touchscreen, in the same way they will use natural instincts to play with a new toy (Sharkins et al., 2015). A study across Australia, New Zealand, the USA and Britain found more 2–5-year-olds are able to manipulate apps than tie their shoelaces or ride a bike (Grose, 2013). Preschool children do not need to develop the manual handling skills to use a separate keyboard and mouse required by general-purpose computers (Olney et al., 2008) in order to gain access to interactive content designed specifically for them (Plowman, 2012; Highfield and Goodwin, 2013; Papadakis et al., 2016a; Papadakis et al., 2016b; Papadakis et al., 2016c). At its best, touchscreen technology offers a mode of interactive experience that mirrors the child's natural constructivist learning (Papadakis and Orfanakis, 2014; Orfanakis and Papadakis, 2014). Ideally, smart devices accompanying applications (apps) can create exciting and effective learning environments for learning and instruction in early childhood (Goodwin, 2012; Papadakis et al., 2016b). Developers are increasingly creating educational apps that target this age group (Hiniker et al., 2015). A content analysis in the 'Education Section' of Apple's app store, conducted by Shuler in 2009, found that almost half (47%) of the top 100 selling apps targeted preschool and primary school age children. Mathematics and literacy apps were the most popular categories of apps (Shuler, 2009a). In a study conducted in 2012, the same researcher found that the number of apps which targeted preschool and primary school age children increased to 72% from 47% in 2009 (Shuler, 2012). In the Android Operating System, educational apps are among the top four categories of apps most accessed by the users and rank third in terms of a total number of paid apps being purchased by users (Avtar, 2014).

Currently, there are few examples of well-designed educational apps for young children (Hirsh-Pasek et al., 2015). The selection of the right app is very important as it can make the difference between the 'digital babysitter' and the tool to support children's learning and development (Goodwin, 2013). As many of the self-proclaimed educational apps are very entertainment-oriented due to several reasons they lack an educational impact on child cognitive development (Yusop and Razak, 2013). Although some parents are advanced and knowledgeable technology users themselves, this does not mean that they necessarily understand the full implications of the ICT products and services when used by young children (Ebbeck et al., 2016, p.2). At this point, the problem of the selection of the appropriate technologies such as apps that facilitate active and creative use by children (NAEYC, 2012) is becoming bigger and bigger for both parents and educators. As not all apps are of the same quality, it is also important to note, cost does not necessarily correlate with quality (Bouck et al., 2016). Given the absence of an industry standard or an official rating system for children's apps, websites or blogs are frequently consulted when choosing apps by parents and educators (Crescenzi-Lanna and Grané-Oró, 2016). However, this selection method is problematic too, and the reason is that the majority of those blogs and websites use a methodology for the assessment of 'educational' apps for children, which is lacking in terms of quality, and does not meet age-appropriate and other pedagogical standards (Crescenzi-Lanna and Grané-Oró, 2016).

1.2 Smart mobile devices in early childhood education

Smart mobile devices in early childhood education Although Apple in 2010 did not introduce the iPad as an educational tool, it has found its way into classrooms across the world. Digital devices are part of the culture in which children grow up; they permeate children's home and school life (Zaranis, 2013). As a result, our society puts pressure on educators and parents to provide digital literacy to young children (Pearsall, 2014). iPads and other forms of tablets are becoming commonplace in schools today, as they are regularly used for teaching. According to the World Bank, educational policy-makers are authorising the purchases of tablets in many education systems around the world as those devices are seen as powerful and iconic symbols of modernity within an education system (Trucano, 2015). In the UK, at the end of 2014, over 70% of all primary and secondary schools had tablet devices in their classrooms and 900,000 tablets were expected to be in schools by 2016 (Coughlan, 2014). In classroom settings, tablets are better than interactive whiteboards, whose fixed position often rendered them inaccessible for children reliant on wheelchairs and other physical supports (Flewitt et al., 2014). Lynch and Redpath (2014) in their research found that learners as young as 2 years old can use iPads independently and Beschorner and Hutchison (2013) also demonstrated that young learners are able to navigate the iPad on their own. As research suggests, children as young as 4 years old find it difficult to use a mouse and especially the use of the left button (Crook, 1992). If children cannot use educational technology effectively, they certainly cannot learn (Muller, 2002). Even 260 S. Papadakis and M.

Kalogiannakis 5-year-old children may encounter difficulty ending an activity using the mouse (Hourcade et al., 2004). The research of Abdul Aziz et al. (2013) shows that all children of age 4 and onwards can use the seven common gestures such as tap, drag-and-drop, slide, pinch, spread, spin/rotate and flick, which mobile applications generally require for usage. The rapid influx of new screen devices poses a special challenge for the early childhood community (Campaign for a Commercial-Free Childhood, Alliance for Childhood and Teachers Resisting Unhealthy Children's Entertainment, 2012). Early Childhood Educators (ECEs) are beginning to think about the role of this new technology in their classrooms and many preschool programs are beginning to purchase tablets, for classroom use (Beschorner and Hutchison, 2013). One of the reasons is that smart mobile devices provide 'significant opportunities for genuinely supporting differentiated, autonomous, and individualised learning' (Shuler, 2009b). Previous research supports that the developmentally appropriate use of technology can encourage the cognitive and social growth of young children (Beschorner and Hutchison, 2013). Touch devices present unique opportunities for enhancing young children's understandings of abstract concepts through the presentation of dynamic representations, opportunities for embodied learning and the inclusion of interactive elements (Goodwin, 2012). Yelland and Gilbert (2011) found in a study they conducted that the use of tablets across three different settings with children aged from 2 to 6 years of age represents a viable learning context in different ways for the children involved. Other emerging studies from the USA and Australia have shown that young children's learning can be enhanced using real ‘educational’ apps (Goodwin, 2013). These mobile devices and their accompanying apps can enhance knowledge acquisition through three different learning styles (VAK learning style): (a) visual, (b) acoustic, and (c) kinaesthetic (or physical, tactile) learning (Beeland, 2002). In 2012, the National Association for the Education of Young Children (NAEYC) stated that they encourage children from birth to 8 years of age to use tablets and age appropriate educational apps to support early literacy development (Ellingson, 2016). In ECE tablets can provide fun activities to allow children to articulate their creative perspectives; foster interest in the research process; and offer a route towards informed consent (Arnott et al., 2016). Findings also indicate that there are several types of learning that occur during app play. These include the tacit learning of the game and how it works; mastering of explicit learning tasks (e.g. matching, counting) embedded in the game narrative; and the use of skills and models learned and applied to other types of games and levels of play. Engaging with creative app activities often shifts the child’s focus away from the subjective experience of winning or losing to a personal best competition (Cohen et al., 2011). While there are thousands of apps available today, choosing the most appropriate educational ones for children is difficult and problematic for both teachers and educators.

1.3 The problems of evaluating educational apps

In general, obtaining the assessment of apps becomes an extremely difficult problem due to the huge number of self-proclaimed educational apps in Apple and Google digital stores (Levine, 2012). An obstacle in finding developmentally appropriate apps targeted at children, according to McKnight and Cassidy (2010), is the fact that even though children are considered a special user group, mobile device design guidelines formed from research based on adult participants may not transfer all that well to children (Lumsden, 2012). Software design principles that are intended for mature audiences cannot successfully be transferred to children's educational software (McKnight and Cassidy, 2010). Crescenzi-Lanna and Grané-Oró (2016), in a design analysis of 100 applications considered by educators and parents as potentially educational resources in 2014, found clear issues related to visual and interaction design, adaptability, layout and navigation, making it evident that there is a lack of quality and adaptation in terms of child development. A characteristic example is the use of textual messages that often accompany key information, instructions, and feedback in apps for preschoolers. Design for children is a unique realm of study as is design for older users (Stephanidis, 2009), and, as such, may interact with technologies in different ways to other users (Lumsden, 2012). Educational software addressed to ECE has certain characteristics compared to programs designed for use in general school education classrooms or to general-purpose software. This kind of software is based primarily on the use of graphics, video, animation effects, and sounds as well as the absence or minimum presence of texts (Nikiforidou and Pange, 2010). Additionally, according to Haugland (1999), educational software for preschoolers and kindergarten kids should give priority to the process rather than the product, providing opportunities and intrinsically motivate children to be more involved in their own learning (Carlton and Winsler, 1998). Although apps are available for preschoolers to build communication skills, pre-literacy skills, pre-math skills, as well as science skills (Heider and Jalongo, 2014), in fact, there are few examples of well-designed educational applications to get the kids learning, creating and playing (Michael Cohen Group LLC, 2011). It is common that text-heavy interfaces are included even in apps for preschoolers, Mobile educational applications for children 269 with the assumptions that parents will read the content to their children (Heider and Jalongo, 2014). Since 2010, Walker, realising the widespread adoption of mobile devices, and associated challenges of a vast array of apps, wondered what aspects are involved in making an app developmentally appropriate (Walker, 2010), concluding that there is not a single feature but a number of features, which vary and are evaluated differently by a parent, a teacher and a software developer (Walker, 2010). For example, he states that a website targeted at software developers places greater emphasis on the technical features of an app such as the platform's reliability and compatibility, putting less emphasis on other characteristics such as the Graphical User Interface (GUI) of the app. However, when assessing the usefulness of educational apps, these criteria represent the minimum, as the focus should be on the educational benefits of the app (Walker, 2010). Another website set priority to the GUI and the applications' adaptation to the user's needs. Walker considered the criteria related to the GUI and the application adaptability significant (Walker, 2010). The American non-profit organisation Sesame Workshop, formerly known as the Children's Television Workshop, states that it can be beneficial to children of preschool age, to get involved with apps which support learner's cognitive needs and development (scaffolding) (Sesame Workshop, 2012). Furthermore, independent organisations such as Common Sense Media, Kindertown, Yogi Play, Children's Technology Review, Parents' Choice and Appolicious have started describing highly effective learning apps' desirable characteristics (Guernsey et al., 2012). With reference to the above-mentioned subject, it is clearly understood that there is either no criteria or criterion to characterise a self-proclaimed educational application as developmentally appropriate or not. International websites that use a solid methodology in terms of quality while reviewing children's apps are the Mind Shift, Children's technology review, Common Sense Media, technology in education, and best apps for kids (Crescenzi-Lanna and Grané-Oró, 2016). Additionally, Lisa Guernsey and Michael Levine, co-authors of the book entitled Tap, Click, Read: Growing Readers in a World of Screens, suggested a series of curation sites for parents and educators which review educational apps. Those sites are Balefire Labs (balefirelabs.com), Children's Technology Review (childrenstech.com), Common Sense Media (commonsensemedia. org), Digital Storytime (digital-storytime.com), Graphite (graphite.org), Know What's Inside (knowwhatsinside.com), Parent's Choice Foundation (parents-choice.org), and Teachers with Apps (teacherswithapps.com) (Guernsey and Levine, 2015). Sesame Street, the non-profit organisation behind children's television programming with the landmark Sesame Street, published recently a checklist, which offers key findings from touchscreen studies and tips for designing and developing apps and e-books for preschoolers (Sesame Workshop, 2012). Although the paper is addressed mostly to evaluators and software developers, it could be also a useful guide both for parents and educator

1.4 The necessity of developing standards for mobile design and development.

The selection of appropriate mobile applications is of particular importance, as developmentally appropriate apps can support children's learning (Bennett, 2011). The quality of mobile applications targeted at early childhood depends on two conditions: taking into consideration the developmental stage of the child when formulating content and activities, and employing an interaction design that is appropriate to the child's cognitive and psychomotor development (Crescenzi-Lanna and Grané-Oró, 2016). According to Educational App Store (EAS) – an independent app marketplace – the following guidelines are found in developmentally appropriate apps (Parmar, 2012): · The apps are purposeful and educational. · The apps are interactive, transparent and intuitive. · The apps encourage the child to be in control. · The apps encourage collaboration between the teacher/facilitator/parent and child. Mobile educational applications for children 265 · The apps can strengthen home and school connections. · The apps must not contain any indication of violence or stereotyping. The great majority of teachers and parents cannot clearly define what constitutes a developmentally appropriate application (Parmar, 2012). There exist whole categories of very good apps that are fun to play with but that have no real educational goals (HirshPasek et al., 2015). As Goodwin (2013) states, it is important for parents and educators to note that just because an app is in the 'Education' section of the digital stores, that does not necessarily mean it is educational. Additionally, there is simply not the time, money, or resources available to evaluate each app as it enters the market (Hirsh-Pasek et al., 2015). Cohen et al. (2011, p.9) state that research findings indicate that except for the lack of real educational goals there are several barriers that inhibit use and learning with 'educational' apps. These include: · Apps' unclear, unfriendly or unresponsive user interface. · App gameplay that lacks reward or feedback. · Apps' obscure objectives. · Too many distractions. · Apps that lack 'palm rest', where buttons trigger themselves if accidentally touched within play area.

CHAPTER-2

OBJECTIVE

Smart mobile devices in early childhood education Although Apple in 2010 did not introduce the iPad as an educational tool, it has found its way into classrooms across the world. Digital devices are part of the culture in which children grow up; they permeate children's home and school life (Zaranis, 2013). As a result, our society puts pressure on educators and parents to provide digital literacy to young children (Pearsall, 2014). iPads and other forms of tablets are becoming commonplace in schools today, as they are regularly used for teaching. According to the World Bank, educational policy-makers are authorising the purchases of tablets in many education systems around the world as those devices are seen as powerful and iconic symbols of modernity within an education system (Trucano, 2015). In the UK, at the end of 2014, over 70% of all primary and secondary schools had tablet devices in their classrooms and 900,000 tablets were expected to be in schools by 2016 (Coughlan, 2014). In classroom settings, tablets are better than interactive whiteboards, whose fixed position often rendered them inaccessible for children reliant on wheelchairs and other physical supports (Flewitt et al., 2014). Lynch and Redpath (2014) in their research found that learners as young as 2 years old can use iPads independently and Beschorner and Hutchison (2013) also demonstrated that young learners are able to navigate the iPad on their own. As research suggests, children as young as 4 years old find it difficult to use a mouse and especially the use of the left button (Crook, 1992). If children cannot use educational technology effectively, they certainly cannot learn (Muller, 2002). Even 260 S. Papadakis and M. Kalogiannakis 5-year-old children may encounter difficulty ending an activity using the mouse (Hourcade et al., 2004). The research of Abdul Aziz et al. (2013) shows that all children of age 4 and onwards can use the seven common gestures such as tap, drag-and-drop, slide, pinch, spread, spin/rotate and flick, which mobile applications generally require for usage. The rapid influx of new screen devices poses a special challenge for the early childhood community (Campaign for a Commercial-Free Childhood, Alliance for Childhood and Teachers Resisting Unhealthy Children's Entertainment, 2012). Early Childhood Educators (ECEs) are beginning to think about the role of this new technology in their classrooms and many preschool programs are beginning to purchase tablets, for classroom use (Beschorner and Hutchison, 2013). One of the reasons is that smart mobile devices provide 'significant opportunities for genuinely supporting differentiated, autonomous, and individualised learning' (Shuler, 2009b). Previous research supports that the developmentally appropriate use of technology can encourage the cognitive and social growth of young children (Beschorner and Hutchison, 2013). Touch devices present unique opportunities for enhancing young children's understandings of abstract concepts through the presentation of dynamic representations, opportunities for embodied learning and the inclusion of interactive elements (Goodwin, 2012). Yelland and Gilbert (2011) found in a study they conducted that the use of tablets across three different settings with children aged from 2 to 6 years of age represents a viable learning context in different ways for the children involved. Other emerging studies from the USA and Australia have shown that young children's learning can be enhanced using real 'educational' apps (Goodwin, 2013). These mobile devices and their accompanying apps can enhance knowledge acquisition through three different learning styles (VAK learning style): (a) visual, (b) acoustic, and (c) kinaesthetic (or physical, tactile) learning (Beeland, 2002). In 2012, the National Association for the Education of Young Children (NAEYC) stated that they encourage children from birth to 8 years of age to use tablets and age appropriate educational apps to support early literacy development (Ellingson, 2016). In ECE tablets can provide fun activities to allow children to articulate their creative perspectives; foster interest in the research process;

and offer a route towards informed consent (Arnott et al., 2016). Findings also indicate that there are several types of learning that occur during app play. These include the tacit learning of the game and how it works; mastering of explicit learning tasks (e.g. matching, counting) embedded in the game narrative; and the use of skills and models learned and applied to other types of games and levels of play. Engaging with creative app activities often shifts the child's focus away from the subjective experience of winning or losing to a personal best competition (Cohen et al., 2011). While there are thousands of apps available today, choosing the most appropriate educational ones for children is difficult and problematic for both teachers and educators.

CHAPTER-3

LITERATURE SURVEY

1. Introduction

The mobile devices being utilitarian, user-friendly, accessible has made it the most popular and indispensable expedient for human essentials from the past few years (Malavolta et al., 2015). Mobile software developers' are driven to release software on time and within budget. Software estimation plays a pivotal role in providing the most accurate sizing figure for building confidence in developers and stakeholders relationship (Soares and Fagundes, 2017). Many approaches used for estimation of traditional software are adapted for mobile application development and testing (Wasserman, 2010). The testing phase of traditional software development proceeds through additional life cycle called Software Testing Life Cycle (STLC) (Katherine and Alagarsamy, 2012). According to Gao et al. (2014) mobile software testing are set of activities for mobile apps on mobile devices by exhausting definite software test techniques and tools in order to confirm quality in functionality, performance, and QoS, as well as features, like mobility, usability, interoperability, connectivity, security and privacy. The main phases of the testing process include test planning, test designing, test execution and test analysis (Farooq et al., 2011, Amen et al., 2015).

The estimation of effort for software testing comprises an estimation of test size, effort (Person per Hour), cost and entire schedule by means of several methods, tools and techniques (Abhilasha and Sharma, 2013). If effort, time and cost required to test the software can be anticipated, the testing resources can be systematically planned within a set target date to ensure lucrative culmination of projects. According to Zhu et al. (2008b), for estimating the test effort the major consideration is given on test designing (creation of test cases) and test execution.

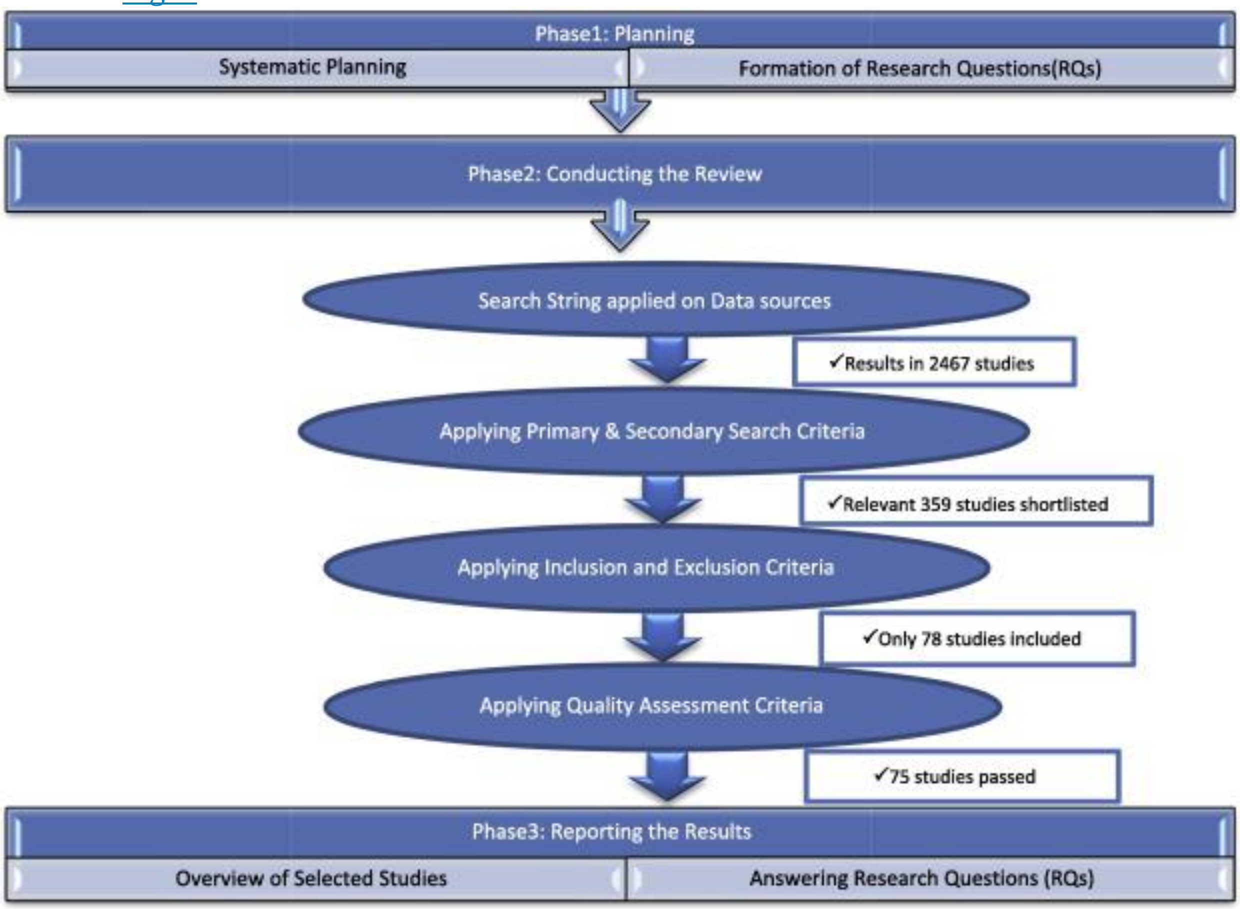
With the advent of Agile Software Development (ASD) (Usman et al., 2014) entire software development community has been driven by the adoption of agile methodology. The Agile approach to mobile application development states an iterative and incremental approach comprising self-organizing teams and cross-functioning teams working together to build the software (Kaur, 2016). The prominent existing agile mobile application development approaches are MOBILE-D, RaPiD7, Hybrid methodology, MASAM, Scrum with Lean Six Sigma (SLeSS) (Dewi and Nur Atiqah Sia, 2015). The Agile espousal to mobile application development is considered as a natural fit by many researchers (Cunha et al., 2011, Rahimian and Ramsin, 2008, Scharff and Verma, 2010). In an agile environment, development and testing are not considered separate phases as in traditional software development (Rahimian and Ramsin, 2008). The estimation of software in agile is prepared for both development and testing together. Estimation of effort in agile development is a new area of focus and very less work is reported literature (Aslam et al., 2017).

The significant contribution of the paper lies in examining the test effort estimation techniques for desktop/laptop software development and mobile software development. Further, the development and test effort estimation techniques are evaluated from two approaches of mobile application development process i.e., traditional software development and agile software development. Another major contribution is identifying the characteristics of mobile apps that make them distinct from traditional software.

Subsequently, the paper is divided as follows: Section 2 presents the research method comprising three phases of Systematic Literature Review (SLR). First and second phase is devoted to forming Research Questions (RQ) and finding relevant literature for studies. The results of the review are analyzed in the third phase i.e. result reporting phase of SLR, answering each Research Question (RQs). In section 3, discussions, research gaps and future directions are presented. Some threats to the validity of SLR are discussed in section 4 followed by conclusions in Section 5.

2. Research method

This section outlines the related literature and findings by the researchers which form the desired background for this research. The guidelines provided by Kitchenham and Charters (2007) are followed by conducting Systematic Literature Review (SLR). SLR is a research manner for carrying out a literature review in an orderly way of charting definite phases. SLR method uses three phases for performing literature review including Planning and specifying research questions, conducting the review that comprises an identification of search string and data sources, selecting studies, quality assessment, and data extraction and finally reporting the review. The steps followed for systematic literature review are undertaken in the following sections of this paper. The overview of the systematic literature review is shown in Fig. 1.



1. Download : Download high-res image (429KB)
2. Download : Download full-size image

Fig. 1. Systematic Literature Review (SLR) Phases by Kitchenham and Charters (2007).

2.1. Planning phase

For the smooth conduct of systematic literature review, proper planning is fundamental for smooth execution of SLR. The research questions derive from the entire systematic literature review planning phase.

Affirming the research questions is the vital part of any systematic review. In accordance with guidelines proposed by Petticrew and Roberts (2006) the criteria to frame research questions are based on PICOC (Population, Intervention, Comparison, Outcomes, and Context). If the research question is not outlined properly, the literature review may turn out off course. For this study, PICOC is defined as shown in Table 1.

Table 1. PICOC with description.

PICOC	Description
Population	Mobile Application projects.
Intervention	Test Effort estimation techniques/methods/process.
Comparison	Traditional software test effort estimation techniques with mobile apps testing estimation.
Outcomes	Mobile software test estimation techniques and characteristics of mobile apps that are considered important in development and testing estimation.
Context	Review the existing studies on test estimation of mobile Apps.

2.1.1. Research questions (RQs)

The review questions steer the entire systematic review methodology. The foremost aim of the review is to answer the following research question:-

RQ1. What are currently known software test estimation techniques for traditional applications?

RQ2. What are mobile development and testing estimation techniques?

This RQ can be subdivided into two sub-categories:-

> RQ2.a. What are mobile development and testing estimation techniques in a traditional software development environment?
>
> RQ2.b. What are mobile development and testing estimation Techniques in agile software development?

RQ3. How is the development and testing of mobile applications different from traditional software and what is different mobile app testing characteristics for estimation?

2.2. Conducting the review phase

2.2.1. Search strategy

The intent of the search strategy is to discover the studies that would assist in answering the RQs. The three phases of the search strategy comprise of identifying keywords and defining search strings, data sources selection and finally search process in data sources.

2.2.1.1. Identifying keywords and defining search strings

The foremost phase of the search strategy is to ascertain the search string. The search strategy is set up to describe search strings and primary data sources. The guidelines provided by Kitchenham and Charters (2007) were followed to define the search string by analyzing the main keywords in RQs, synonyms of the keywords and on any other spellings of the words. Following are the identified keywords and synonyms are shown in Table 2:

Table 2. List of keywords and synonyms.

Keywords	Synonymous Terms
Software	Software, project, system, application
Testing	Test, verification, validation
Effort	cost, resource, size, metric,
Estimation	Estimating, estimate, prediction, predicting, predict, assessment, forecasting, forecast, calculation, calculate, calculating, sizing, measure, measuring
Mobile Application	Mobile software, Mobile Apps, Mobile project
Development	Improvement, Progress
Method	Process, techniques, models, approaches
Agile	Scrum, XP, lean, crystal
Characteristics	Features, attribute, factors

Based on the identified keywords, the search string was obtained by joining synonymous terms using the 'OR', other keywords using logical 'AND' and wildcard character (*́). Here wildcard character represents 0, 1, or any number of alphanumeric characters. The search string is categorized in four ways according to the RQs formed. Table 3 lists the categories and corresponding search string.

Table 3. Search string categories.

Category	Based on RQs	Search string
1	Software Testing effort Estimation Techniques(RQ1)	("software" OR "project" OR "system" OR "application") AND ("Test*" OR "verification" OR "validation") AND ("Effort" OR "cost" OR "resource" OR "size" OR "metric") AND ("estimate*" OR "predict*" OR "assessment" OR "forecast*" OR "calculate*" OR "sizing" OR " measure*") AND ("Process" OR "techniques" OR "models" OR " approaches")
2	Mobile Application Development and Testing effort Estimation Techniques (RQ2.a.)	("Mobile Application" OR "Mobile software" OR " Mobile App" OR " Mobile project") AND ("Develop*" AND "Test*" OR "verification" OR "validation") AND ("Effort" OR "cost" OR "resource" OR "size" OR "metric") AND ("estimate*" OR "predict*" OR "assessment" OR "forecast*" OR "calculate*" OR "sizing" OR " measure*") AND ("Improvement" OR "Progress") AND ("Process" OR "techniques" OR "models" OR " approaches")

Category	Based on RQs	Search string
3	Agile Mobile Application Development and testing estimation(RQ2.b.)	("agile" OR "scrum" OR "XP" OR "lean" Or "crystal") AND ("Mobile Application" OR "Mobile software" OR "Mobile App" OR "Mobile project") AND ("Develop*") AND ("Test*" OR "verification" OR "validation") AND ("Effort" OR "cost" OR "resource" OR "size" OR "metric" AND ("estimate*" OR "predict*" OR "assessment" OR "forecast*" OR "calculate*" OR "sizing" OR "measure*")
4	Mobile Application characteristics(RQ3)	("Mobile Application" OR "Mobile software" OR " Mobile App" OR " Mobile project") AND ("Characteristics" OR "Features" OR "Attribute" OR "Factors")

2.2.1.2. Data sources

The digital databases that were used to search the keywords are SpringerLink, IEEE Xplore, ACM Digital Library, Elsevier Science Direct, Research Gate, CiteSeer, and InderScience.

2.2.1.3. Search process in data sources

The next phase is to apply the search string to the chosen electronic data sources to find all the entailed studies. This phase is divided into two sub-phases: primary and secondary search phase. In the Primary Search Phase, the electronic data sources identified are examined based on the search string defined earlier. Initially, a total of 2467 results was retrieved with the chosen search string. These results from data sources are monitored to include search string in title and abstracts. The search string is again refined each time to check the outcome and analyzed for better results. Additionally, results are restricted to peer-reviewed conference papers and journal papers. The duplicate titles and abstracts are removed. In the secondary search phase, a technique called snowball tracking is used for studying all the references of primary studies to exploit further studies and increase the chances of inclusion of important papers in the systematic literature review. Table 4 lists the refined results from data sources after primary and secondary search phase.

Table 4. Overview of search results.

Data Sources	Relevant Search Results
SpringerLink	53
IEEE Xplore	75
ACM Digital Library	57
Elsevier Science Direct	62
ResearchGate	83
Others(Google, ProQuest)	20
InderScience	5
CiteSeer	4
Total	**359**

2.2.2. Inclusion/Exclusion criteria for selecting studies

The results acquired through the various studies generated with the search string defined previously in the electronic databases were analyzed according to the Inclusion/Exclusion criteria. Table 5 enlists the search string category along with inclusion and exclusion criteria.

Table 5. Inclusion and exclusion criteria.

Search String Category	Included	Excluded
Category 1(RQ1)	Studies related to test estimation techniques for traditional software	Studies related to software development estimation techniques are excluded which do not feature estimation of testing phase for traditional software
Category2 (RQ2.a.)	Studies related to estimation methods for mobile application development and testing	Studies not related to mobile application development and testing estimation are removed
Category3(RQ2.b.)	Studies related to estimation methods in agile mobile application development and testing	Studies not related to agile mobile application development and testing are eliminated
Category 4(RQ3)	Studies that include characteristics of mobile application only	Studies having software characteristic and not mobile software/application characteristic are excluded
Applied to all Categories	Described in English Peer-reviewed papers are selected	Studies not defined in the English Language Not peer reviewed

Both the authors carried the paper selection process independently. The list of studies from primary and secondary search phase is reviewed by authors. The authors then analyze the studies independently and then mark them as In (Include), Un (Uncertain) and Ex (Exclude). The authors followed exclusion criterion through two stages:

1. First by reviewing the title and abstract. If title and abstract is in accordance to required information i.e. as per shown in Table5 for each RQ then,
2. Second stage is to review the full text, especially conclusion part.

The list of studies after marking each one with In, Un or Ex from authors is now reviewed collectively. In case of disagreement among authors, the decision on whether to exclude or include the study is based on decision rule table proposed by (Petersen et al., 2015). The decision rule table is exhibited in Table 6. The rules (A to F) against all cases of agreement and disagreement are shown in Table 6.The studies having Ex from both the authors are excluded right away following rule F in the decision table. Rest of the studies following under A to E is included for further analysis. The inclusion and exclusion criteria ended with 78 appropriate papers out of 359.

Table 6. Decision table rules in disagreement case followed from (Petersen et al., 2015).

	Author1			
		Include(In)	Uncertain(Un)	Exclude(Ex)
Author2	Include (In)	A	B	D
	Uncertain(Un)	B	C	E
	Exclude (Ex)	D	E	F

2.2.3. Quality assessment

To assess the quality of the shortlisted papers; a set of 7 questions is prepared to be answered for each shortlisted paper. The question can be answered as 'Y = 1′, 'M = 0.5 'or 'N = 0′. The score of 1 (Y = Yes) means the paper under consideration for quality assessment is explicitly well answered for a particular question (Q1-Q7); score 0.5 (M = Medium) means partially answered and score 0 (N = No) means not answered. The questionnaire was developed by using the guidelines defined by Kitchenham and Charters (2007). Following are the questions in the questionnaire:

Q1.Are the research motives clearly stated?

Q2.Was the study designed to achieve the aims?

Q3.Are the development and testing estimation techniques for mobile apps well defined?

Q4.Are the estimation accuracy measures soundly construed?

Q5.Is the research process documented sufficiently?

Q6.Are all research questions answered sufficiently?

Q7.Are the key findings specified plainly in rapport to creditability, validity, and reliability?

The authors have executed the quality assessment of all the carefully chosen primary studies. Due to the low quality, four papers are excluded from selected studies. Hereafter, 75 papers are designated to report four RQs. The final scores can be seen in Appendix A for 75 selected studies out of 78 along with its study IDs and references. The Study IDs are numbered according to RQ numbers where S stands for Study; RQ stands for Research Question followed by research question number and then identified study number. For answering an RQ1 total of 26 studies are devoted from 75 selected studies, 22 studies to RQ2 and finally 27 studies to RQ3.

2.2.4. Data extraction

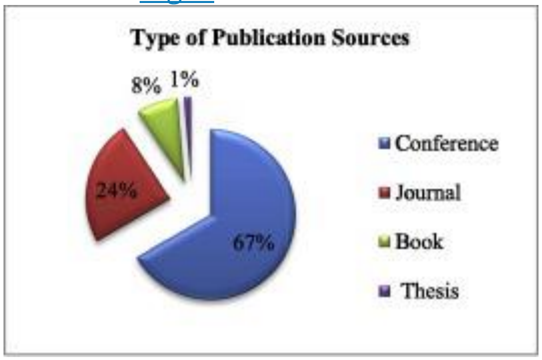
The data extraction phase elaborates the mining of data from the final selected studies that address the peculiarities of RQs. The data extraction for the finally chosen studies are done in an MS Excel sheet.

2.3. Results reporting

This section defines the results relayed to the systematic literature review questions. The results are presented in a tabular format for each study.

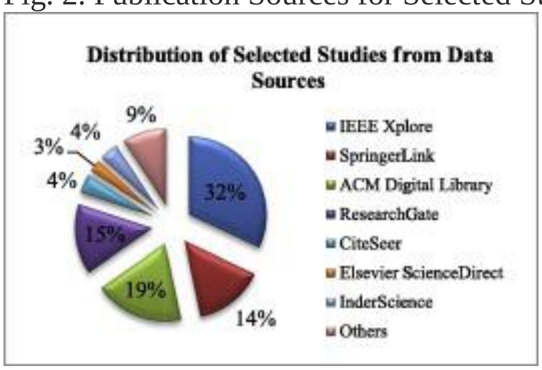
2.3.1. Selected studies overview

Fig. 2 shows the distribution of the chosen studies through the published sources. Out of the 75 studies, 24(32%) came from IEEExplore, 11 studies (14%) came from SpringerLink, ACM Digital Library 14(19%), 11(15%) from Research Gate, 3(4%) studies from CiteSeer, 2(3%) from Elsevier ScienceDirect, 3(4%) from InderScience and 7(9%) from others (ProQuest, GoogleScholar, TU/e Repository, scielo.org.co, SemanticScholar, IGI Global). The distribution of selected studies from different sources is shown in Fig. 3. Maximum papers are referred to the year 2014, 2015, 2016 and one each from 1999, 2001 and 2005. The distribution of selected studies according to the published year can be seen in Fig. 4.



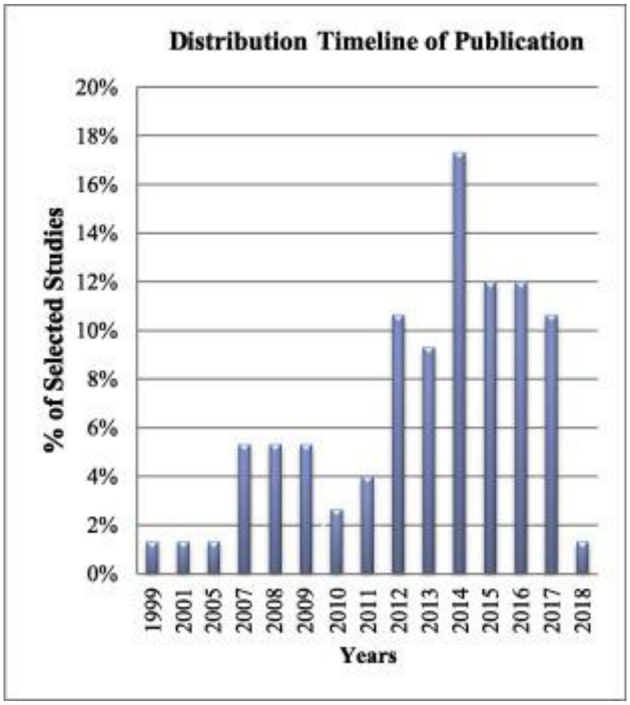
1. Download : Download high-res image (64KB)
2. Download : Download full-size image

Fig. 2. Publication Sources for Selected Studies.



1. Download : Download high-res image (109KB)
2. Download : Download full-size image

Fig. 3. Distribution of Selected Studies from Data Sources.



1. Download : Download high-res image (181KB)
2. Download : Download full-size image

Fig. 4. Distribution of Selected Studies (Year-Wise).

2.3.2. Results reporting on RQ1

To answer RQ1, out of seventy-five selected studies, twenty-six studies cover all the facets of RQ1.

Test Point Analysis (TPA) proposed by Veenendaal et al. (1999) is based on function point analysis used for estimating the functional size of software with additional attributes such as testing strategy and productivity.

Use Case Point Analysis (UCP) by Nageswaran (2001) examines testing characteristics and their complexity along with software development complexity.

Test Execution Points is a technique proposed by (E Aranha and Borba, 2007a, Aranha and Borba, 2007; Eduardo Aranha and Borba, 2007) based on test specification. The test cases are assigned execution points and then the effort is calculated

The model proposed by Abran et al. (2007) for estimating the test volume and effort, the functional requirements are taken as bases to form test estimation and then nonfunctional requirements are taken into consideration.

An approach by Kushwaha and Misra (2008) cognitive information complexity calculation and McCabe's Cyclomatic complexity measure is used to estimate the test execution effort.

Another approach by Zhu et al. (2008a) consist of three attributes for test effort estimation namely test case number, test execution complexity, and tester and then uses a historical database to assign effort.

Another approach by Zhu et al. (2008b) is an extension to existing UCP and considers test execution as a two-dimensional vector having testing experience and knowledge of the application.

The method suggested by Lazić and Mastorakis (2009) covers white box testing and test activities based on Software/System Test Point (STP) metric. The model is implemented on estimating object-oriented software projects by computing size and then applying three steps of COTECOMO model.

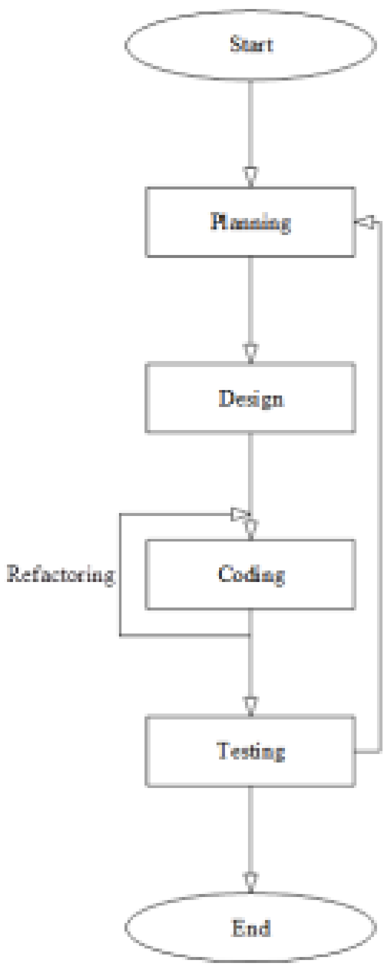
A model presented by Silva et al. (2009) is based on historical efficiency data of testers and functional test execution effort estimation and then the model accuracy is measured against different software as a case study.
Another model presented by Abhishek et al. (2010) studies the use of use case and neural network in Precoding phase and then in postcoding phase again neural network with variable, complexity and criticalness component as input is used to calculate test efforts.
In the model presented by Souza and Barbosa (2010) modified TPA is used by making it simpler and hence easy to use. In this model, there are two steps:-one followed by the test designer and second by the tester. Each of the steps has further sub-steps to follow to finally provide test effort estimation.
Aloka et al. (2011) presented an approach which is a combination of UCP, TPA, and particle swarm optimization (PSO) based approach to optimize the test effort estimation.
The approach proposed by Srivastava et al. (2012) is an adaptable model of UCP along with cuckoo search, a *meta*-heuristic approach, for test effort estimation.
A model is proposed by Sharma and Kushwaha (2013) based on SRS, then the complexity of requirements is computed. Requirement based test function points (RBTFP) is calculated based on the complexity of requirements and Technical and Complexity Factors (TEF).
Bhattacharya et al. (2012) proposed an approach which considers features of the software testing effort (STE) estimation by proposing a soft computing technique, Particle Swarm Optimization (PSO) along with COCOMO and test effort drivers into a single stage.
In the approach by Nguyen et al. (2013) the test case point is derived by measuring checkpoints, preconditions, test data and type of testing. The authors compared the proposed approach on two industrial case studies that used the experienced-based approach of testers.
Another heuristic approach by Srivastava et al. (2014) based on bat algorithm used along with existing test effort estimation techniques, UCP and TPA. Later the results obtained after applying bat algorithm are compared with those obtained from UCP and TPA to conclude that findings are improved and nearer to the actual effort using bat algorithm.
In the model proposed by Zapata-Jaramillo and Torres-Ricaurte (2014), the concept of a pre-conceptual schema is used which is a graphical technique to show domain knowledge in a natural language.
An approach by Hauptmann et al. (2014), the test suits are taken as an input and then a cost model is designed based on estimation done by an expert. In the cost model estimation is provided for test suite creation, test suite maintenance, and test execution.
The model presented by Srivastava (2015) uses fuzzy logic and fuzzy multiple linear regression techniques along with COCOMO-II to estimate software test effort. The problem with the model is usability while designing fuzzy rules. However, results produced using this model are better than existing methods.
The method proposed by Arumugam and Babu (2015) is based on the UCM (Use Case Model) and Function Point Analysis (FPA). The use case model is adapted to use case graph and later the edges acting as alternatives for required components are assigned weights. Then FPA is followed for assigning appropriate complexity weights to System Testing Technical Complexity Factors.

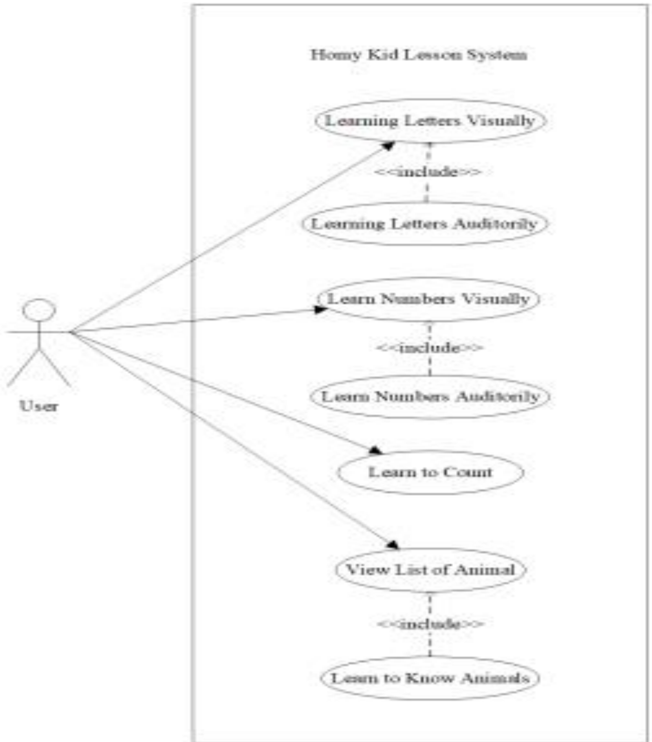
CHAPTER-4

Methodology

Mobile learning in pre-school education has potential benefits suggested by Mike et. al [5]. With the implementation of a systematic study, researchers wanted a positive impact on the use of mobile devices and ICT (information and communication technology) in *Corresponding Authors: Afan Galih Salman, Jl. KH Syahdan No. 9, Kemanggisan, Jakarta Barat, DKI Jakarta, 11480, asalman@binus.edu *Bayu Kanigoro, Jl. KH Syahdan No. 9, Kemanggisan, Jakarta Barat, DKI Jakarta, 11480, bkanigoro@binus.edu www.astesj.com https://dx.doi.org/10.25046/aj050526 212 Gulam et al. / Advances in Science, Technology and Engineering Systems Journal Vol. 5, No. 5, 212-216 (2020) kindergarten, not only of computers in the classroom but also on the technology used to assist teachers. The main concern is how to maximize the benefits of this instrument to improve the children’s experience of ICT in education. Mobile application for teaching mathematics has been proposed by Zaranis et. al [6] in kindergarten to helps students more active in learning and improve their initiative and opportunity for self-study. It shows that mobile application is very effective in increasing the motivation of kindergarten students for this is the position of kindergarten students to organize what, when, and where they learn. It becomes an easy platform for learning. The basic functional requirements for mobile apps for children with special needs is proposed by Kraleva et. al [7]. This issue became very important in the last years because an increase in the number of children with special needs on a worldwide scale is observed and at the same time the increase in the use of mobile technologies. Research design-based method proposed by

Soykan et. al [8] developed for students and teachers in teaching concept skills to develop disabled students and to examine the process of developing software based on the infrastructure of operant conditioning theory. Their design-based research method developed for students and teachers in teaching concept skills to developmentally disabled students and to examine the process of developing software based on the infrastructure of operant conditioning theory. Analysis of this type of application is done by analyzing similar apps that have been developed before. It gets an overview of these applications and gets additions so that apps can be made better. A comparison of these applications is shown in table 1. Table 1: Comparison with Similar Applications ABC Sound Apps Number Baby Flash Cards Apps Animal 123 Apps Homy Kid Lesson Apps Learn Alphabet ✓ ✗ ✗ ✓ Learn Number ✗ ✓ ✗ ✓ Learn Animals ✗ ✗ ✓ ✓ Animation ✗ ✗ ✗ ✓ Learn Music ✗ ✗ ✗ ✓ The development combines three different levels for the system integration, framework, and structure of the game. On a conceptual level, a game is considered as a system (i.e. a set of interrelated elements). A game is designed by specifying certain relevant factors, taking into account the two fundamental dimensions of space and time: the space dimension overs the static configuration of gaming locations (virtual) and includes associated objects, attributes, and relationships, and its evolution over time covers the game dynamics. On a technical level, the framework describes the basic architecture of the game development system which describes the system and its tools for developing the places, objects, actor roles, and scenarios of the video game. On a practical level, i.e. the structure of the game, the options offered to the players, and the multimedia representation of the game environment. The research method used in this study is shown in Figure 1. Figure 1: The research method used in this study. The proposed app uses a model of Agile Extreme Programming (XP) [9], which has stages of development as follows: Planning, that focus to get an overview of the functions and application software through communication with the user. This phase consists of several parts: literature studies, questionnaires, interviews, and analysis of similar applications





KIDDO				
	ABC Sound Apps Cards	Number Baby Flash	Apps Animal 123 Apps	Hommy Kid Lesson Apps
Learn Alphabet	✓	✓	✓	✓
Learn Number	✓	✓	✓	✓
Learn Animals	✓	✓	✓	✓
Video Learning	✓	✓	✓	✓
Look & Choose	✓	✓	✓	✓
Listen & Guess	✓	✓	✓	✓

CHAPTER-5

DATA FLOW DIAGRAM (DFD)

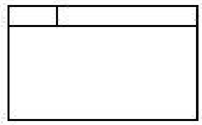
Data Flow Diagram (DFD) is an important technique for modeling a system's high level detail by showing how input data is transformed to output results through a sequence of functional transformations. DFDs reveal relationships among and between the various components in a program or system. DFDs consist of four major components: entities, processes, data stores and data flow.

A DFD represents the following:-

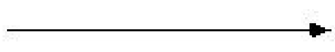
1. External devices sending and receiving data
2. Processes that change the data
3. Data flows themselves
4. Data storage locations

There are only four symbols used to write DFD as follows:-

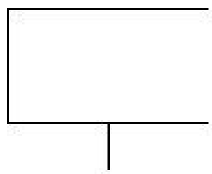
1. External Entities → Rectangular box
2. Data Flow → Arrow headed lines
3. Process → Bubble(Circle or round corner square)
4. Data Store → Narrow opened rectangle



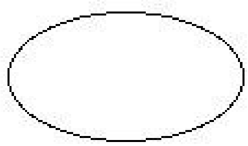
An activity. A title for the activity should be placed inside the rectangle.



Data flow. Arrow should be labeled with the name of the data moving between the activities.



A file or store. A straight line connects the file to the activity that accesses the file. The name of the file should be indicated within the rectangle.



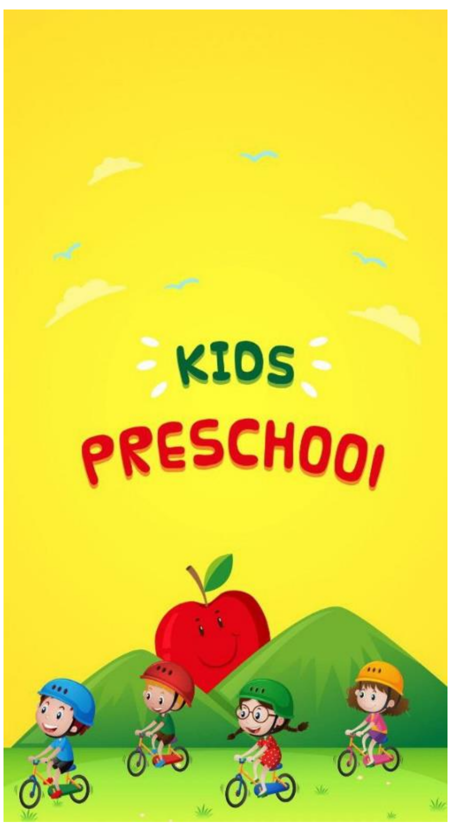
A source or recipient. A label should be placed within the oval.

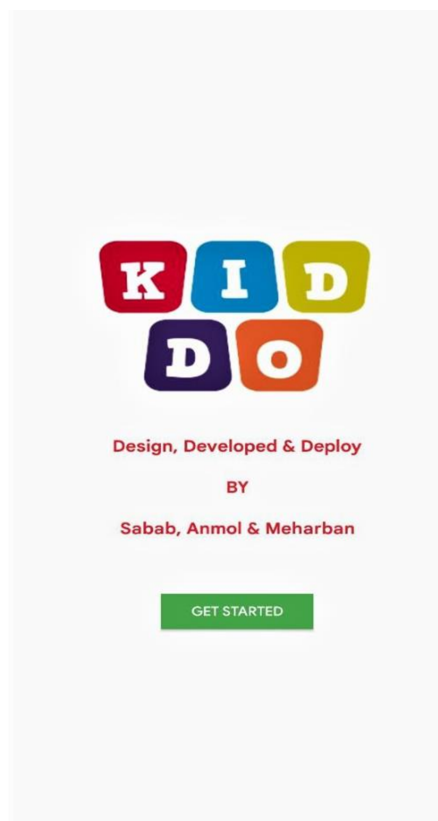
CHAPTER-6

DESIGN AND IMPLEMENTATION

6.1

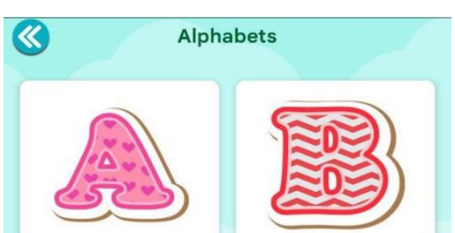
Mobile applications (apps) At the apex of the boom of smart mobile devices is the introduction of applications (apps) for tablets and smartphones (Hirsh-Pasek et al., 2015). Various researchers mention that besides the technological features of the smart mobile devices, these are in favour of young children mostly due to the existence of mobile applications specially Mobile educational applications for children 261 designed for these devices (Zaranis et al., 2013). Research has revealed that young children are very engaged with the apps and love to play with them for various amounts of time depending on their needs and interests and the content and structure of the app (Yelland and Gilbert, 2011). A mobile application is a computer program designed to run on mobile devices such as smartphones and tablet computers (Yusop and Razak, 2013; Wikipedia, 2016; Bouck et al., 2016). A mobile application may also be known as an app, iPhone app or smartphone app (Techopedia, 2016). Goodwin (2013) states that there are more than just ‘paid’ and ‘free’ apps in the app stores. There are in fact five different types of apps (see Figure 1). According to their pedagogical dimension, various researchers have tried to categorise apps designed for children in different categories depending on their open- or close-ended character, their level of activity, etc. Goodwin (2012, p.12) distinguishes apps into three different categories: · Constructive or ‘productivity’ apps, which are characterised by an open-ended design that allows users to create their own content or digital artefact using the app. Constructive apps are designed for creative expression. · Instructive apps have elements of ‘drill-and-practice’ design whereby the app delivers a predetermined ‘task’ which elicits a homogenous response from the user. These apps require minimal cognitive investment on behalf of the learner. Most game apps are classified as instructive apps. · Manipulable apps allow for guided discovery and experimentation within a predetermined context or framework. These apps require more cognitive involvement than instructive apps but less than constructive apps. Respectively, according to Cohen et al. (2011, p.9) the ‘world of apps’ currently designed for children includes three general types: gaming apps, creating apps and e-books. · In gaming apps, the activity includes a range of challenges, actions and reactions that lead to skill acquisition and achievement as levels are played and mastered. · In reading apps or e-books, the story or the reading of the story is the activity. Playful features or mini activities are integrated into a familiar schema of reading a book. The curriculum is in this context either explicit in the text or implicit and embedded in the activities. · Creating apps provide tools, workspace and activities (e.g. cupcakes, robots, painting, etc.). Research suggests that children learn best when they are cognitively active and engaged, when learning experiences are meaningful and socially interactive, and when learning is guided by a specific goal (see Figure 2) (Hirsh-Pasek et al., 2015). Additionally, children progress quickly from novice to mastery when using a well-designed app (Cohen et al., 2011, p.3). The majority of apps in today's marketplace can be considered part of the 'first wave' of the digital revolutio (Hirsh-Pasek et al., 2015). In this wave, apps are simply digital worksheets, games, and puzzles that have been reproduced in an e-format without any explicit consideration of how children learn or how the unique affordances of electronic media can be harnessed to support learning (Hirsh-Pasek et al., 2015). 262 S. Papadakis and M. Kalogiannakis Figure 1 The five different types of apps Source: Adapted from Goodwin (2013) Figure 2 The four pillars of an educational app Source: Adapted from Hirsh-Pasek et al. (2015)





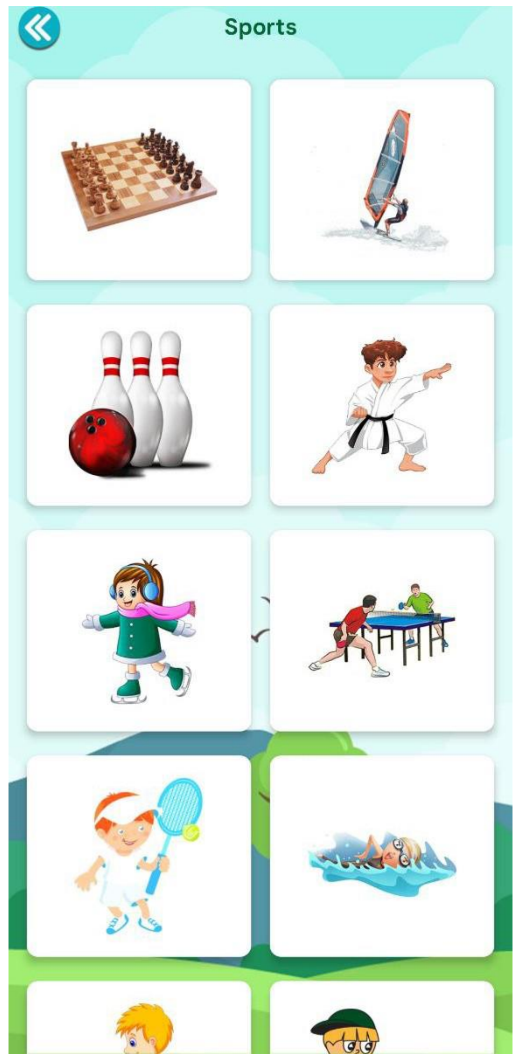

KIDDO
Let's Start Learning
SCHOOL
Video Learning
100 – 50 = ?
Look And Choose
A for Apple
B for Ball
C for Cat
D for Dog
Listen And Guess
Deer
Lion
Wolf
Tiger

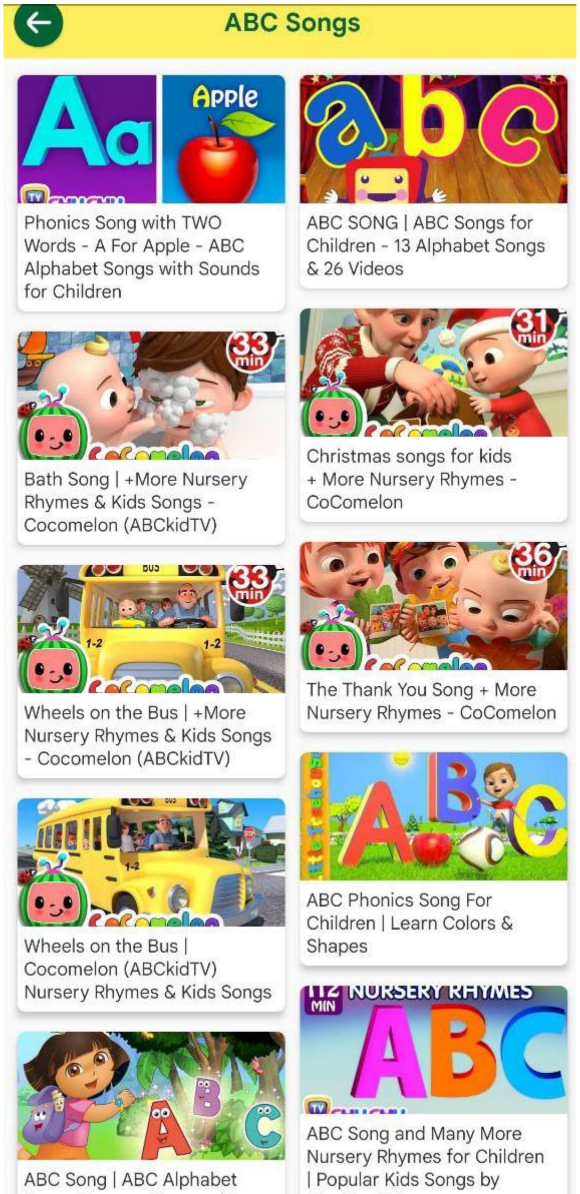

Learning
Alphabet
Number
Color
Shapes
Animal
Birds
Flower
Fruit
Month
Vigitable

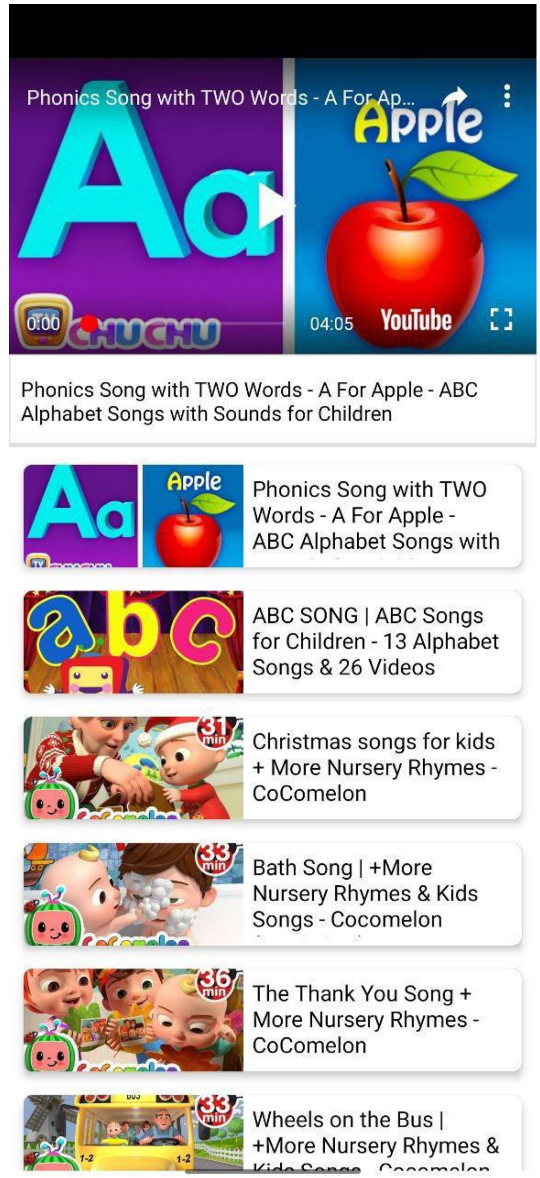

Alphabets


Animals


Look and Choose Quiz
Alphabet
Number
Color
Shapes
Animal
Birds
Flower
Fruit
Month
Vegitable


Sports


ABC Songs
Phonics Song with TWO Words - A For Apple - ABC Alphabet Songs with Sounds for Children
ABC SONG | ABC Songs for Children - 13 Alphabet Songs & 26 Videos
Bath Song | +More Nursery Rhymes & Kids Songs - Cocomelon (ABCkidTV)
Christmas songs for kids + More Nursery Rhymes - CoComelon
Wheels on the Bus | +More Nursery Rhymes & Kids Songs - Cocomelon (ABCkidTV)
The Thank You Song + More Nursery Rhymes - CoComelon
Wheels on the Bus | Cocomelon (ABCkidTV) Nursery Rhymes & Kids Songs
ABC Phonics Song For Children | Learn Colors & Shapes
ABC Song | ABC Alphabet
ABC Song and Many More Nursery Rhymes for Children | Popular Kids Songs by


Phonics Song with TWO Words - A For Ap...
0:00
04:05
YouTube
Phonics Song with TWO Words - A For Apple - ABC Alphabet Songs with Sounds for Children
Phonics Song with TWO Words - A For Apple - ABC Alphabet Songs with
ABC SONG | ABC Songs for Children - 13 Alphabet Songs & 26 Videos
Christmas songs for kids + More Nursery Rhymes - CoComelon
Bath Song | +More Nursery Rhymes & Kids Songs - Cocomelon
The Thank You Song + More Nursery Rhymes - CoComelon
Wheels on the Bus | +More Nursery Rhymes &

6.2 Tool & Technologies.

6.2.1 JAVA

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers *write once, run anywhere* (WORA),[17] meaning that compiled Java code can run on all platforms that support Java without the need to recompile.[18] Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most popular programming languages in use according to GitHub,[19][20] particularly for client–server web applications, with a reported 9 million developers.[21]

Java was originally developed by James Gosling at Sun Microsystems and released in May 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GPL-2.0-only license. Oracle offers its own HotSpot Java Virtual Machine, however the official reference implementation is the OpenJDK JVM which is free open-source software and used by most developers and is the default JVM for almost all Linux distributions.

As of March 2022, Java 18 is the latest version, while Java 17, 11 and 8 are the current long-term support (LTS) versions. Oracle released the last zero-cost public update for the legacy version Java 8 LTS in January 2019 for commercial use, although it will otherwise still support Java 8 with public updates for personal use indefinitely. Other vendors have begun to offer zero-cost builds of OpenJDK 8 and 11 that are still receiving security and other upgrades.

Oracle (and others) highly recommend uninstalling outdated and unsupported versions of Java, due to unresolved security issues in older versions.[22] Oracle advises its users to immediately transition to a supported version, such as one of the LTS versions (8, 11, 17).

6.2.2 ANDROID STUDIO.

Android Studio is the official[8] integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.[9] It is available for download on Windows, macOS and Linux based operating systems or as a subscription-based service in 2020.[10][11] It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013, at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014.[12] The first stable build was released in December 2014, starting from version 1.0.[13]

On May 7, 2019, Kotlin replaced Java as Google's preferred language for Android app development.[14] Java is still supported, as is C++.[15]

Features.

The following features are provided in the current stable version:[16][17]

- Gradle-based build support
- Android-specific refactoring and quick fixes
- Lint tools to catch performance, usability, version compatibility and other problems
- ProGuard integration and app-signing capabilities
- Template-based wizards to create common Android designs and components
- A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations[18]
- Support for building Android Wear apps
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine[19]
- Android Virtual Device (Emulator) to run and debug apps in the Android studio.

Android Studio supports all the same programming languages of IntelliJ (and CLion) e.g. Java, C++, and more with extensions, such as Go;[20] and Android Studio 3.0 or later supports Kotlin[21] and "all Java 7 language features and a subset of Java 8 language features that vary by platform version."[22] External projects backport some Java 9 features.[23] While IntelliJ states that Android Studio supports all released Java versions, and Java 12, it's not clear to what level Android Studio supports Java versions up to Java 12 (the documentation mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android.[24]

Once an app has been compiled with Android Studio, it can be published on the Google Play Store. The application has to be in line with the Google Play Store developer content

6.2.3 ANDROID SDK AND TOOLS.

The **Android SDK** is a software development kit that includes a comprehensive set of development tools.[2][3] These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, and Windows 7 or later. As of March 2015, the SDK is not available on Android itself, but software development is possible by using specialized Android applications.[4][5][6]

Until around the end of 2014, the officially-supported integrated development environment (IDE) was Eclipse using the Android Development Tools (ADT) Plugin.[7] As of 2015, Android Studio,[8] is the official IDE; however, developers are free to use others, but Google made it clear that ADT was officially deprecated since the end of 2015 to focus on Android Studio as the official Android IDE. Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely).[9][4][10]

Enhancements to Android's SDK go hand-in-hand with the overall Android platform development. The SDK also supports older versions of the Android platform in case developers wish to target their applications at older devices. Development tools are downloadable components, so after one has downloaded the latest version and platform, older platforms and tools can also be downloaded for compatibility testing.[11]

Android applications are packaged in .apk format and stored under `/data/app` folder on the Android OS (the folder is accessible only to the root user for security reasons). APK package contains .dex files[12] (compiled byte code files called Dalvik executables), resource files, etc.

Android SDK Platform Tools.

The Android SDK Platform Tools are a separately downloadable subset of the full SDK, consisting of command-line tools such as adb and fastboot.

Android Emulator 2.0:

- Performance improvements:
- Emulator now uses CPU acceleration on x86 emulator system images by default.
- Added SMP support to take advantage of host multi-core architecture when emulating Android 6.0 (API level 23) or higher, resulting in much better performance and speed than the physical counterpart. Also with SMP support, you can test apps that specifically target multi-core Android devices.
- Improved data and APK push-pull protocol between the Android Debug Bridge and devices running Android 5.0 (API level 21) or higher. See speed improvements up to five times faster than using a physical device.
- Extended UI controls and a floating toolbar provide easy access to features previously available only through the command line, such as taking screen captures, adjusting the battery level, rotating the screen, and managing virtual calls.
- Upload KML and GPX files to play back a set of custom location points.
- Dynamically resize the emulator by dragging a corner or zoom into the emulator window.
- Install APKs or add media files to the emulator's internal SD card by dragging and dropping files into the emulator window.
- Simulate multi-touch input. While interacting with the emulator screen, enter multi-touch mode by holding down the **Ctrl** key on Window/Linux, or **Command** key on Mac OSX.

CHAPTER-7

CODING AND IMPLEMENTATION

7.1.1

ACTIVITY HOME .XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:aapt="http://schemas.android.com/aapt"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    android:background="@drawable/bg_main"
    android:orientation="vertical">



    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="@dimen/header_top_padding">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="?actionBarSize">


            <ImageView
                android:id="@+id/imgBackButton"
                android:layout_width="45.0dip"
                android:layout_height="45.0dip"
                android:layout_margin="5.0dip"
                android:onClick="onClickBack"
                android:layout_centerVertical="true"
                android:src="@drawable/btn_back" />

            <TextView
                android:id="@+id/txtTitleSubHome"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerInParent="true"
                android:text="KIDDO"
                android:textColor="@color/colorBlue"
                android:textSize="20.0dip"
                android:textStyle="bold" />


            <TextView
                android:id="@+id/txtExamTitle"
                android:layout_width="0dp"
```

```xml
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:gravity="center_vertical"
                android:text="Learning"
                android:textColor="@color/colorBlue"
                android:textSize="20dp"
                android:textStyle="bold" />

            <ImageView
                android:id="@+id/imgSoundOn"
                android:layout_width="40dp"
                android:layout_height="40dp"
                android:layout_gravity="center"
                android:layout_margin="10dp"
                android:src="@drawable/btn_volume_on"
                android:visibility="gone" />

            <ImageView
                android:id="@+id/imgSoundOff"
                android:layout_width="40dp"
                android:layout_height="40dp"
                android:layout_gravity="center"
                android:layout_margin="10dp"
                android:src="@drawable/btn_volume_off"
                android:visibility="gone" />

        </RelativeLayout>

    </RelativeLayout>

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rvHomeCategories"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_margin="5dp"
        android:layout_weight="1" />

    <RelativeLayout
        android:id="@+id/llAdView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:gravity="center"
        android:visibility="visible" />


</LinearLayout>
```

7.1.2

ACTIVITY MAIN .XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="@dimen/header_top_padding">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="?actionBarSize">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerInParent="true"
                android:text="@string/app_name"
                android:textColor="@color/colorBlue"
                android:textSize="20.0dip"
                android:textStyle="bold" />

            <ImageView
                android:id="@+id/imgNotice"
                android:layout_width="40dp"
                android:layout_height="40dp"
                android:layout_gravity="center"
                android:layout_margin="10.0dip"
                android:src="@drawable/btn_disclaimer"
                android:visibility="gone" />

            <ImageView
                android:id="@+id/imgExitApp"
                android:layout_width="40.0dip"
                android:layout_height="40.0dip"
                android:layout_gravity="center"
                android:layout_margin="10.0dip"
                android:src="@drawable/btn_exit"
                android:visibility="gone" />

            <ImageView
                android:id="@+id/imgSetting"
                android:layout_width="40dp"
                android:layout_height="40dp"
                android:layout_marginRight="8dp"
                android:layout_alignParentRight="true"
                android:adjustViewBounds="true"
                android:onClick="onClickSetting"
                android:layout_centerVertical="true"
```

```
                android:src="@drawable/ic_setting" />
        </RelativeLayout>

    </RelativeLayout>

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rvCategory"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_margin="5.0dip"
        android:layout_weight="1.0" />

    <RelativeLayout
        android:id="@+id/llAdView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:gravity="center"
        android:visibility="visible" />



</LinearLayout>
```

7.1.3

ACTIVITY SETTING .XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorGray">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1">
    <RelativeLayout
        android:id="@+id/rltTop"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="@dimen/header_top_padding">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="?actionBarSize">
```

```
        <ImageView
            android:id="@+id/imgBackButton"
            android:layout_width="45.0dip"
            android:layout_height="45.0dip"
            android:layout_margin="5.0dip"
            android:onClick="onClickBack"
            android:layout_centerVertical="true"
            android:src="@drawable/btn_back" />

        <TextView
            android:id="@+id/txtTitleSubHome"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Settings"
            android:textColor="@color/colorBlue"
            android:textSize="20.0dip"
            android:textStyle="bold"
            android:layout_centerInParent="true"/>

    </RelativeLayout>

</RelativeLayout>

<RelativeLayout
    android:id="@+id/rltSound"
    android:layout_width="match_parent"
    android:layout_height="@dimen/menu_cell_height"
    android:layout_below="@+id/rltTop"
    android:layout_marginTop="16dp"
    android:background="@color/colorWhite">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/sound"
        android:layout_centerVertical="true"
        android:textSize="18dp"
        android:layout_marginLeft="@dimen/menu_cell_margin_left"
        android:textColor="@color/colorBlack"/>

    <Switch
        android:id="@+id/soundOnOff"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_centerVertical="true"
        android:theme="@style/SCBSwitch"
        android:checked="true"
        android:padding="16dp"/>

</RelativeLayout>

<LinearLayout
    android:id="@+id/rltMenu"
    android:layout_width="match_parent"
```

```xml
    android:layout_height="wrap_content"
    android:layout_below="@+id/rltSound"
    android:layout_marginTop="16dp"
    android:orientation="vertical"
    android:visibility="gone"
    android:background="@color/colorWhite">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="@dimen/menu_cell_height"
        android:text="@string/rate"
        android:textSize="18dp"
        android:textColor="@color/colorBlack"
        android:gravity="center_vertical"
        android:onClick="RateUs"
        android:layout_marginLeft="@dimen/menu_cell_margin_left"/>

    <View
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:background="@color/colorGray"/>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="@dimen/menu_cell_height"
        android:text="@string/share"
        android:textSize="18dp"
        android:textColor="@color/colorBlack"
        android:gravity="center_vertical"
        android:onClick="shareApp"
        android:layout_marginLeft="@dimen/menu_cell_margin_left"/>

    <View
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:background="@color/colorGray"/>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="@dimen/menu_cell_height"
        android:text="@string/send_feedback"
        android:textSize="18dp"
        android:textColor="@color/colorBlack"
        android:gravity="center_vertical"
        android:onClick="ContactUs"
        android:layout_marginLeft="@dimen/menu_cell_margin_left"/>

    <View
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:background="@color/colorGray"/>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="@dimen/menu_cell_height"
        android:text="@string/privacy_policy"
        android:textSize="18dp"
```

```
            android:textColor="@color/colorBlack"
            android:gravity="center_vertical"
            android:onClick="PrivacyPolicy"
            android:layout_marginLeft="@dimen/menu_cell_margin_left"/>

        <View
            android:layout_width="match_parent"
            android:layout_height="1dp"
            android:background="@color/colorGray"/>

        <TextView
            android:visibility="gone"
            android:layout_width="match_parent"
            android:layout_height="@dimen/menu_cell_height"
            android:text="More App"
            android:textSize="18dp"
            android:textColor="@color/colorBlack"
            android:gravity="center_vertical"
            android:onClick="MoreApp"
            android:layout_marginLeft="@dimen/menu_cell_margin_left"/>

    </LinearLayout>

    </RelativeLayout>
    <RelativeLayout
        android:id="@+id/llAdView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:gravity="center"
        android:visibility="visible" />


</LinearLayout>
```

7.1.4

ACTIVITY VIDEO LEARNING .XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">



    <RelativeLayout
        android:layout_width="match_parent"
```

```xml
        android:layout_height="wrap_content"
        android:background="@color/colorTheme"
        android:paddingTop="24dp">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="?actionBarSize">


            <ImageView
                android:id="@+id/imgBackButton"
                android:layout_width="45.0dip"
                android:layout_height="45.0dip"
                android:layout_margin="5.0dip"
                android:onClick="onClickBack"
                android:layout_centerVertical="true"
                android:src="@drawable/btn_back_150" />

            <TextView
                android:id="@+id/txtTitleSubHome"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Learning From Video Song"
                android:textColor="@color/colorBlue"
                android:textSize="20.0dip"
                android:textStyle="bold"
                android:layout_centerInParent="true"/>

        </RelativeLayout>

    </RelativeLayout>


    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rvVideoCategory"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:layout_margin="5dp" />

    <RelativeLayout
        android:id="@+id/llAdView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:gravity="center"
        android:visibility="visible" />


</LinearLayout>
```

7.2.1

HOME ACTIVITY.JAVA

```
package com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.activity;

import android.content.Context;
import android.content.Intent;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import androidx.recyclerview.widget.GridLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.view.View;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;

import com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.R;
import
com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.adapter.HomeCategoriesAd
apter;
import com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.utils.Utils;

import uk.co.chrisjenx.calligraphy.CalligraphyContextWrapper;

public class HomeActivity extends AppCompatActivity {

    Context context;
    RelativeLayout llAdView;
    LinearLayout llAdViewFacebook;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
        getSupportActionBar().hide();
        context = this;
        initDefine();
        llAdView = findViewById(R.id.llAdView);

    }



    public int[] mainCategoryList;
    String[] homeCategoryTitles;
    TextView txtExamTitle;
    int type=1;
    private void initDefine() {
        rvHomeCategories = findViewById(R.id.rvHomeCategories);
        txtExamTitle = findViewById(R.id.txtTitleSubHome);
        Intent intent=getIntent();
```

```
        type=intent.getIntExtra("Type",1);

        if(type==1){
            txtExamTitle.setText("Learning");
        }else if(type==2){
            txtExamTitle.setText("Look and Choose Quiz");
        }else if(type==3){
            txtExamTitle.setText("Listen and Guess");
        }
        homeCategoryTitles = new String[]{"Alphabets", "Numbers", "Colors",
"Shapes", "Animals", "Birds", "Flowers", "Fruits", "Month", "Vegetable", "Body
Parts", "Clothes", "Country", "Food", "Geometry", "House", "Jobs", "School",
"Sports", "Vehicle"};
        mainCategoryList = new int[]{R.drawable.home_alphabet,
R.drawable.home_number, R.drawable.home_color, R.drawable.home_shape,
R.drawable.home_animal, R.drawable.home_birds, R.drawable.home_flower,
R.drawable.home_fruits, R.drawable.home_month, R.drawable.home_vegetable,
R.drawable.home_body_parts, R.drawable.home_clothes, R.drawable.home_country,
R.drawable.home_food, R.drawable.home_geometry, R.drawable.home_house,
R.drawable.home_jobs, R.drawable.home_school, R.drawable.home_sports,
R.drawable.home_vehicle};
        setRvAdapter();
    }


    HomeCategoriesAdapter homeCategoriesAdapter;
    RecyclerView rvHomeCategories;

    private void setRvAdapter() {
        GridLayoutManager gridLayoutManager = new GridLayoutManager(context, 2,
GridLayoutManager.VERTICAL, false);
        rvHomeCategories.setLayoutManager(gridLayoutManager);
        homeCategoriesAdapter = new HomeCategoriesAdapter(context,
mainCategoryList,homeCategoryTitles,type);
        rvHomeCategories.setAdapter(homeCategoriesAdapter);
    }

    public void onClickBack(View view) {
        finish();
    }
}
```

7.2.2

MAIN ACTIVITY.JAVA

```java
package com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.activity;

import android.content.Context;
import android.content.Intent;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.pm.ApplicationInfo;
import android.content.pm.PackageManager;
import android.os.Bundle;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.util.Log;
import android.view.View;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.messaging.FirebaseMessaging;
import com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.R;
import
com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.adapter.HomeAdapter;
import
com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.customclasses.Constant;
import
com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.database.DatabaseHelper;
import
com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.interfaces.CallbackListe
ner;
import com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.utils.Utils;

import uk.co.chrisjenx.calligraphy.CalligraphyContextWrapper;

public class MainActivity extends AppCompatActivity implements CallbackListener {

    Context context;
    DatabaseHelper databaseHelper;
    RelativeLayout llAdView;
    LinearLayout llAdViewFacebook;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getSupportActionBar().hide();
        context=this;
        initDefine();
        databaseHelper = new DatabaseHelper(context);
        try {
```

```java
            databaseHelper.createDataBase();
        } catch (Exception e) {
            e.printStackTrace();
        }
        //successCall();
        try {
            //subScribeToFirebaseTopic();
        } catch (Exception e) {
            e.printStackTrace();
        }
        llAdView = findViewById(R.id.llAdView);


    }

    private void subScribeToFirebaseTopic() {
        FirebaseMessaging.getInstance().subscribeToTopic("kids_play_topic")
                .addOnCompleteListener(new OnCompleteListener<Void>() {
                    @Override
                    public void onComplete(@NonNull Task<Void> task) {
                        if (!task.isSuccessful()) {
                            Log.e("subScribeFirebaseTopic", ": Fail");
                        } else {
                            Log.e("subScribeFirebaseTopic", ": Success");
                        }
                    }
                });

    }


    private void successCall() {
        if (Utils.isNetworkConnected(this)) {
            if (Constant.ENABLE_DISABLE.equals(Constant.ENABLE)) {

                Utils.setPref(MainActivity.this, Constant.AD_TYPE_FB_GOOGLE,
Constant.AD_TYPE_FACEBOOK_GOOGLE);
                Utils.setPref(MainActivity.this, Constant.FB_BANNER,
Constant.FB_BANNER_ID);
                Utils.setPref(MainActivity.this, Constant.FB_INTERSTITIAL,
Constant.FB_INTERSTITIAL_ID);
                Utils.setPref(MainActivity.this, Constant.GOOGLE_BANNER,
Constant.GOOGLE_BANNER_ID);
                Utils.setPref(MainActivity.this, Constant.GOOGLE_INTERSTITIAL,
Constant.GOOGLE_INTERSTITIAL_ID);
                Utils.setPref(MainActivity.this, Constant.STATUS_ENABLE_DISABLE,
Constant.ENABLE_DISABLE);

                setAppAdId(Constant.GOOGLE_ADMOB_APP_ID);
            } else {
                Utils.setPref(MainActivity.this, Constant.STATUS_ENABLE_DISABLE,
Constant.ENABLE_DISABLE);
            }
        } else {
            Utils.openInternetDialog(this, true,this);
        }
    }
```

```
    public void setAppAdId(String id) {
        try {
            ApplicationInfo applicationInfo =
getPackageManager().getApplicationInfo(getPackageName(),
PackageManager.GET_META_DATA);
            Bundle bundle = applicationInfo.metaData;
            String beforeChangeId =
bundle.getString("com.google.android.gms.ads.APPLICATION_ID");
            Log.e("TAG", "setAppAdId:BeforeChange::::  " + beforeChangeId);

applicationInfo.metaData.putString("com.google.android.gms.ads.APPLICATION_ID", id);
            String AfterChangeId =
bundle.getString("com.google.android.gms.ads.APPLICATION_ID");
            Log.e("TAG", "setAppAdId:AfterChange::::  " + AfterChangeId);
        } catch (PackageManager.NameNotFoundException | NullPointerException e) {
            e.printStackTrace();
        }

    }


    public int[] arrOfCategory;
    private void initDefine() {
        rvCategory=findViewById(R.id.rvCategory);
        arrOfCategory = new int[]{R.drawable.card_one, R.drawable.card_two,
R.drawable.card_three, R.drawable.card_four};
        setRvAdapter();
    }


    public void onClickSetting(View view){
        Intent intent= new Intent(MainActivity.this,SettingActivity.class);
        startActivityForResult(intent,111);
    }

    HomeAdapter homeAdapter;
    RecyclerView rvCategory;
    private void setRvAdapter() {
        LinearLayoutManager linearLayoutManager=new
LinearLayoutManager(context,LinearLayoutManager.VERTICAL,false);
        rvCategory.setLayoutManager(linearLayoutManager);
        homeAdapter=new HomeAdapter(context,arrOfCategory);
        rvCategory.setAdapter(homeAdapter);
    }

    @Override
    public void onSuccess() {

    }

    @Override
    public void onCancel() {

    }
```

```
    @Override
    public void onRetry() {

    }
}
```

7.2.3

LIST VIDEO ACTIVITY.JAVA

```
package com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.activity.video;

import android.content.Context;
import android.content.Intent;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import androidx.recyclerview.widget.RecyclerView;
import androidx.recyclerview.widget.StaggeredGridLayoutManager;
import android.view.View;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;

import com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.R;
import
com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.adapter.VideoListAdapter
;
import
com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.database.DatabaseHelper;
import com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.model.ModelVideo;
import com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.utils.Utils;

import java.util.ArrayList;

import uk.co.chrisjenx.calligraphy.CalligraphyContextWrapper;

public class ListVideoActivity extends AppCompatActivity {

    Context context;
    DatabaseHelper databaseHelper;
    RelativeLayout llAdView;
    LinearLayout llAdViewFacebook;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_list_video);
        getSupportActionBar().hide();
        context=this;
```

```
        initDefine();
    }



    private void initDefine() {
        rvVideoList=findViewById(R.id.rvVideoList);
        TextView txtTitleSubHome=findViewById(R.id.txtTitleSubHome);
        Intent intent=getIntent();
        txtTitleSubHome.setText(intent.getStringExtra("Category"));
        setRvVideoListAdapter();
        llAdView = findViewById(R.id.llAdView);

    }

    RecyclerView rvVideoList;
    ArrayList<ModelVideo> arrOfVideoList;
    VideoListAdapter videoListAdapter;
    private void setRvVideoListAdapter() {
        databaseHelper=new DatabaseHelper(context);
        arrOfVideoList=databaseHelper.getVideoDetails();
        StaggeredGridLayoutManager staggeredGridLayoutManager=new
StaggeredGridLayoutManager(2,StaggeredGridLayoutManager.VERTICAL);
        rvVideoList.setLayoutManager(staggeredGridLayoutManager);
        videoListAdapter=new VideoListAdapter(context,arrOfVideoList);
        rvVideoList.setAdapter(videoListAdapter);
    }

    public void onClickBack(View view) {
        finish();
    }
}
```

7.2.4

ONBOARDING ACTIVITY.JAVA

```java
package com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.activity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.R;

public class OnboardingActivity extends AppCompatActivity {
    TextView txt;
    Button btn_started;
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_onboarding_screen);
        getSupportActionBar().hide();
        txt=findViewById(R.id.txt);
        btn_started=findViewById(R.id.btn_started);
        btn_started.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(OnboardingActivity.this, MainActivity.class);
                startActivity(i);
                finish();
            }
        });
    }
}
```

7.3.1

ADAPTER CLASS.

HOME ADAPTER

```
package com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.adapter;

import android.annotation.SuppressLint;
import android.content.Context;
import android.content.Intent;
import androidx.annotation.NonNull;
import androidx.cardview.widget.CardView;
import androidx.recyclerview.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;

import com.bumptech.glide.Glide;
import com.bumptech.glide.load.engine.DiskCacheStrategy;
import com.bumptech.glide.request.RequestOptions;
import com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.R;
import
com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.activity.HomeActivity;
import
com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.activity.video.VideoLear
ningActivity;

/**
 * Created by Naynesh Patel on 06-Feb-19.
 */
public class HomeAdapter extends RecyclerView.Adapter<HomeAdapter.ViewHolder> {

    Context context;
    int[] arrOfCategory;

    public HomeAdapter(Context context, int[] arrOfCategory) {
        this.context = context;
        this.arrOfCategory = arrOfCategory;
    }

    public int getItemCount() {
        return arrOfCategory.length;
    }


    @NonNull
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {
        return new
ViewHolder(LayoutInflater.from(context).inflate(R.layout.card_list_start, viewGroup,
false));
    }

    public void onBindViewHolder(@NonNull ViewHolder viewHolder,
```

```
@SuppressLint("RecyclerView") final int i) {

        Glide.with(context).load(arrOfCategory[i]).apply(new
RequestOptions().diskCacheStrategy(DiskCacheStrategy.ALL)).into(viewHolder.imgThumbn
ail);

        viewHolder.cVHomeCategories.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                switch (i) {
                    case 0:
                        Intent intent1 = new Intent(context, HomeActivity.class);
                        intent1.putExtra("Type", 1);
                        context.startActivity(intent1);
                        break;
                    case 1:
                        Intent intent2 = new Intent(context,
VideoLearningActivity.class);
                        context.startActivity(intent2);
                        break;
                    case 2:
                        Intent intent3 = new Intent(context, HomeActivity.class);
                        intent3.putExtra("Type", 2);
                        context.startActivity(intent3);
                        break;
                    case 3:
                        Intent intent4 = new Intent(context, HomeActivity.class);
                        intent4.putExtra("Type", 3);
                        context.startActivity(intent4);
                        break;
                }
            }
        });
    }


    class ViewHolder extends RecyclerView.ViewHolder {
        CardView cVHomeCategories;
        ImageView imgThumbnail;

        ViewHolder(@NonNull View view) {
            super(view);
            this.cVHomeCategories = view.findViewById(R.id.cVHomeCategories);
            this.imgThumbnail = view.findViewById(R.id.imgThumbnail);
        }
    }


}


//        viewHolder.cVHomeCategories.setOnClickListener(new OnClickListener() {
//            public void onClick(View view) {
//                switch (i) {
//                    case null:
//                        StartActivity.this.startActivity(new
Intent(StartActivity.this.getApplicationContext(), HomeActivity.class));
```

```
//                          return;
//                      case 1:
//                          if (StartActivity.this.isNetworkConnected() != null) {
//                              StartActivity.this.startActivity(new
Intent(StartActivity.this.getApplicationContext(), VideoLearningActivity.class));
//                              StartActivity.requestInterstitialAds();
//                              return;
//                          }
//                          Toast.makeText(StartActivity.this.getApplicationContext(),
"Turn on internet connection", 1).show();
//                          return;
//                      case 2:
//                          StartActivity.this.startActivity(new
Intent(StartActivity.this.getApplicationContext(),
ExamMainActivity.class).putExtra("index", 2));
//                          return;
//                      case 3:
//                          StartActivity.this.startActivity(new
Intent(StartActivity.this.getApplicationContext(),
ExamMainActivity.class).putExtra("index", 3));
//                          return;
//                      default:
//                          return;
//                  }
//              }
//          });
```

7.3.2

EXAM QUESTION ADAPTER

```
package com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.adapter;

import android.content.Context;
import android.speech.tts.TextToSpeech;
import androidx.annotation.NonNull;
import androidx.cardview.widget.CardView;
import androidx.recyclerview.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.animation.AnimationUtils;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.R;
import
com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.customclasses.AppControl
;
import
com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.customclasses.Constant;
import
com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.model.LearningDataModel;
import com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.utils.Utils;

import java.util.ArrayList;

/**
 * Created by Naynesh Patel on 06-Feb-19.
 */
public class ExamQuestionAdapter extends
RecyclerView.Adapter<ExamQuestionAdapter.ViewHolder> {


    Context context;
    ArrayList<LearningDataModel> examQuestionAnswerList;
    LearningDataModel learningDataModel;

    public ExamQuestionAdapter(Context context, ArrayList<LearningDataModel>
examQuestionAnswerList, LearningDataModel learningDataModel) {
        this.context = context;
        this.examQuestionAnswerList = examQuestionAnswerList;
        this.learningDataModel = learningDataModel;
    }


//    TextToSpeech textToSpeech;

    class ViewHolder extends RecyclerView.ViewHolder {
        CardView cardView;
        LinearLayout lloutExamAnswer;
        TextView tVExamAnswer;
```

```
        ViewHolder(@NonNull View view) {
            super(view);
            this.cardView = (CardView) view.findViewById(R.id.cvCardSubHome);
            this.tVExamAnswer = (TextView) view.findViewById(R.id.tVExamAnswer);
            this.lloutExamAnswer = (LinearLayout)
view.findViewById(R.id.lloutExamAnswer);
//              textToSpeech = new TextToSpeech(context, new
TextToSpeech.OnInitListener() {
//                  @Override
//                  public void onInit(int status) {
//                      if (status != TextToSpeech.ERROR) {
//                          textToSpeech.setLanguage(Locale.UK);
//                      }
//                  }
//              });
        }
    }


    @NonNull
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {
        return new
ViewHolder(LayoutInflater.from(viewGroup.getContext()).inflate(R.layout.card_exam_an
swer, viewGroup, false));
    }



    public void onBindViewHolder(@NonNull final ViewHolder viewHolder, final int i)
{
        viewHolder.tVExamAnswer.setText(examQuestionAnswerList.get(i).showTitle);
        viewHolder.cardView.startAnimation(AnimationUtils.loadAnimation(context,
R.anim.bubble_anim));
        viewHolder.cardView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if
(learningDataModel.showTitle.equals(examQuestionAnswerList.get(i).showTitle)) {
                    Toast.makeText(context, "Correct Answer",
Toast.LENGTH_SHORT).show();

viewHolder.lloutExamAnswer.setBackgroundColor(context.getResources().getColor(R.colo
r.colorCorrect));
                    if (Utils.getPref(Constant.SOUND,true)) {
                        AppControl.textToSpeech.speak("Correct Answer",
TextToSpeech.QUEUE_FLUSH, null);
                    }
                } else {
                    Toast.makeText(context, "Wrong Answer",
Toast.LENGTH_SHORT).show();

viewHolder.lloutExamAnswer.setBackgroundColor(context.getResources().getColor(R.colo
r.colorWrong));

viewHolder.tVExamAnswer.setTextColor(context.getResources().getColor(R.color.colorWh
ite));
```

```
                    if (Utils.getPref(Constant.SOUND,true)){
                        AppControl.textToSpeech.speak("Wrong Answer",
TextToSpeech.QUEUE_FLUSH, null);
                    }
                }
            }
        });
    }

    public int getItemCount() {
        return examQuestionAnswerList.size();
    }
}
```

7.3.3

LIST VIDEO ADAPTER

```
package com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.adapter;

import android.content.Context;
import android.content.Intent;
import androidx.annotation.NonNull;
import androidx.cardview.widget.CardView;
import androidx.recyclerview.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import com.bumptech.glide.Glide;
import com.bumptech.glide.load.engine.DiskCacheStrategy;
import com.bumptech.glide.request.RequestOptions;
import com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.R;
import
com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.activity.video.VideoPlay
Activity;
import
com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.customclasses.Constant;
import com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.model.ModelVideo;

import java.util.ArrayList;

/**
 * Created by Naynesh Patel on 12-Feb-19.
 */
public class VideoListAdapter extends
RecyclerView.Adapter<VideoListAdapter.ViewHolder> {

    Context context;
```

```
    ArrayList<ModelVideo> arrOfVideoList;

    public VideoListAdapter(Context context, ArrayList<ModelVideo> arrOfVideoList) {
        this.context = context;
        this.arrOfVideoList = arrOfVideoList;
    }

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {
        return new
ViewHolder(LayoutInflater.from(context).inflate(R.layout.card_item_video_list,
viewGroup, false));
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder viewHolder, final int i) {

        Glide.with(context)
                .load(arrOfVideoList.get(i).getVideoThumb())
                .apply(new
RequestOptions().diskCacheStrategy(DiskCacheStrategy.ALL))
                .into(viewHolder.imgThumbnail);

        viewHolder.txtVideoTitle.setText(arrOfVideoList.get(i).getVideoTitle());

        viewHolder.cardView.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                Constant.VIDEO_ID = arrOfVideoList.get(i).getVideoId();
                context.startActivity(new Intent(context,
VideoPlayActivity.class).putExtra("Position",i).putExtra("ArrayOfVideo",arrOfVideoLi
st));
            }
        });
    }

    @Override
    public int getItemCount() {
        return arrOfVideoList.size();
    }

    public class ViewHolder extends RecyclerView.ViewHolder {
        CardView cardView;
        ImageView imgThumbnail;
        TextView txtVideoTitle;

        public ViewHolder(@NonNull View view) {
            super(view);
            txtVideoTitle = view.findViewById(R.id.txtVideoDescription);
            cardView = view.findViewById(R.id.cardViewMain);
            imgThumbnail = view.findViewById(R.id.ivThumbnailView);
        }
    }
}
```

7.4.1

CUSTOM CLASS.

APP CONTROL

```
package com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.customclasses;

import android.app.Activity;
import android.app.Application;
import android.content.Context;
import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.util.Log;
import android.widget.Toast;

import com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.R;
import com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.utils.Utils;

import java.util.Locale;

import uk.co.chrisjenx.calligraphy.CalligraphyConfig;

/**
 * Created by Naynesh Patel on 06-Mar-19.
 */
public class AppControl extends Application {
    Context context;
    public static TextToSpeech textToSpeech;
    @Override
    public void onCreate() {
        super.onCreate();
        context=this;
        new Utils(context);
        CalligraphyConfig.initDefault(new CalligraphyConfig.Builder()
                .setDefaultFontPath("Fon/OhWhale.ttf")
                .setFontAttrId(R.attr.fontPath)
                .build()
        );
        textToSpeech = new TextToSpeech(context, new TextToSpeech.OnInitListener() {
            @Override
            public void onInit(int status) {
                if (status != TextToSpeech.ERROR) {
                    textToSpeech.setLanguage(Locale.UK);
//                      textToSpeech.setLanguage(Locale.forLanguageTag(""));
                }
            }
        });
    }


}
```

7.4.2

CONSTANT

```
package com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.customclasses;

import org.jetbrains.annotations.Nullable;

/**
 * Created by Naynesh Patel on 12-Feb-19.
 */
public class Constant {
    public static String DB_PATH =
"data/data/com.kidslearning.kidsplay.kidsgames.kidseducation.preschool/databases/";
    public static String DATABASE_NAME = "preschool.db";
    public static int DATABASE_VERSION = 1;
    @Nullable
    public static String VIDEO_CATEGORY_ID="1";
    public static String ID="id";
    public static String VIDEO="data";
    public static String VIDEO_ID="";

    public static boolean switchState=true;
    public static String SOUND="Sound";


    public static final String PREF_LAST_UN_COMPLETE_DAY =
"PREF_LAST_UN_COMPLETE_DAY";

    public static final String FB_BANNER_TYPE_AD = "FB_BANNER_TYPE_AD";
    public static final String FB_RECTANGLE_BANNER_TYPE_AD =
"FB_RECTANGLE_BANNER_TYPE_AD";
    public static final String GOOGLE_BANNER_TYPE_AD = "GOOGLE_BANNER_TYPE_AD";
    public static final String GOOGLE_RECTANGLE_BANNER_TYPE_AD =
"GOOGLE_RECTANGLE_BANNER_TYPE_AD";

    public static final String AD_TYPE_FB_GOOGLE = "AD_TYPE_FB_GOOGLE";
    public static final String FIRST_TIME_OPEN = "FIRST_TIME_OPEN";
    public static final String GOOGLE_BANNER = "GOOGLE_BANNER";
    public static final String GOOGLE_INTERSTITIAL = "GOOGLE_INTERSTITIAL";
    public static final String GOOGLE_REWARDED_VIDEO = "GOOGLE_REWARDED_VIDEO";
    public static final String FB_BANNER = "FB_BANNER";
    public static final String FB_BANNER_RECTANGLE_AD = "FB_BANNER_RECTANGLE_AD";
    public static final String FB_INTERSTITIAL = "FB_INTERSTITIAL";
    public static final String FB_REWARDED_VIDEO = "FB_REWARDED_VIDEO";
    public static final String FIRST_CLICK_COUNT = "FIRST_CLICK_COUNT";
    public static final String EXTRA_REMINDER_ID = "Reminder_ID";
    public static final String WEIGHT_TABLE_DATE_FORMAT = "yyyy-MM-dd";
    public static final String SPLASH_SCREEN_COUNT = "splash_screen_count";
    public static final String START_BTN_COUNT = "start_btn_count";
    public static final String EXIT_BTN_COUNT = "exit_btn_count";
    public static final String STATUS_ENABLE_DISABLE = "STATUS_ENABLE_DISABLE";
    public static final String AD_TYPE = "AD_TYPE";
```

```
    public static final String GOOGLE_ADMOB_APP_ID = "ca-app-pub-
3940256099942544~3347511713";
    public static final String GOOGLE_BANNER_ID = "ca-app-pub-
3940256099942544/6300978111";
    public static final String GOOGLE_INTERSTITIAL_ID = "ca-app-pub-
3940256099942544/1033173712";


    public static final String FB_BANNER_ID =
"IMG_16_9_APP_INSTALL#YOUR_PLACEMENT_ID";
    public static final String FB_INTERSTITIAL_ID =
"IMG_16_9_APP_INSTALL#YOUR_PLACEMENT_ID";


    public static final String AD_FACEBOOK = "facebook";
    public static final String AD_GOOGLE = "google";
    public static final String AD_TYPE_FACEBOOK_GOOGLE = AD_GOOGLE;


    public static final String ENABLE = "Enable";
    public static final String DISABLE = "Disable";
    public static final String ENABLE_DISABLE = ENABLE;
}
```

7.4.3

NON SWIPE ABLE VIEWER Page

```
package com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.customclasses;

import android.content.Context;
import androidx.viewpager.widget.ViewPager;
import android.util.AttributeSet;
import android.view.MotionEvent;
import android.view.animation.DecelerateInterpolator;
import android.widget.Scroller;

import java.lang.reflect.Field;

/**
 * Created by Naynesh Patel on 3/24/2018.
 */

public class NonSwipeAbleViewPager extends ViewPager {

    public NonSwipeAbleViewPager(Context context) {
        super(context);
        setMyScroller();
```

```
    }

    public NonSwipeAbleViewPager(Context context, AttributeSet attrs) {
        super(context, attrs);
        setMyScroller();
    }

    @Override
    public boolean onInterceptTouchEvent(MotionEvent event) {
        // Never allow swiping to switch between pages
        return false;
    }

    @Override
    public boolean onTouchEvent(MotionEvent event) {
        // Never allow swiping to switch between pages
        return false;
    }

    //down one is added for smooth scrolling

    private void setMyScroller() {
        try {
            Class<?> viewpager = ViewPager.class;
            Field scroller = viewpager.getDeclaredField("mScroller");
            scroller.setAccessible(true);
            scroller.set(this, new MyScroller(getContext()));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public class MyScroller extends Scroller {
        public MyScroller(Context context) {
            super(context, new DecelerateInterpolator());
        }

        @Override
        public void startScroll(int startX, int startY, int dx, int dy, int
duration) {
            super.startScroll(startX, startY, dx, dy, 350 /*1 secs*/);
        }
    }
}
```

7.5 DATABASES.

7.5.1

DATABASE HELPER

```
package com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.database

import android.content.Context
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper
import
com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.customclasses.Constant
import com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.model.ModelVideo
import java.io.File
import java.io.FileOutputStream
import java.io.IOException

/**
 * Created by Naynesh Patel on 12-Feb-19.
 */
class DatabaseHelper(internal var context: Context) : SQLiteOpenHelper(context,
Constant.DATABASE_NAME, null, Constant.DATABASE_VERSION) {

    val videoDetails: ArrayList<ModelVideo>
        get() {
            var arrPfVideoDetails = ArrayList<ModelVideo>()
            val db = this.writableDatabase
            val cursor = db.rawQuery("SELECT * FROM kids WHERE id=" +
Constant.VIDEO_CATEGORY_ID, null)
            cursor.moveToFirst()
            while (!cursor.isAfterLast) {
                var modelVideo = ModelVideo()
                var videoDescription =
cursor.getString(cursor.getColumnIndexOrThrow(Constant.VIDEO))
                modelVideo.videoId = videoDescription.split("#")[0]
                modelVideo.videoTitle = videoDescription.split("#")[1]
                modelVideo.videoThumb = "https://i3.ytimg.com/vi/" +
videoDescription.split("#")[0]+"/hqdefault.jpg"
                arrPfVideoDetails.add(modelVideo)
                cursor.moveToNext()
            }
            return arrPfVideoDetails
        }

    override fun onCreate(db: SQLiteDatabase) {
    }

    override fun onUpgrade(db: SQLiteDatabase, oldVersion: Int, newVersion: Int) {
    }

    @Throws(IOException::class)
    fun createDataBase() {
```

```
        val databaseExist = checkDataBase()
        if (databaseExist) {
        } else {
            this.writableDatabase
            copyDataBase()
        }
    }

    private fun checkDataBase(): Boolean {
        val databaseFile = File(Constant.DB_PATH + Constant.DATABASE_NAME)
        return databaseFile.exists()
    }

    @Throws(IOException::class)
    private fun copyDataBase() {
        val myInput = context.assets.open("databases/" + Constant.DATABASE_NAME)
        val outFileName = Constant.DB_PATH + Constant.DATABASE_NAME
        val myOutput = FileOutputStream(outFileName)
        val buffer = ByteArray(1024)
        while (myInput.read(buffer) > 0) {
            myOutput.write(buffer)
        }
        myOutput.flush()
        myOutput.close()
        myInput.close()
    }

}
```

7.6 INTERFACES.

7.6.1

ADS CALLBACK

```
package com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.interfaces;

public interface AdsCallback {

    void adLoadingFailed();

    void adClose();

    void startNextScreen();

    void onLoaded();
}
```

7.6.2

CALLBACK LISTENER

```
package com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.interfaces;

public interface CallbackListener {

    void onSuccess();
    void onCancel();
    void onRetry();

}
```

7.7 MODEL CLASSES.

7.7.1

LEARNING DATA MODEL

```
package com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.model;

/**
 * Created by Naynesh Patel on 06-Feb-19.
 */
public class LearningDataModel {
    public int image;
    public String showTitle;
    public String speakTitle;
    boolean isTrue=false;
//
    public boolean isTrue() {
        return isTrue;
    }

    public void setTrue(boolean aTrue) {
        isTrue = aTrue;
    }

    public LearningDataModel(int i, String str, String str2) {
        this.speakTitle = str;
        this.showTitle = str2;
        this.image = i;
    }

    public int getImage() {
        return this.image;
    }

    public void setImage(int i) {
        this.image = i;
    }

    public String getSpeakTitle() {
        return this.speakTitle;
    }

    public void setSpeakTitle(String str) {
        this.speakTitle = str;
    }

    public String getShowTitle() {
        return this.showTitle;
    }

    public void setShowTitle(String str) {
        this.showTitle = str;
    }
}
```

7.7.2

MODEL VIDEO

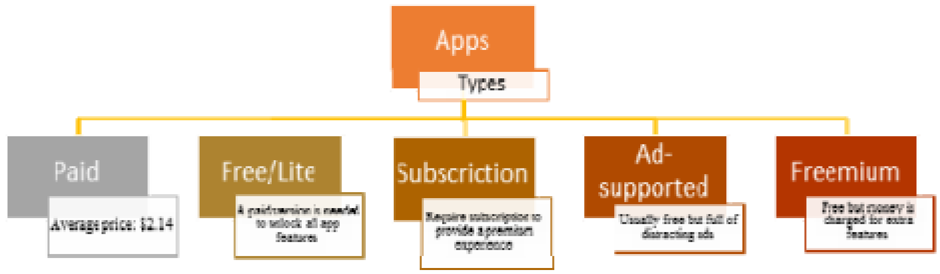
```java
package com.kidslearning.kidsplay.kidsgames.kidseducation.preschool.model;

import java.io.Serializable;

/**
 * Created by Naynesh Patel on 12-Feb-19.
 */
public class ModelVideo implements Serializable {

    String videoId,videoTitle,videoThumb;

    public String getVideoId() {
        return videoId;
    }

    public void setVideoId(String videoId) {
        this.videoId = videoId;
    }

    public String getVideoTitle() {
        return videoTitle;
    }

    public void setVideoTitle(String videoTitle) {
        this.videoTitle = videoTitle;
    }

    public String getVideoThumb() {
        return videoThumb;
    }

    public void setVideoThumb(String videoThumb) {
        this.videoThumb = videoThumb;
    }
}
```

CHAPTER-8

RESULT

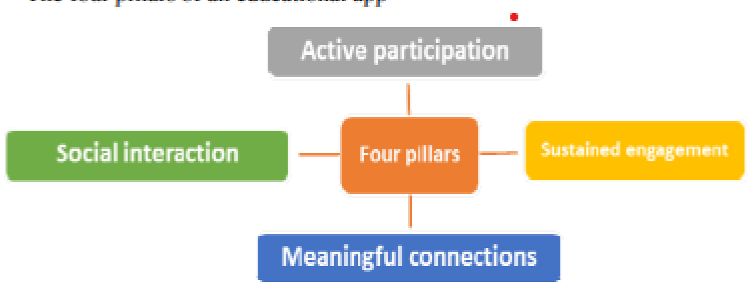
Mobile applications (apps) At the apex of the boom of smart mobile devices is the introduction of applications (apps) for tablets and smartphones (Hirsh-Pasek et al., 2015). Various researchers mention that besides the technological features of the smart mobile devices, these are in favour of young children mostly due to the existence of mobile applications specially Mobile educational applications for children 261 designed for these devices (Zaranis et al., 2013). Research has revealed that young children are very engaged with the apps and love to play with them for various amounts of time depending on their needs and interests and the content and structure of the app (Yelland and Gilbert, 2011). A mobile application is a computer program designed to run on mobile devices such as smartphones and tablet computers (Yusop and Razak, 2013; Wikipedia, 2016; Bouck et al., 2016). A mobile application may also be known as an app, iPhone app or smartphone app (Techopedia, 2016). Goodwin (2013) states that there are more than just 'paid' and 'free' apps in the app stores. There are in fact five different types of apps (see Figure 1). According to their pedagogical dimension, various researchers have tried to categorise apps designed for children in different categories depending on their open- or close-ended character, their level of activity, etc. Goodwin (2012, p.12) distinguishes apps into three different categories: · Constructive or 'productivity' apps, which are characterised by an open-ended design that allows users to create their own content or digital artefact using the app. Constructive apps are designed for creative expression. · Instructive apps have elements of 'drill-and-practice' design whereby the app delivers a predetermined 'task' which elicits a homogenous response from the user. These apps require minimal cognitive investment on behalf of the learner. Most game apps are classified as instructive apps. · Manipulable apps allow for guided discovery and experimentation within a predetermined context or framework. These apps require more cognitive involvement than instructive apps but less than constructive apps.

Figure 1 The five different types of apps



Source: Adapted from Goodwin (2013)

Figure 2 The four pillars of an educational app



Educational applications “KIDDO” can help children to learn about letters, numbers, and names of animals and to perform simple calculations. The educational application “KIDDO” can help children learn to spell letters and words. The educational application “KIDDO” becomes an alternative learning tool for children. More features such as Bahasa as an add-on feature for recognizing letters and numbers. Add math material such as multiplication and division; add more objects or images for features to recognize animal names. The contribution to this research is that this application acts as an alternative medium of learning for children. The learning method used in this application uses a fun playstyle approach.

CHAPTER-9

CONCLUSION AND FUTURE WORK

9.1

CONCLUSION.

The 21st century is an age of mobility and ease (Ellingson, 2016). Young children are being increasingly exposed to media, technology, and screen time (MeTS) at home and in instructional settings (Sharkins et al., 2015). The use of smart mobile devices among primary school children and toddlers is also growing exponentially as children have more access to smartphones through their parents (increasing mobile penetration globally) and 270 S. Papadakis and M. Kalogiannakis as schools embrace technology (Zytnik, 2014). The ease of use, portability, speed and responsiveness of the smart mobile devices and especially tablets were said to make it an ideal learning tool (Watts et al., 2012). Although some educators argue that they have no place in young children's lives the majority enthusiastically embrace those new media for learning (Flewitt et al., 2014). The reason is that 'active, appropriate use of technology and media can support and extend traditional materials in valuable ways … both cognitive and social' (NAEYC, 2012, p.7). The amount of time that children spend with digital media and the surge in educational apps' popularity suggest that at least some apps are being used in an attempt to supplement learning outside of school (Hirsh-Pasek et al., 2015). At the same time, this new technology is not a panacea. As the Office of Educational Technology advise, 'we have to be cognizant of a new digital divide – the disparity between students who use technology to create, design, build, explore, and collaborate and those who simply use technology to consume media passively' (Office of Educational Technology, 2016, p.18). Although tablets themselves are highly versatile and user-friendly for children, they must be coupled with equally well-designed software. According to the NAEYC, in a 1996 position statement point out that: 'Choosing appropriate software is similar to choosing appropriate books for the classroom' (NAEYC, 1996). Smart mobile devices and especially tablets are a technological tool which, when combined with applications that have the appropriate content and design, is able to support the cognitive development of young children. Touchscreen tablets offer many features that enable emergent literacy development as children are able to interact with a range of single and multitouch gestures (Ellingson, 2016). Thus, 'educational' apps are largely unregulated and untested (Hirsh-Pasek et al., 2015). Only when this is achieved, may tablets offer children a mechanism to articulate their creative process from their own perspective (Arnott et al., 2016). Incorporating touchscreen technology in the repertoire of young children's everyday literacy experiences offers new opportunities for ECE (Flewitt et al., 2014). Apps present a significant opportunity for out-of-school, informal learning when designed in educationally appropriate ways (Hirsh-Pasek et al., 2015). However, researchers point out both teachers and parents face difficulties in evaluating self-proclaimed educational apps. In today's technologically saturated society, parents and teachers are both challenged and obligated to ensure that their children's interactions with touchscreen technologies are developmentally appropriate, interactive, and beneficial (Sharkins et al., 2015). Despite the fact that the market is saturated with applications for children, the label 'educational' or 'for children' does not indicate that an app has been validated and tested (Guernsey, 2013, as cited in Crescenzi-Lanna and Grané-Oró, 2016). We live in the first wave of app development, when apps are often just migrations of games and learning scenarios that already exist in non-digital form (Hirsh-Pasek et al., 2015). On the one hand, they face an increasing number of educational apps that become available daily. On the other hand, the majority of the apps have failed to keep their promises to support learning in a purposeful, effective, and enjoyable way. Additionally, in the educational category on Apple's and Google's app store there is often intruding, 'parasitic' content that calls itself as educational in order to exploit the popularity of this category to reap financial benefits. App developers need to consider the design and production of content creation in constructive apps, in order to capitalise on the unique functionality and capabilities of the tablets (Goodwin, 2012). We have to move, in the second wave of educational apps, to apps that foster digital experiences that are Mobile educational applications for children 271 cognitively active, deeply engaging, meaningful, and socially interactive within the context of a learning goal, rather than simply mimicking and extending older media like books, worksheets, television, or even video games (Hirsh-Pasek et al., 2015). We have to keep in mind that, although the selection of good app for children is important, previous studies have shown that their learning performances could be disappointing without proper guidance or learning design (Hwang et al., 2015). For this to happen, it is necessary to create an updated comprehensive guide and/or framework to provide researchers, designers, practitioners, evaluators, educators and parents with the necessary tools in order to progressively refine their practice with children's apps and to enrich traditional design-based research with novel affordances of 21st-century technologies (Kucirkova, 2016). ‘Mobile devices are an integral part of children’s lives and they are here to stay’ (Shuler, 2009b).

9.2 FUTURE WORK

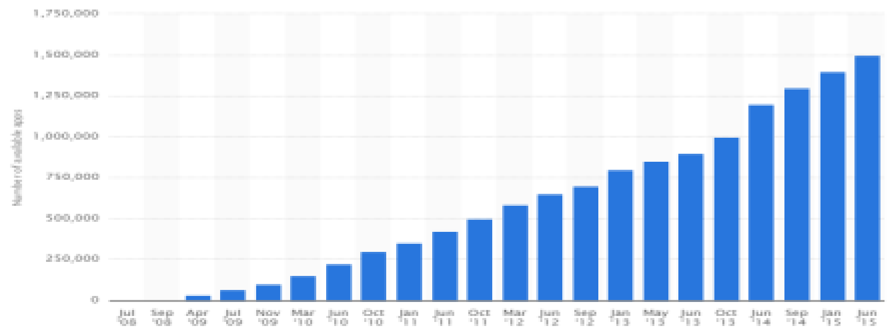
9.2.1

THE APP MARKET PLACE

Mobile applications have turned into a quick enrichment tool for the software industry. Since its entrance into the iPhone business in 2007 (the first smart mobile device with touchscreen), the mobile app industry business has matured and has become part and parcel of the economy itself (Dogtiev, 2015). This rapidly expanding mobile-app industry is worth billions of dollars (Nielsen, 2012). Apple’s digital app store, called App Store, opened for the first time on 10 July 2008, providing free or paid applications for iPhone, and later for iPod touch and iPad devices. In 2011, more than 2.5 billion dollars were ‘paid’ to the software industry (Rideout, 2011) while in 2014 apps generated over 10 billion dollars in revenue for developers (Apple, 2015). As Apple’s CEO stated, at the end of 2014, users had downloaded 75 billion applications and had visited the App Store 300 million times per week (Perez, 2014). Three models, paid, in-app purchase and advertising, power the mobile app business. Total app revenues are projected to grow from 45.37 billion dollars in 2015 to 76.52 billion dollars in 2017. In 2012, in-app purchases accounted for 11.4% of global mobile app revenues and are expected to grow to 48.2% in 2017. In-app purchase revenues will reach 28.9 billion dollars by 2017 (Dogtiev, 2015). According to a report of the Federal Trade Commission (FTC), an independent agency of the US government, entitled ‘Mobile Apps for Kids: Current Privacy Disclosures Are Disappointing’, smart mobile devices users in 2008 could choose from Mobile educational applications for children 263 about 600 apps (Mohapatra and Hasty, 2012). In 2012, there were more than 500,000 apps in the Apple App Store and other 380,000 in Google Play (the digital distribution platform for mobile apps on the Android Operating System). As of June 2015, 1.5 million mobile apps were available in the Apple App Store (Statista, 2016a). Accordingly, the number of available apps in the

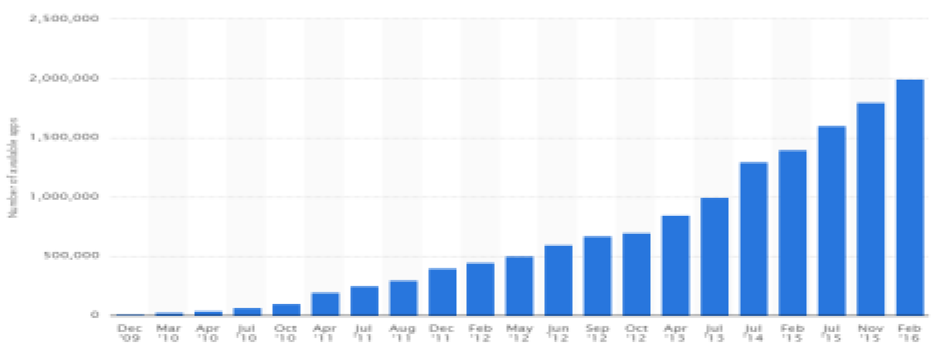
Google Play Store, formerly known as Android Market, surpassed 1 million apps in July 2013 and was most recently placed at 2 million apps in February 2016 (Statista, 2016b). Figures 3 and 4 give information on the number of available apps in the Apple App Store and Google Play Store since July 2008 and December 2009.

Figure 3 Number of available apps in the Apple App Store (see online version for colours)



Source: Statista (2016a)

Figure 4 Number of available apps in the Google Play Store (see online version for colours)



Source: Statista (2016b)

Figure 3 Number of available apps in the Apple App Store (see online version for colours) Source: Statista (2016a) Figure 4 Number of available apps in the Google Play Store (see online version for colours) Source: Statista (2016b) Beginning in infancy, screen technologies dominate the lives of many young children (Campaign for a Commercial-Free Childhood, Alliance for Childhood and Teachers Resisting Unhealthy Children's Entertainment, 2012) and there has been a rush to fill this new space (Donahoo, 2012). The market for apps for children has emerged as apps for preschool and primary school age children are gaining children's attention due to their attractive graphical and interactive elements (Robijt and Van den Broeck, 2013). As children become more familiar with new technologies, target marketing of this particular age group with new products is inevitable. Developers of all sizes in their turn aim at the 264 S. Papadakis and M. Kalogiannakis educational category and the design and development of applications that are targeted at children under 8 years old (Crescenzi-Lanna and Grané-Oró, 2016). Especially the educational app market is a continuously growing industry and becoming one of the few sectors whose numbers improved every year (Cardenal and López, 2015). As children of all ages have embraced smart mobile devices for entertainment and educational purposes they browse and download thousands of apps everyday either paid or free (Mohapatra and Hasty, 2012). In the middle of 2013, the Apple announced a new Apple Kids Store for children under 12 years old. Apps compliant for this category must be made specifically for kids ages 5 and under, ages 6–8, or ages 9–11. As Apple states, the company's aim is to provide 'parents with a place to find age-appropriate apps for their children' (Zytnik, 2014). Figure 5 shows the number of cumulative app downloads from Apple's and Google's App Store from July 2008 to June 2015 (Satish, 2015). Figure 5 The number of cumulative app downloads from Apple's and Google's App Store from July 2008 to June 2015 Source: Satish (2015)

9.2.2

The difficulties that parents and educators are faced while selecting educational apps

With more than 80% of educational mobile applications in the digital stores of iTunes and Google Play appealing to children, parents and educators are faced with a multitude of decisions regarding their choice of the right app. The educational value and appropriateness of the app content can be difficult to determine (Yusop and Razak, 2013). As the App Store features over 80,000 education apps (Apple, 2016a), RodríguezArancón et al. (2013) state that it would be difficult to conclude that all of the apps have been designed according to current theoretical understanding about effective pedagogical practices (Walsh et al., 2010) so as to be beneficial for learners of any age – and especially young children. As already mentioned, in 18 September 2013, the Apple company announced the addition of a 'Kids Category' to the App Store enabling software companies to target specific groups of customers (5 years and under, 6–8 years and 9–11 years). The creation of this new market category has given app developers a direct and relatively unregulated communication channel to a childhood audience. At the same time, Apple provides little information or guidance regarding the appropriate design and content guidelines for the apps targeting these age categories (Chau, 2014). Apps content undergo minimum custody. The criteria for the apps in this ‘Kids category’ do not include pedagogical criteria rather focused on apps’ compliance with certain legal requirements, such as children’s online privacy protection, no behavioural advertising and the prohibition in implementing financial transactions (Zytnik, 2014; Apple, 2016b). Moreover, as Ly (2015) points out, as more apps get uploaded to the digital stores, the competition between the developers becomes more fierce. Although, Vaala et al. (2015) suggest parents and educators try looking for information about apps of interest across 266 S. Papadakis and M. Kalogiannakis app stores and expert review sites or at producers’ websites, this kind or research is not enough or reliable. The reason is that little information on the quality of apps is available, beyond the star ratings published on retailers’ web pages or digital stores (Stoyanov et al., 2015), reviewer comments, or the inclusion of the app on a list (Bouck et al., 2016). Parents and educators do not know how the algorithms work for each of the app stores though some app marketing firms speculate that factors like the number of downloads, ratings, and engagement may be taken into account. Additionally, that competition has led many publishers to look outside of Apple and Google’s ecosystem when the time comes to promote their creations to a wider audience. For example, they use Facebook, with the social network reporting its mobile ad business jumping from nothing only a few years ago, to 59% of its ad revenue (Perez, 2014). Given the preponderance of apps available in the digital stores, teachers and parents need to make critical and informed decisions when selecting apps. Specifically, parents need evidence-based information about the safe and effective use of mobile devices, where to seek quality apps, and suggestions of ways these devices can be used at home to support learning (Goodwin, 2012). Yelland and Gilbert (2011) found in research that the majority of apps are classified as being ‘drill and practice’ and characterised by limited choice and specifically controlled outcomes. As a result, they suggest that educators and parents should take the time to play and become familiar with apps to ensure that they suit their goals for learning with the particular age range of their children. The reason is that ‘claims of the developers are often overinflated and the scope of the app very limited and did not fully use the dynamic features and full potential of the tablet’ (Yelland and Gilbert, 2011, p.19). Henderson and Yeow (2012) point out that the choice of developmentally appropriate applications can be quite a difficult process for those interested. Often parents participate in mobile media activities with their children only if they find the activities enjoyable without worrying about the content of the apps (Heider and Jalongo, 2014). On the other hand, there is a multitude of ‘edutainment’ apps in the iTunes App Store, which are often like ‘digital worksheet’ or ‘skill-and-drill’ apps. Many of the apps in the market have interactive yet repetitive game formats with ‘closed’ content, that is the content could not be changed or extended by the user (Flewitt et al., 2014). Such apps rely on low levels of thinking skills and often do little more than promote rote learning, a memorisation of technique based on repetition (Grose, 2013), such as apps to rote learn colours, numbers, shapes or letters (Goodwin, 2013). For example, drill and practice may foster rote learning of facts, but it is not likely to promote deeper conceptual understanding (Hirsh-Pasek et al., 2015). Additionally, most educational apps that can be found on various websites like Eduapps or eduTecher are very simple apps without a defined goal that can be applied to formal education (Cardenal and López, 2015).

9.2.4

The necessity for the introduction of new standards and assessments tools for the evaluation of the educational value of apps

The popularity of mobile devices and their accompanying apps, as a new educational technology which shapes the way children learn, inevitably creates the need for the introduction of new standards and an evaluation system for the educational value of mobile apps, especially those that target young children (Brown et al., 2010). In a recent research study, Falloon (2013) studied interface design and content of selected apps for primary school children and younger and discussed the imperative need for both Mobile educational applications for children 267 researchers and software developers to work together so as to increase the educational value of applications addressed to young children. In a similar result, Kucirkova et al. (2014) emphasised the need for educational researchers, educators and software companies to find a common framework for consultation, given the growing demand for teachers to integrate mobile technologies and apps into their teaching, in order to assist 21st-century students in meaningful and real-world learning. However, despite the popularity of mobile devices, there are relatively few public sources available for everyone associated with the development and use of educational apps for children. In fact, parents and educators have limited education themselves and/or limited evaluation tools to assess the potential benefits of those apps on children’s learning and development (Emeeyou, 2012; Goodwin and Highfield, 2012). Software developers, parents and educators do not have a comprehensive guide to creating or evaluating mobile educational applications at their disposal. There are, on the internet, mainly scattered and often difficult to find fragmentary suggestions relating the educational appropriateness of an app. Especially for parents it is not enough to focus on an amount of time children are interacting with digital media. Shuler (2009b) advises parents and educators who are not sure which app is appropriate for their children and/or students to use the rule of 'Three C's'. The rule of 'Three C's' which was introduced by the early education researcher Lisa Guernsey for the evaluation of digital media for children is a method for checking the appropriateness of an app and is based on three different criteria: · Content: if the application is age appropriate and if the learning object which it deals with is developmentally appropriate. · Context: if the application interacts with the child and if the child learns through play. · Child: what stimuli a child could acquire from this application. Respectively, according to the non-profit educational organisation 'Tablets for Schools' the surest way for teachers and parents to select developmentally appropriate apps is to browse websites specialised in educational applications, such as the EAS (http://www. educationalappstore.com) (Tablets for Schools, 2014). The EAS is an independent app marketplace in which visitors can find reliable information, as well as applications that have been evaluated by other teachers and are organised by topic, age and curriculum relevance. This organisation advises parents and educators to look for applications, which meet the academic standards, and not choose applications according to their popularity in the digital stores. For example, an app in Google's digital store is considered popular among users when it has many ratings according to Google Play's 5-star rating system or the number of cumulative app downloads is big enough.

9.2.5

Assessment challenges in the usefulness of educational apps for children.

The creation of educational software for preschool and primary school children has long preoccupied many within the education research community, since the introduction of personal computers in classrooms by the early 1980s. Falloon (2013) states that researchers attempting to measure the effectiveness and educational value of mobile applications face the same challenges as those who tried to measure the educational 268 S. Papadakis and M. Kalogiannakis impact of the computer software. The use of design principles applied to the elements of educational software for personal computers can be generally used in the design of educational apps. However, with the new interactive technologies, the criteria for assessing the quality of mobile products targeting children as young as 2 years old must be continually updated to reflect recent trends in handling behaviour and content of self-proclaimed educational apps (Shoukry et al., 2012). McKnight and Fitton (2010) refer to usability issues that emerge in numerous aspects of interactive activities with apps, such as a lack of haptic feedback, which is associated with the use of 'soft buttons' (or 'virtual buttons'), as well as interaction style changes with which children should familiarise themselves. Similarly, Brown et al. (2010) point out that design approaches, though widely used for decades, are based on traditional interaction technologies and methods such as the use of mouse or keyboards as input devices require a complete overhaul due to the use of touchscreens and gesture-based interaction technologies. Only, a small number of developers at both small start-ups and bigger toy/media companies have used research-based approaches with preliminary results of research (Hirsh-Pasek et al., 2015).

CHAPTER-10

REFERENCES

1. Abdul Aziz, N.A.B., Batmaz, F., Stone, R. and Chung, P.W.H. (2013) 'Selection of touch gestures for children's applications', Proceedings of the Science and Information Conference (SAI), 7–9 October, London, pp.721–726.

2. Apple (2015) App Store Rings in 2015 with New Records. Available online at: https://www. apple.com/pr/library/2015/01/08App-Store-Rings-in-2015-with-New-Records.html (accessed on 16 July 2016).

3. Apple (2016a) IPad in Education. Available online at: http://www.apple.com/education/appsbooks-and-more/ (accessed on 12 June 2016). 4. Apple (2016b) App Store Review Guidelines. Available online at: https://developer.apple.com/appstore/review/guidelines/#kids-category (accessed on 18 June 2016).

5. Arnott, L., Grogan, D. and Duncan, P. (2016) 'Lessons from using iPads to understand young children's creativity', Contemporary Issues in Early Childhood, Vol. 17, No. 2, pp.157–173.

6. Avtar, T. (2014) How App Markets are Adding a Lot of Value in Education, Think Tab Blogs. Available online at: http://www.thinktab.com/how-app-markets-are-adding-a-lot-of-value-ineducation/ (accessed on 2 July 2016).

7. Bayles, J. and Knoke-Staggs, G. (2013) 'Tablets and children: intuitive technology and new literacies', Tennessee Libraries, Vol. 63, No. 2, p.1.

8. https://www.geeksforgeeks.org/

9. https://developer.android.com/

10. https://stackoverflow.com/

11. https://www.wikipedia.com/