Contents

**List Of Figures 3**

[1. INTRODUCTION: 5](#_Toc415597471)

[1.1 Introduction to Project 5](#_Toc415597472)

[1.2 Introduction to Windows 8 Platform 6](#_Toc415597473)

[2.1.1 Windows Application Programming Interface 9](#_Toc415597474)

[1.3. Introduction to C#.NET 11](#_Toc415597475)

[1.3.1 What is C#? 11](#_Toc415597476)

[1.3.2 Why C#? 11](#_Toc415597477)

[1.3.3 History of C#? 11](#_Toc415597478)

[1.3.4 .NET Framework Platform Architecture 13](#_Toc415597479)

[1.3.5 Features of C# language 15](#_Toc415597480)

[1.4 eLearning Objective 16](#_Toc415597481)

[2. LITERATURE SURVEY: 17](#_Toc415597482)

[2.1 Education System in Mauritius 17](#_Toc415597483)

[2.2 Alternative 1 18](#_Toc415597484)

[2.3 Alternative 2 18](#_Toc415597485)

[2.3 Alternative 3 19](#_Toc415597486)

[2.4 Existing System 20](#_Toc415597487)

[2.4.1 Microsoft Encarta Kids: 20](#_Toc415597488)

[2.4.2 Proposed System Features Over Existing System: 20](#_Toc415597489)

[2.4.3 eLearning Website: 21](#_Toc415597490)

[3. REQUIREMENT ENGINEERING: 22](#_Toc415597491)

[3.1 Hardware Requirement 22](#_Toc415597492)

[3.2 Software Requirement 22](#_Toc415597493)

[4. SYSTEM ANALYSIS AND DESIGN: 23](#_Toc415597494)

[4.2 Data Flow Diagrams 23](#_Toc415597495)

[4.2.1 DFD Level 0: 23](#_Toc415597496)

[4.2.2 DFD Level 1: 24](#_Toc415597497)

[4.2.3 DFD Level 2 Playgroup: 24](#_Toc415597498)

[4.2.4 DFD Level 2 LKG: 25](#_Toc415597499)

[4.2.4 DFD Level 2 UKG: 25](#_Toc415597500)

[4.3 Use Case Diagram for Only Playgroup 26](#_Toc415597501)

[5. GUI SCREENSHOTS: 27](#_Toc415597502)

[6. TESTING AND RESULTS: 35](#_Toc415597503)

[6.2 Types of Testing 35](#_Toc415597504)

[6.2.1 Unit Testing: 35](#_Toc415597505)

[6.2.2 Integration Testing: 35](#_Toc415597506)

[7. SYSTEM DEVELOPMENT: 36](#_Toc415597507)

[7.1 Analytical Model 36](#_Toc415597508)

[7.2 Introduction of SDLC Model 37](#_Toc415597509)

[Stage 1: Planning and Requirement Analysis: 37](#_Toc415597510)

[Stage 2: Defining Requirements: 38](#_Toc415597511)

[Stage 3: Designing the product architecture: 38](#_Toc415597512)

[Stage 4: Building or Developing the Product: 38](#_Toc415597513)

[Stage 5: Testing the Product: 39](#_Toc415597514)

[Stage 6: Deployment in the Market and Maintenance: 39](#_Toc415597515)

[7.3 Waterfall Model 40](#_Toc415597516)

[7.3.1 Introduction: 40](#_Toc415597517)

[7.3.2 Applications: 41](#_Toc415597518)

[8. CONCLUSION AND FUTURE SCOPE: 42](#_Toc415597519)

[8.1 Future Scope: 42](#_Toc415597520)

[9. REFERENCES: 43](#_Toc415597521)

List of Figures:

1. ***Figure 1-1.***Developer Preview 6
2. ***Figure 1-2.*** Consumer Preview 7
3. ***Figure 1-3.*** Release Preview 7
4. ***Figure 1-4.*** Windows 8 Platform overview 8
5. ***Figure 1-5.***C# Programming Structure 13
6. ***Figure 2-1.*** Home page of E-learning for Kids 20
7. ***Figure 4.1 to 4.4*** DFD Diagrams……………………………………………....23 to 25
8. ***Figure 4.5*** Use Case Diagrams…………………………………………………...26
9. ***Figure 5.1 to 5.16*** GUI Screenshot…………………………………………….27 to 34
10. ***Figure 7-1.***Software Development Life Cycle 37
11. ***Figure 7-2.***Waterfall Model 40

# 1. INTRODUCTION:

## 1.1 Introduction to Project

As our project name suggests we have designed such a software which is mainly developed for kids with their full interest as they will show more interest in such a learning than our normal teaching. We have developed software which can fix at anywhere as per the requirements of schools, teachers and also be used at private classes, at homes too (parents can teach their kids as per their convenience at home).

It provide user friendly and enjoyable environment to kids while learning and a SPEAKING TUTOR (Video Interface).The project mainly contains the PLAY GROUP, LKG and UKG syllabus contents.

The PLAY GROUP includes the Alphabets and three associated word of each alphabets with pictures and pronunciations which helps to recognize it easily. Numbers from 1 to 10 with full of pictures and interesting things. We have also included Video Poems and General Knowledge related to PLAY GROUP. The LKG Syllabus includes Alphabets and their Writing Patterns with Capital and Small letters. Numbers from 1 to 100, Counting, Comparison, Poems and General Knowledge related to LKG. All are containing the Tasks associated with it to check their knowledge.

The UKG includes introduction to Cursive Writing, in Mathematics we are giving the basic information to Addition and Subtraction. Interesting Poems and Other Things.

## 1.2 Introduction to Windows 8 Platform

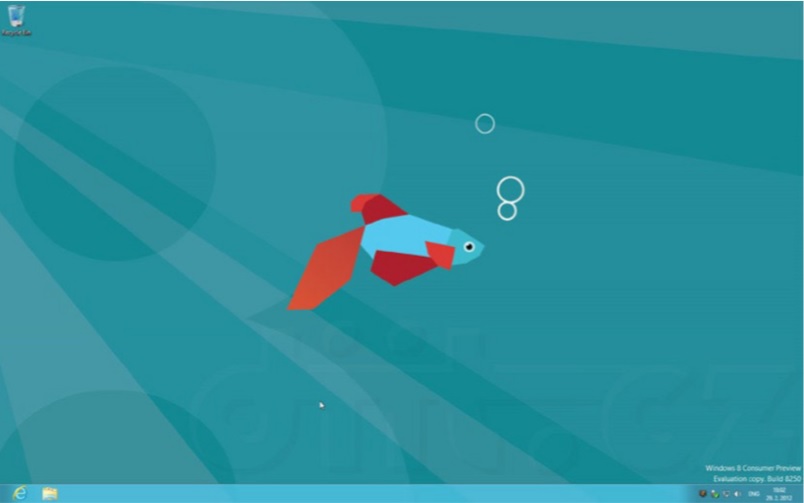
Windows 8 is an exciting release. It tries to reimagine the way we use computers; it’s a fresh wind through an old, established environment. Before we dive into the technical stuff, we’ll have a small history lesson about the development process of Windows 8.

History Windows 8 was first announced at the Consumer Electronics Show (CES) in 2011. Back then, it was just the next Windows iteration. A very early version found its way onto the Internet pretty fast, and inside was evidence of native Universal Serial Bus (USB) 3.0 support, United Extensible Firmware Interface (UEFI), and even some slight hints at an application (app) store. The leaked version still had the classic Start button and Start menu like in Windows 7. The biggest news, however, was not really the announcement of the new operating system but the fact that Windows was getting an Acom RISC Machine (ARM) version next to the classic x86/x64 versions. This immediately fueled rumors of Windows 8 focusing on touch and taking on capabilities of the iPad and Android tablets. A few months later Microsoft showed off the new interface for the first time at the Taipei Computex 2011 show. This was followed closely by the Build Windows 8 blog, where the Windows 8 product team would reflect on certain decisions and features involving Windows 8. On September 13, 2011, the keynote presentation of the Build conference went into detail about the Windows 8 app store and the development process. Microsoft Belgium asked all their technical communities to set up a live stream for their members. I remember sitting there and getting really excited about getting started developing for this brand new platform with a potential huge reach of customers. In that same keynote there was an announcement of the Windows 8 Developer Preview, marking the start of one of the biggest beta programs of any Windows release so far. The Developer Preview came with Visual Studio 2012 beta (known as Visual Studio 11 back then) and a Blend 5 beta, giving developers the chance to dive into the platform early on before release. It also included some installed example apps that had been developed by interns at Microsoft. The Developer Preview did not yet have an app store, but its sole purpose was to get developers familiar with the new Windows Software Development Kit. The operating system and Visual Studio bits were still in a very early stage, resulting in many crashes and frustration among those trying to learn the new development environment. The Developer Preview, shown in Figure 1-1, still had a Start button, it was square this time but it was still there. It would take the user to the Start screen, so the classic menu was already gone. A registry hack appeared on the Web faster than you could say metro. That hack brought back the classic Windows 7 shell.



***Figure 1-1.*** *Developer Preview*

In February 2012 Microsoft announced the Windows 8 Consumer Preview, as shown in Figure 1-2. This version was aimed toward normal users to get them familiar with the new interface. In this version the Start button was gone, but other than that it still strongly resembled Windows 7 with all the glass effects. Developers were able to get a new Visual Studio 11 beta version and could actually get work done on this. There was an increase in stability in both Windows and Visual Studio. Microsoft reported that the Consumer Preview was downloaded over 1 million times in 24 hours, making it one of the most popular beta products ever released by the software company.



***Figure 1-2.*** *Consumer Preview*

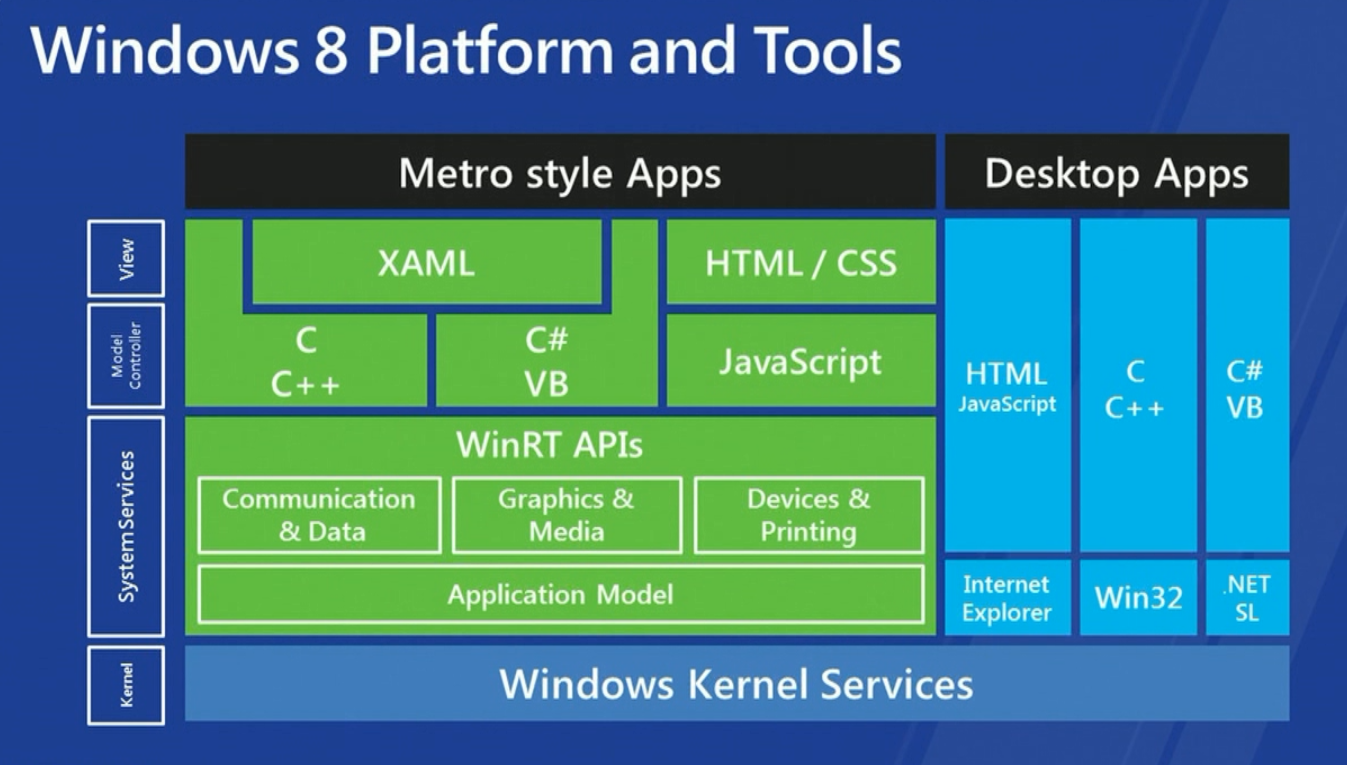
June 2012 marked the release of the last beta version, called the Windows 8 Release Preview. The Release Preview, as shown in Figure 1-3, included a multitude of apps and a working app store. Developers had to go through a long process with both their local Microsoft evangelists and the Microsoft test team before they could get their apps in the store.

***Figure 1-3.*** *Release Preview*

The final version of Windows 8 was released to Microsoft Developer Network (MSDN) and Technet subscribers in August 2012 with general availability in October. After this quick history lesson, we’ll have a look at the internals of Windows 8 and the available application programming interfaces (APIs). The information in this chapter is accurate at the time of writing but, as with any software, might change over time. The best place to check if this information is still up to date is the MSDN documentation.

### 2.1.1 Windows Application Programming Interface

At Build, the chart presented in Figure 1-4 was shown to the public, laying out the differences between desktop applications and Windows Store apps.



***Figure 1-4.*** *Windows 8 Platform overview*

This image came with the announcement of different technologies for building Windows Store apps. We can use C, C++, C#, and Visual Basic to develop apps with Extensible Application Markup Language as the design language or use JavaScript with Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS). The fact that JavaScript and HTML can be used to develop apps meant that, in theory, web developers could hop on the bandwagon and start developing Windows Store apps. In reality they do have to learn some platform-specific stuff, but it should be quite easy to get started. The image also shows that the Windows Store apps (still called Metro back then) and Desktop apps are both built on top of the Windows kernel, but they live in separate environments and have their own specific API set. A good example to prove this is the use of the webcam. Ever tried accessing a webcam from .NET? It requires interop calls to the avicap32 DLL, there are a bunch of wrappers, but it’s still quite a pain to get it working. In a Windows Store app, you need two lines of code to show a webcam stream, as shown in Listing 1-1.

Listing 1-1. Accessing a Webcam in a Windows Store App

var dialog = new CameraCaptureUI(); await dialog.CaptureFileAsync(CameraCaptureUIMode.PhotoOrVideo);

Quite the difference! In the meantime, there has been some success in using the Windows Store API set in normal .NET applications using reflection, but to get those applications to work you need all users to have Windows 8.

## 1.3. Introduction to C#.NET

### 1.3.1 What is C#?

C# (pronounced as ‘C Sharp’) is the language that has been designed from ground up with Internet in mind. It is modern language that combines the power of C++ with productivity of VB and elegance of Java.

### 1.3.2 Why C#?

C# is a modern and innovative programming language that carefully incorporates features found in the most common industry and research languages. In keeping with the design philosophy of C#, Microsoft has introduced several potential new features to the C# language that increase developer productivity with language constructs.

### 1.3.3 History of C#?

Since its introduction in February 2001, a number of developers have begun building software using the C# programming language. Even within Microsoft, C# has been used to build several applications, including the .NET Framework and the Tablet PC SDK. As such, C# has proven itself as a language suitable for the construction of high-quality commercial software.

Developed by **Anders Hejlsberg.** Thisis a language designed for the .Net platform. C# promotes one-stop coding, the grouping of Classes, interfaces and implementations together in one file so that developer can edit code more easily.

C# syntax is highly expressive, yet it is also simple and easy to learn. The curly-brace syntax of C# will be instantly recognizable to anyone familiar with C, C++ or Java. Developers who know any of these languages are typically able to begin to work productively in C# within a very short time. C# syntax simplifies many of the complexities of C++ and provides powerful features such as nullable value types, enumerations, delegates, lambda expressions and direct memory access, which are not found in Java. C# supports generic methods and types, which provide increased type safety and performance, and iterators, which enable implementers of collection classes to define custom iteration behaviors that are simple to use by client code. Language-Integrated Query (LINQ) expressions make the strongly-typed query a first-class language construct.

As an object-oriented language, C# supports the concepts of encapsulation, inheritance, and polymorphism. All variables and methods, including the Main method, the application's entry point, are encapsulated within class definitions. A class may inherit directly from one parent class, but it may implement any number of interfaces. Methods that override virtual methods in a parent class require the override keyword as a way to avoid accidental redefinition. In C#, a struct is like a lightweight class; it is a stack-allocated type that can implement interfaces but does not support inheritance.

In addition to these basic object-oriented principles, C# makes it easy to develop software components through several innovative language constructs, including the following:

* Encapsulated method signatures called delegates, which enable type-safe event notifications.
* Properties, which serve as accessors for private member variables.
* Attributes, which provide declarative metadata about types at run time.
* Inline XML documentation comments.
* Language-Integrated Query (LINQ) which provides built-in query capabilities across a variety of data sources.

If you have to interact with other Windows software such as COM objects or native Win32 DLLs, you can do this in C# through a process called "Interop." Interop enables C# programs to do almost anything that a native C++ application can do. C# even supports pointers and the concept of "unsafe" code for those cases in which direct memory access is absolutely critical.

The C# build process is simple compared to C and C++ and more flexible than in Java. There are no separate header files, and no requirement that methods and types be declared in a particular order. A C# source file may define any number of classes, structs, interfaces, and events.

The following are additional C# resources:

* For a good general introduction to the language, see Chapter 1 of the C# Language Specification.
* For detailed information about specific aspects of the C# language, see the C# Reference.
* For more information about LINQ, see LINQ (Language-Integrated Query).
* To find the latest articles and resources from the Visual C# team, see the [Visual C# Developer Center](http://go.microsoft.com/fwlink/?LinkId=47811).

### 1.3.4 .NET Framework Platform Architecture

C# programs run on the .NET Framework, an integral component of Windows that includes a virtual execution system called the common language runtime (CLR) and a unified set of class libraries. The CLR is the commercial implementation by Microsoft of the common language infrastructure (CLI), an international standard that is the basis for creating execution and development environments in which languages and libraries work together seamlessly.

Source code written in C# is compiled into an intermediate language (IL) that conforms to the CLI specification. The IL code and resources, such as bitmaps and strings, are stored on disk in an executable file called an assembly, typically with an extension of .exe or .dll. An assembly contains a manifest that provides information about the assembly's types, version, culture, and security requirements.

When the C# program is executed, the assembly is loaded into the CLR, which might take various actions based on the information in the manifest. Then, if the security requirements are met, the CLR performs just in time (JIT) compilation to convert the IL code to native machine instructions. The CLR also provides other services related to automatic garbage collection, exception handling, and resource management. Code that is executed by the CLR is sometimes referred to as "managed code," in contrast to "unmanaged code" which is compiled into native machine language that targets a specific system. The following diagram illustrates the compile-time and run-time relationships of C# source code files, the .NET Framework class libraries, assemblies, and the CLR.



***Figure 1-5.*** *C# Programming Structure.*

Language interoperability is a key feature of the .NET Framework. Because the IL code produced by the C# compiler conforms to the Common Type Specification (CTS), IL code generated from C# can interact with code that was generated from the .NET versions of Visual Basic, Visual C++, or any of more than 20 other CTS-compliant languages. A single assembly may contain multiple modules written in different .NET languages, and the types can reference each other just as if they were written in the same language.

In addition to the run time services, the .NET Framework also includes an extensive library of over 4000 classes organized into namespaces that provide a wide variety of useful functionality for everything from file input and output to string manipulation to XML parsing, to Windows Forms controls. The typical C# application uses the .NET Framework class library extensively to handle common "plumbing" chores.

For more information about the .NET Framework, see Overview of the Microsoft .NET Framework.

### 1.3.5 Features of C# language

* A unified type system and simplifying the way that value and reference types are used by the language
* A component-based design established through features such as XML comments, attributes, properties, events and delegates.
* Practical developer headroom established through the unique capabilities of the C# language, including safe pointer manipulation, overflow checking, and more.
* Realistic language constructs, such as the foreach and using statements, which improve developer productivity.

.NET FRAMEWORK

## 1.4 eLearning Objective

Our e-learning tool will be customized for Mauritian kids. The online e-learning system is a web based application mainly for academic purposes. The website is meant for pupils and students aged from 8 to 16 years old. The system will deliver academic resources especially which are helpful to students in their studies. Similarly it also aims in improving the quality of education as well as encourages self-learning among student and promotes ICT.

In the development of the system, the Rational Unified Process (Kruchten, 2003) has been used as methodology. The system has been developed using several up-to-date technologies. ASP.NET 2.0 (http://www.asp.net) platform and Visual Basic.Net have been used. JavaScript has been the main client-side scripting language used for validations and the display of user friendly messages to the user. Cascading Style Sheets technology has been used massively which renders the constant display of the design in all the forms. CSS reduces the size of the pages which makes it faster to load than the use of images. XML Technologies have also been used so as to dynami- cally configure the server from a client host by the administrator. Flash has been used for anima- tions and to provide interaction with the site.

# 2. LITERATURE SURVEY:

## 2.1 Education System in Mauritius

Education System in Mauritius since Mauritius was a British colony before it became independent in 1968, the education system in Mauritius has largely been modeled on the British system. Post-independence, education has always been a priority of the government. To be able to meet the challenges awaiting the country, considerable investment of resources, both human and material, has been put into the education sector. As a result, an impressive progress has been achieved in terms of free, universal, compulsory primary education. Free textbooks are provided to all pupils in primary schools. Free secondary education has also been initiated in Mauritius since 1976 and there is a variety of undergraduate and postgraduate courses offered by the tertiary institutions in Mauritius. Since 1988, the University of Mauritius is providing free full time undergraduate courses. As we can note, the Government of Mauritius has made a substantive effort to provide sufficient funding for education. The government has also been subsidizing significantly private confessional schools, which are administered by the Catholic Church.

In general, children are enrolled in primary schools at the age of six. They are admitted in Stan- dard I and from there, they move automatically to higher standards until they reach Standard IV, where they are categorized via a streaming process. Preparation for the national examination, which is the Certificate of Primary Education, starts as from Standard V. The pupils must study five compulsory subjects, namely English, French, Mathematics, Science, and History & Geogra- phy. These pupils are also given the opportunity to study Asian languages.

The Certificate of Primary Education results determine the admission of the children into secon- dary schools. There are two important examinations at the secondary level. After successfully completing five years of study, the students sit for the School Certificate examination. After passing the School Certificate examination, the students sit for the Higher School Certificate ex- amination after studying for two additional years. At colleges, a variety of subjects are taught. From Form 1 up to Form 3, all subjects are compulsory. The science subjects, Economics, Ac- counting, among others, are optional as from Form 4.

## 2.2 Alternative 1

The first priority is to have an education relevant to the needs of the country bearing in mind that new Information Technology based skills are rapidly replacing the traditional skills all around the world. This means the primary school curriculum must be reviewed so as to give our children a broad-based education that will promote their mental, physical, aesthetic, spiritual and cultural development and at the same time prepare them for the opportunities, responsibilities and experiences of life.

The new curriculum should meet the development requirements of the country. It must provide the country with an educated and a skilled workforce. It will make our children responsible citizens and equip them with the necessary qualities to face the new challenges of the economy and the trends of the 21st century society.

## 2.3 Alternative 2

Fundamentally, one of the main objectives of education is to train individuals based on the needs of the society. And the needs of any society are highly influenced by the needs of other societies throughout the world. As a result, the Mauritian educational system must provide education of international standard to its citizens. Nowadays, to be equipped with the ability to access information, to compose, to evaluate, to use information and to communicate with others have become an absolute requirement for trained people. One way to make this necessity easier and more efficient can be located in the integration of the Mauritian educational system into ICT. This integration has two basic elements complementing each other which reflect on the students’ needs and this process necessitates both teachers and school principals.

Teachers are required to bring behavioral changes in the students. The teachers are expected to integrate their lessons with ICT in order to train the individuals of an information society. Thus, the teachers should have a sound knowledge of ICT. The teacher should be trained if required.

## 2.3 Alternative 3

Since the Internet is a craze among youngsters in Mauritius then we should make use of it to pro- vide a way of learning to our students, which can encourage self-learning as well as developing their skills in ICT. It would be interesting to have an online system for kid’s age from 8 to 16 years where they can interact with each other and where they are provided with a wide range of resources for their studies.

Analysis of the Different Alternatives Alternative 1 seems to be quite an appropriate solution, but it will be a lengthy process to fully implement it. For the second alternative, we can say that providing training is good solution to the problem but it is not cost effective besides the implementation being a long process. The ma- jor problem with the second alternative is that we will still be sticking to our traditional way of learning. Despite living in a modern life, students will not be able to develop their ICT skills. The third alternative appears to be the best option. The development of an e-learning system for kids will be appropriate. Anyway, if Mauritius aims at becoming a competitive knowledge-based economy, there is an urgent need to foster and actively support the widespread deployment and adoption of e-learning throughout the country, in education, in the home and in industry, espe- cially among youngsters.

## 2.4 Existing System

### 2.4.1 Microsoft Encarta Kids:

* + Provides encyclopedia articles, photos and illustrations, videos and animations, audio, word dictionary, maps, visual tours and web links.
  + Bright colors, large type sizes.
  + Loads of multimedia.
  + Easy to navigate, and uses language suitable for young readers.
  + Anticipates children’s misspellings in helping to guide them to the desired content.
  + Games are fun and informative

### 2.4.2 Proposed System Features Over Existing System:

* + We will provide better user interface and group wise divide the content of learning. And we will not add the maps, encyclopedia articles and dictionary in project.
  + We provide syllabus as per Indian culture.
  + We will add poem and stories and their videos in it as per different classes.
  + We will also add the cursive writing tutorials in it.
  + We provide only in single language i.e. English to the user.
  + We will add all possible information in General Knowledge (Videos, images and audios).

### 2.4.3 eLearning Website:

An Existing E-Learning Tool for Kids E-Learning for Kids is an existing non-profit foundation dedicated to fun and free learning on the Internet for children ages 5 – 12 with courses in math, science, reading, health and computers. More than 25 e-learning-related companies, several associations and 70 individuals are sponsors and supporters of the foundation. The website address is http://www.e-learningforkids.org. The homepage of the website is shown in Figure 1.



***Figure 2-1.*** *Home page of E-learning for Kids (http://www.e-learningforkids.org)*

# 3. REQUIREMENT ENGINEERING:

## 3.1 Hardware Requirement

1) Minimum 2 GB RAM

2) 1 GB Disk Free Space Required

3) Processor 1.5 GHz or Greater

## 3.2 Software Requirement

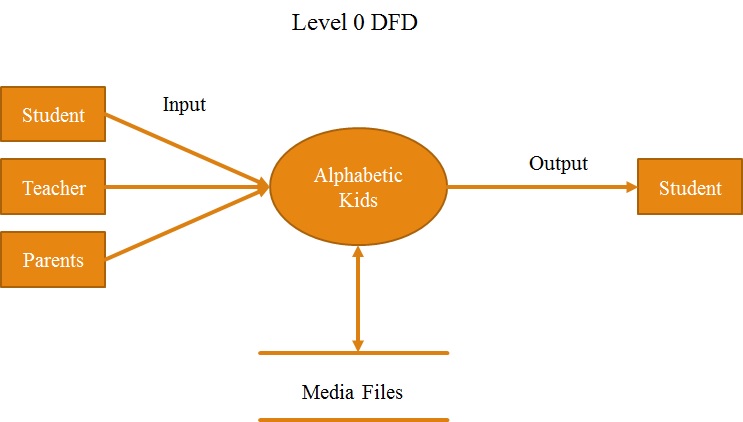
1) Windows 8.1 or Greater (Recommended)

2) MICROSOFT .NET 3.0 or Greater

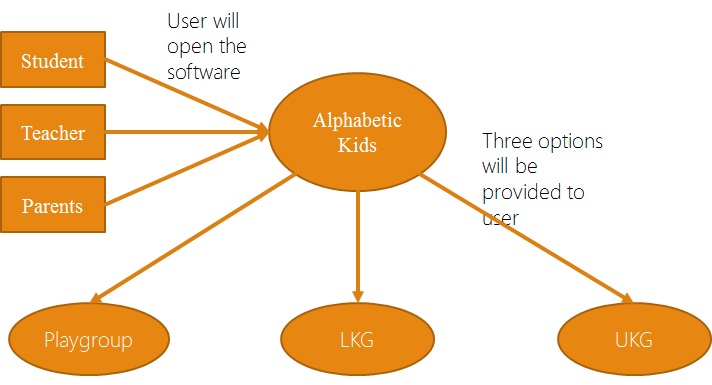
# 4. SYSTEM ANALYSIS AND DESIGN:

## 4.2 Data Flow Diagrams

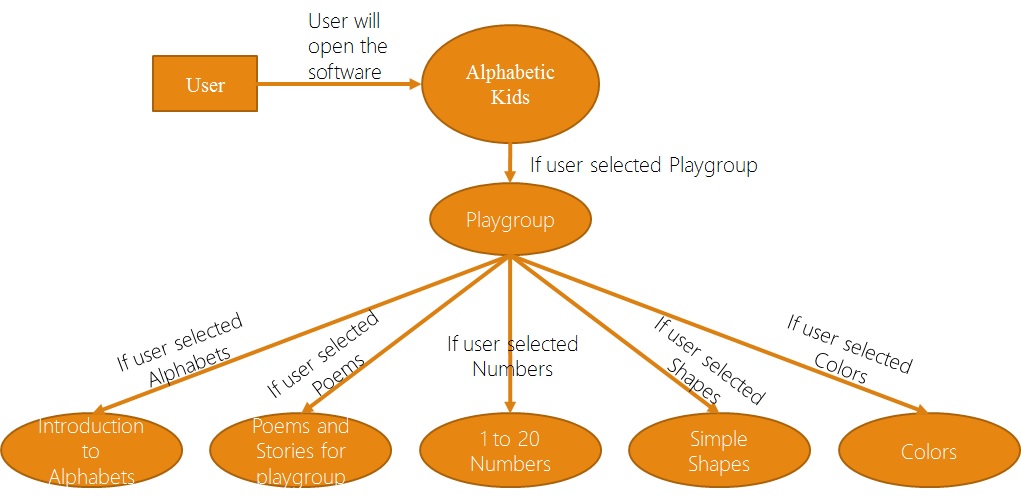
### 4.2.1 DFD Level 0:



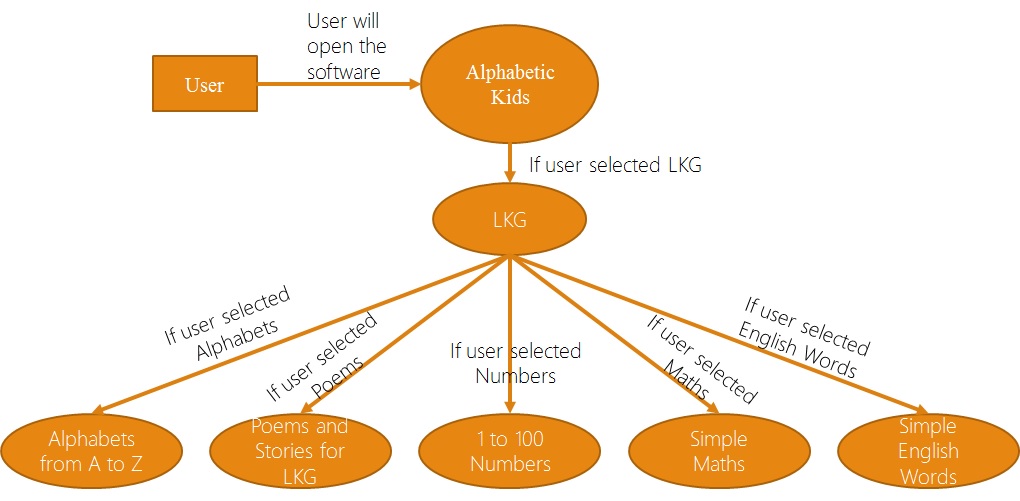
### 4.2.2 DFD Level 1:



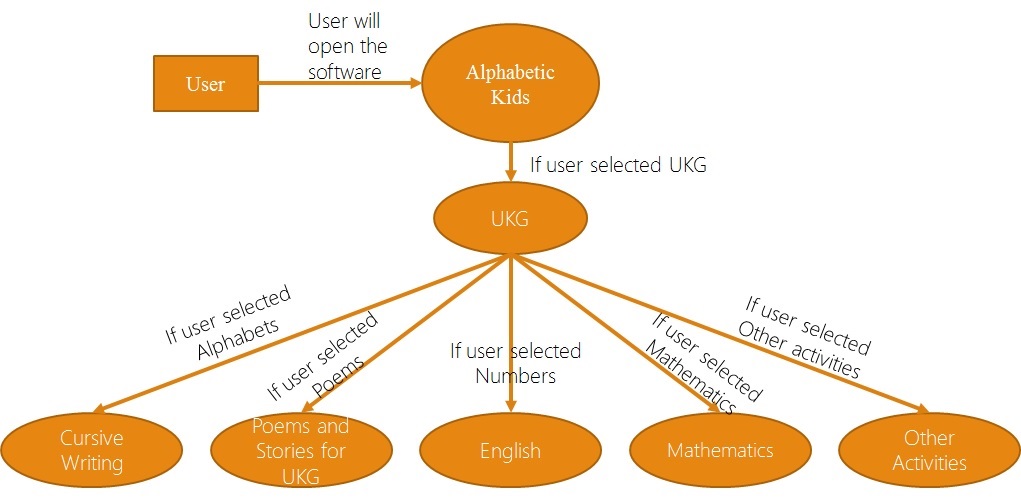
### 4.2.3 DFD Level 2 Playgroup:



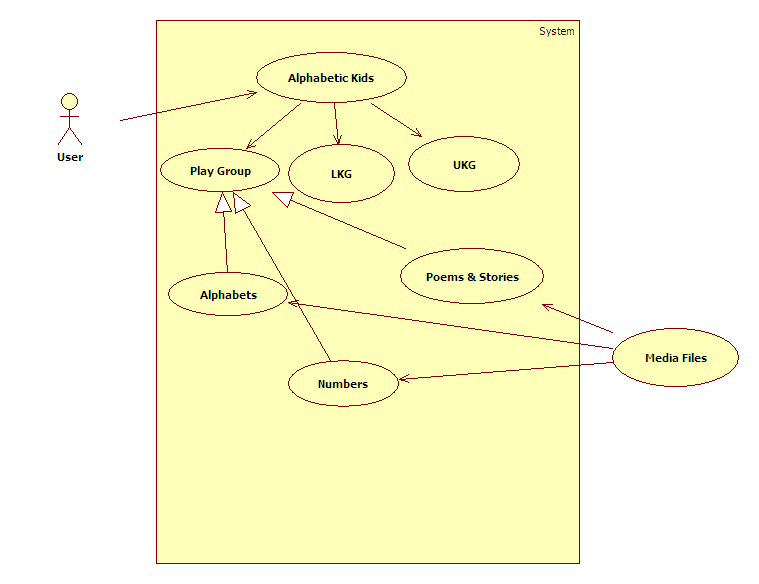
### 4.2.4 DFD Level 2 LKG:



### 4.2.4 DFD Level 2 UKG:

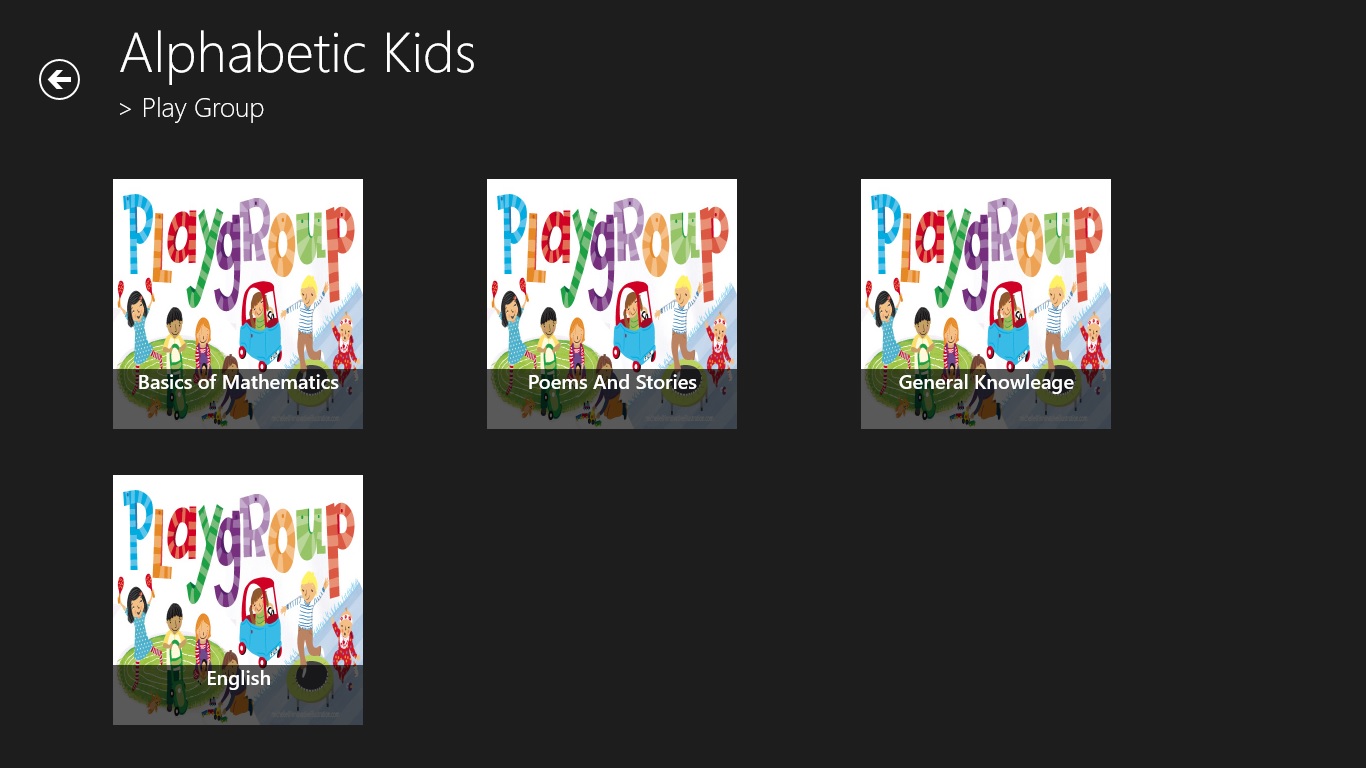


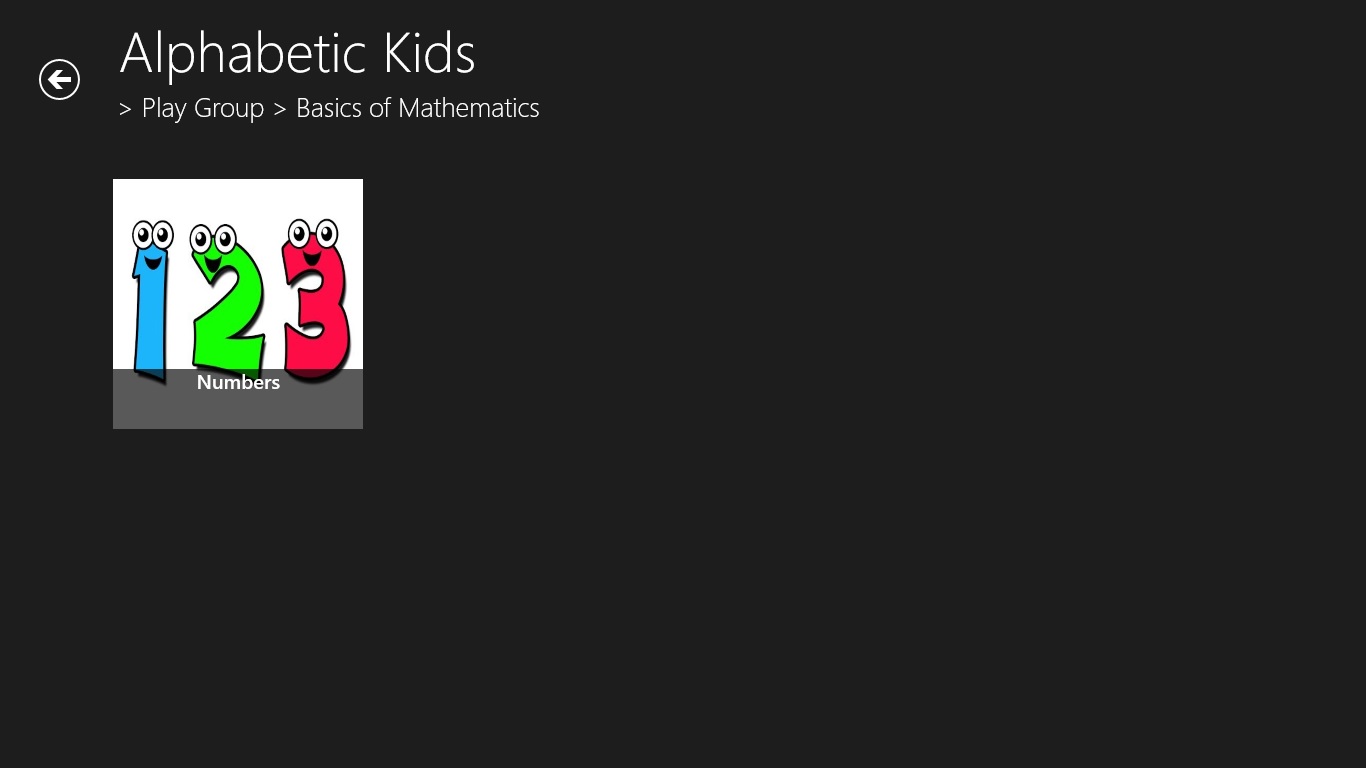
## 4.3 Use Case Diagram for Only Playgroup

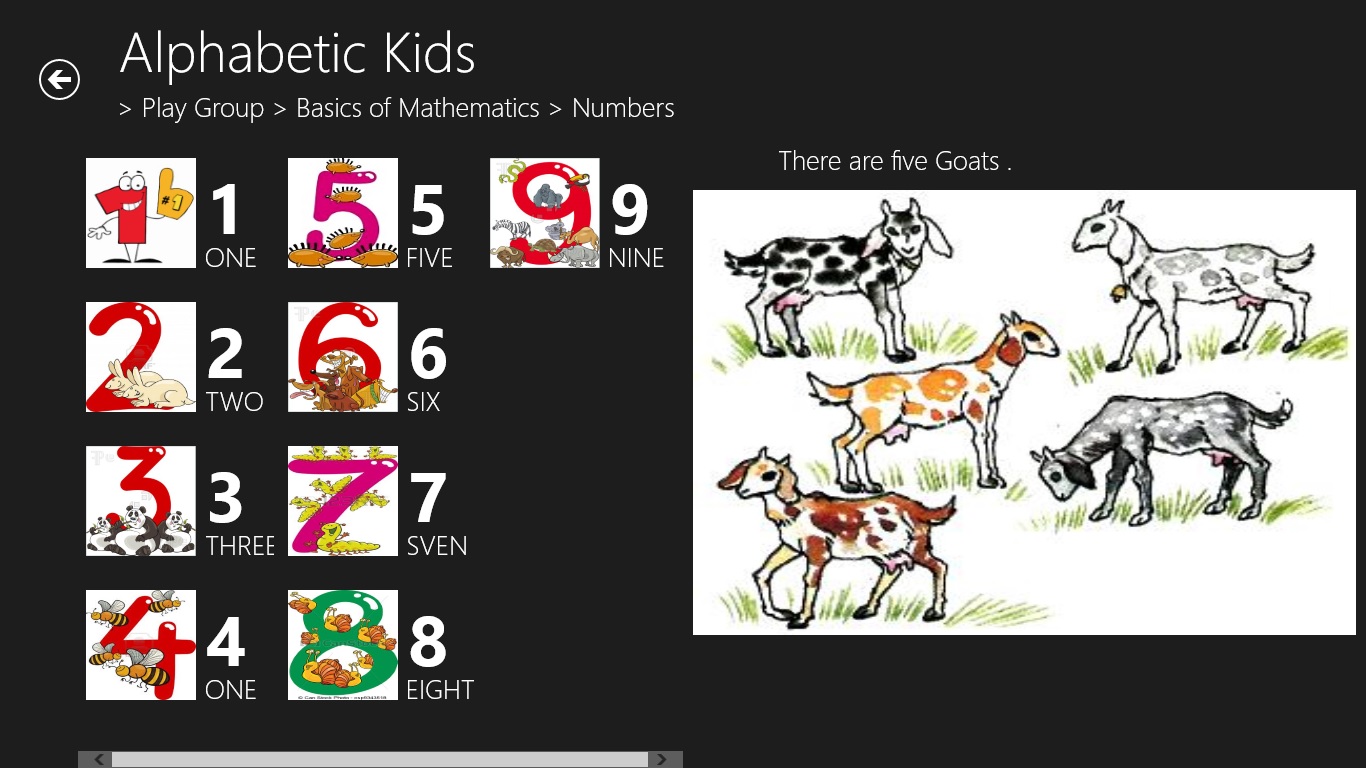


# 5. GUI SCREENSHOTS:



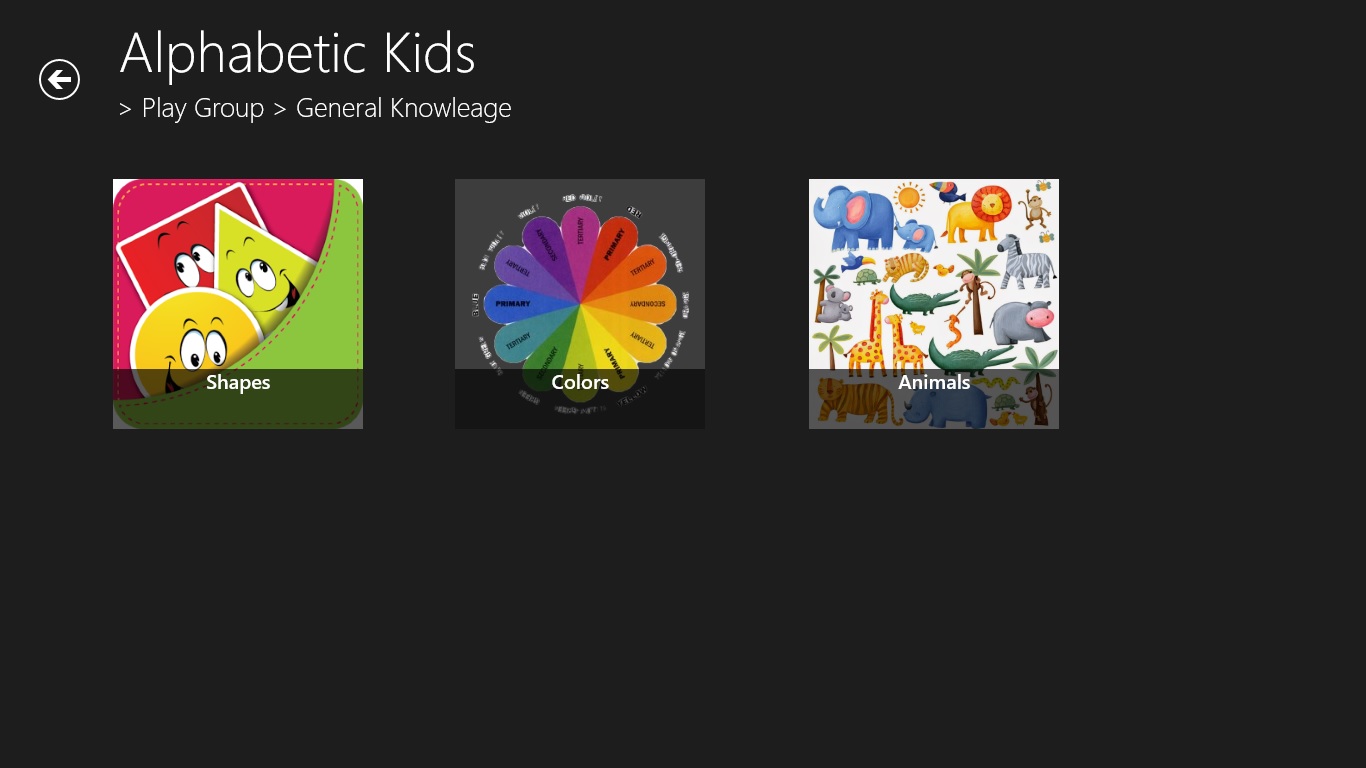






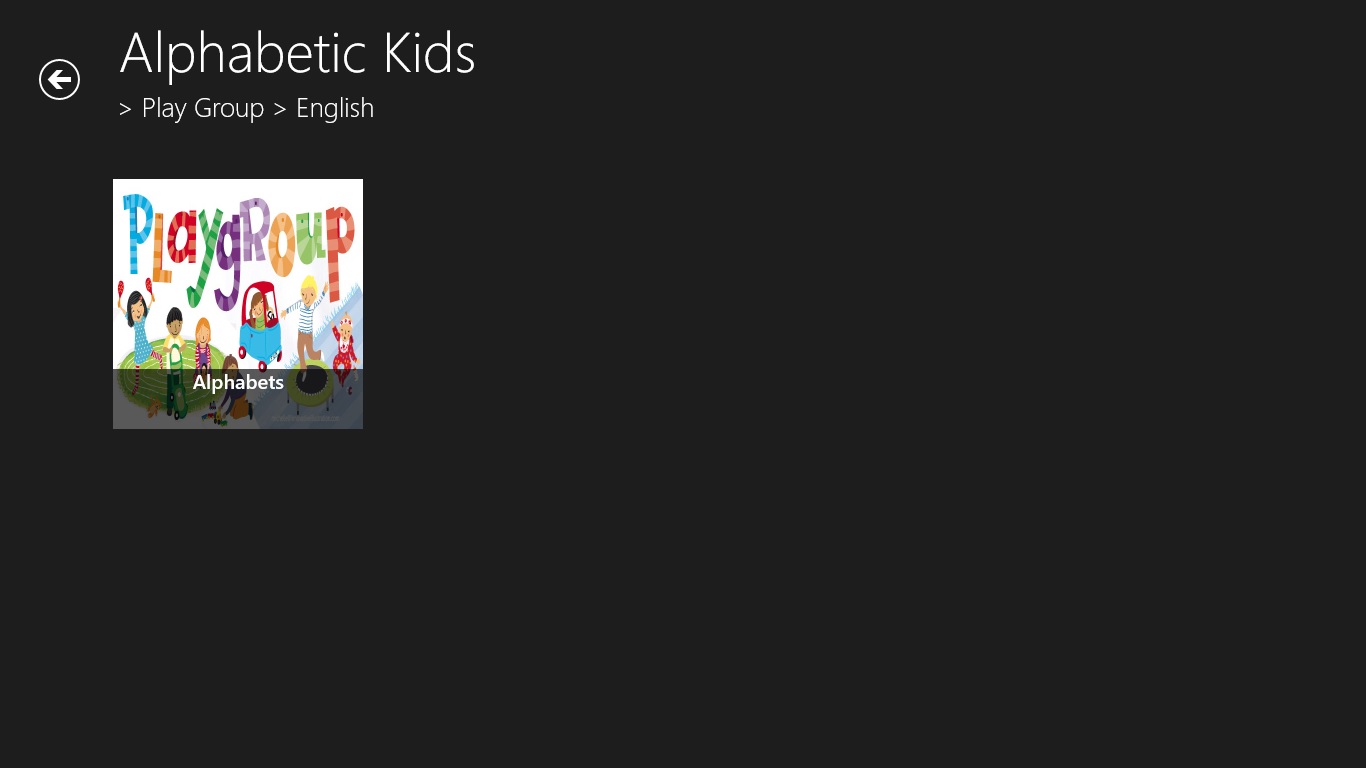


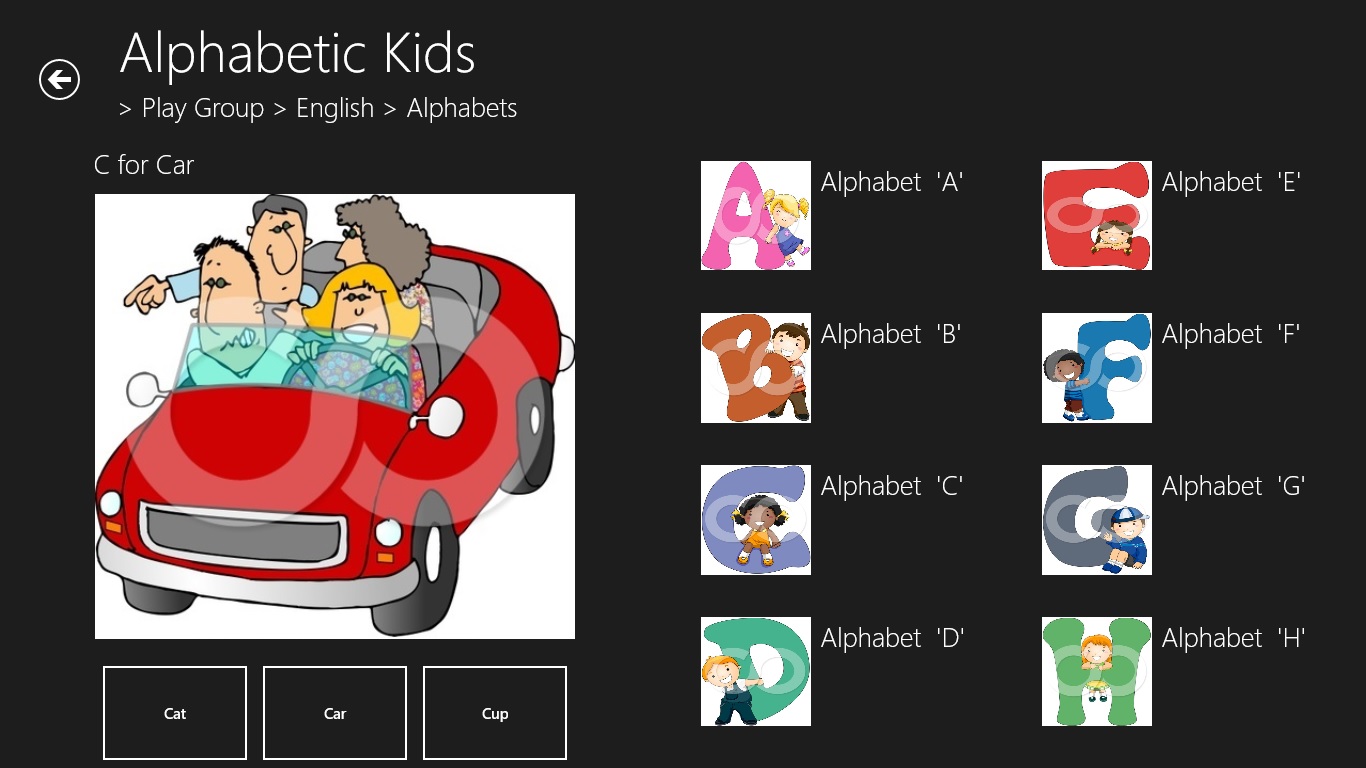


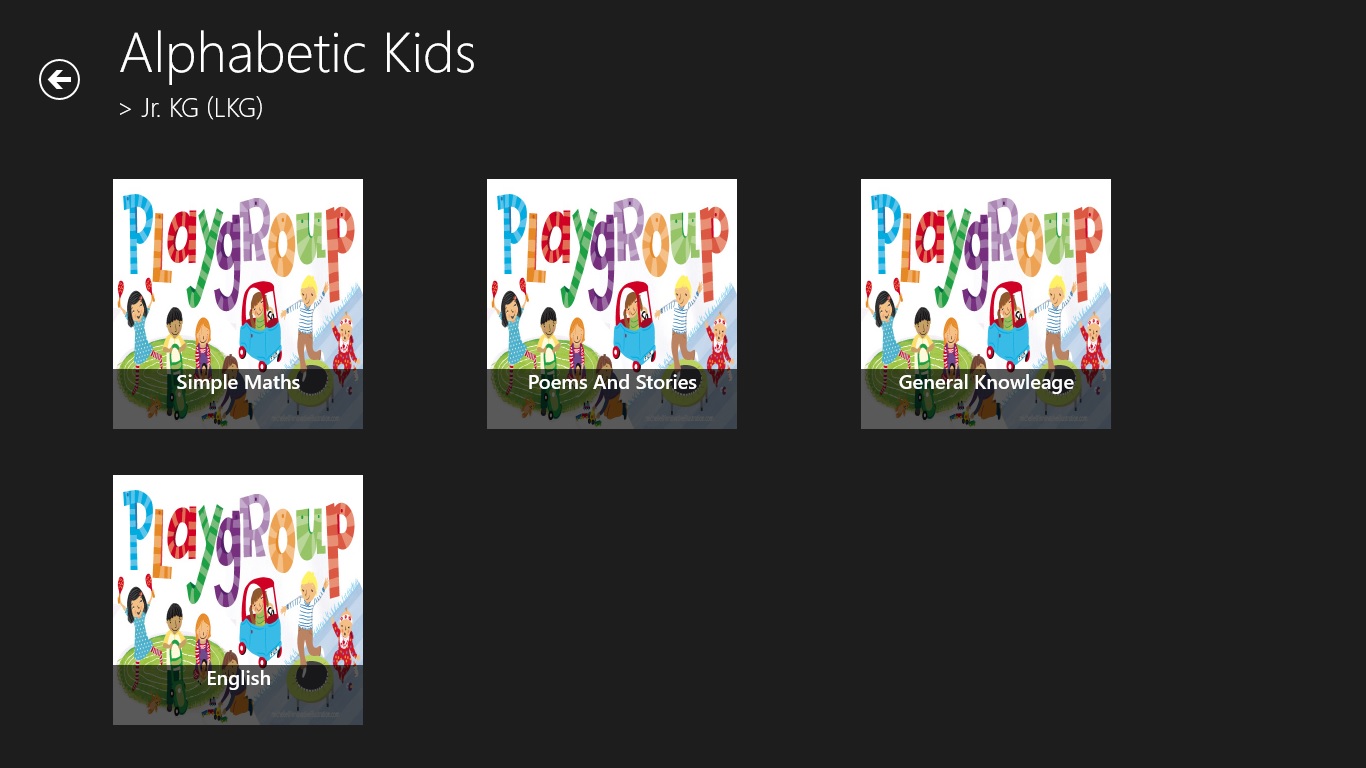


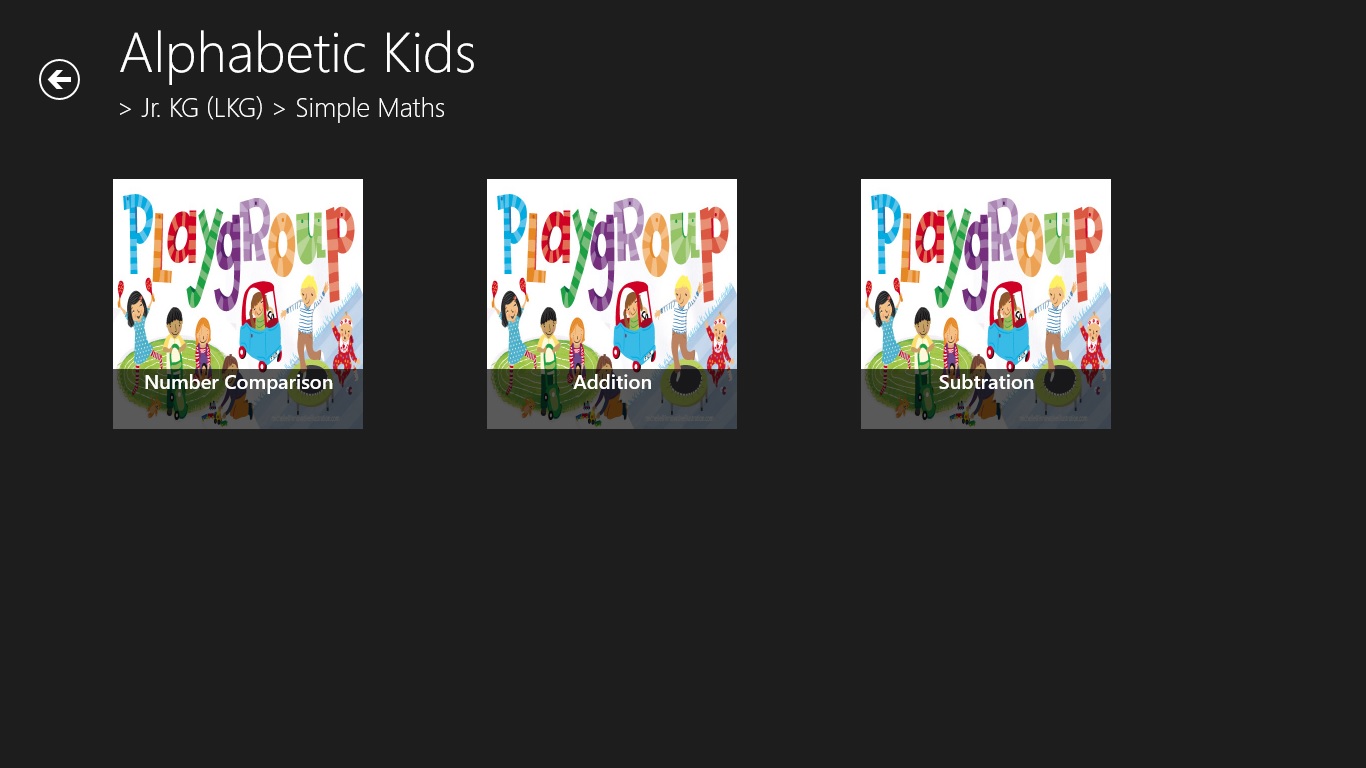


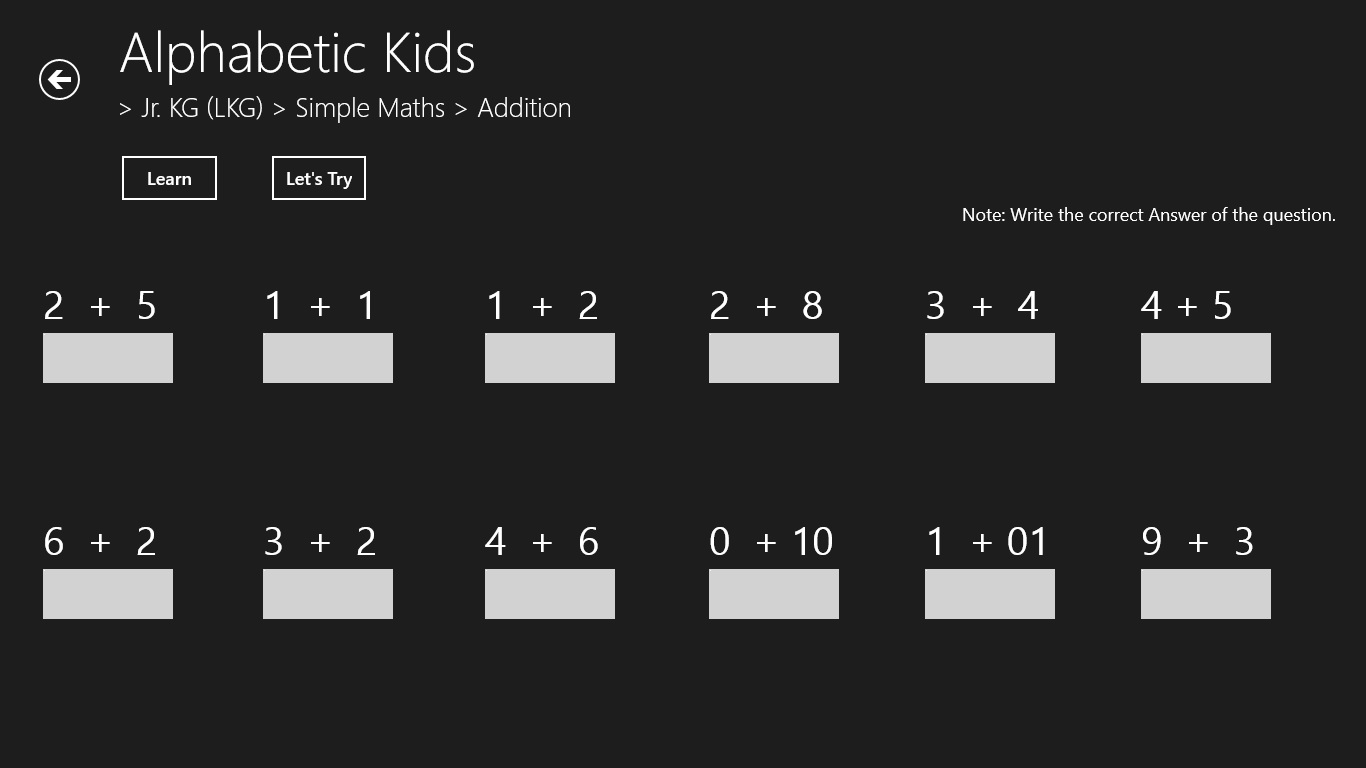


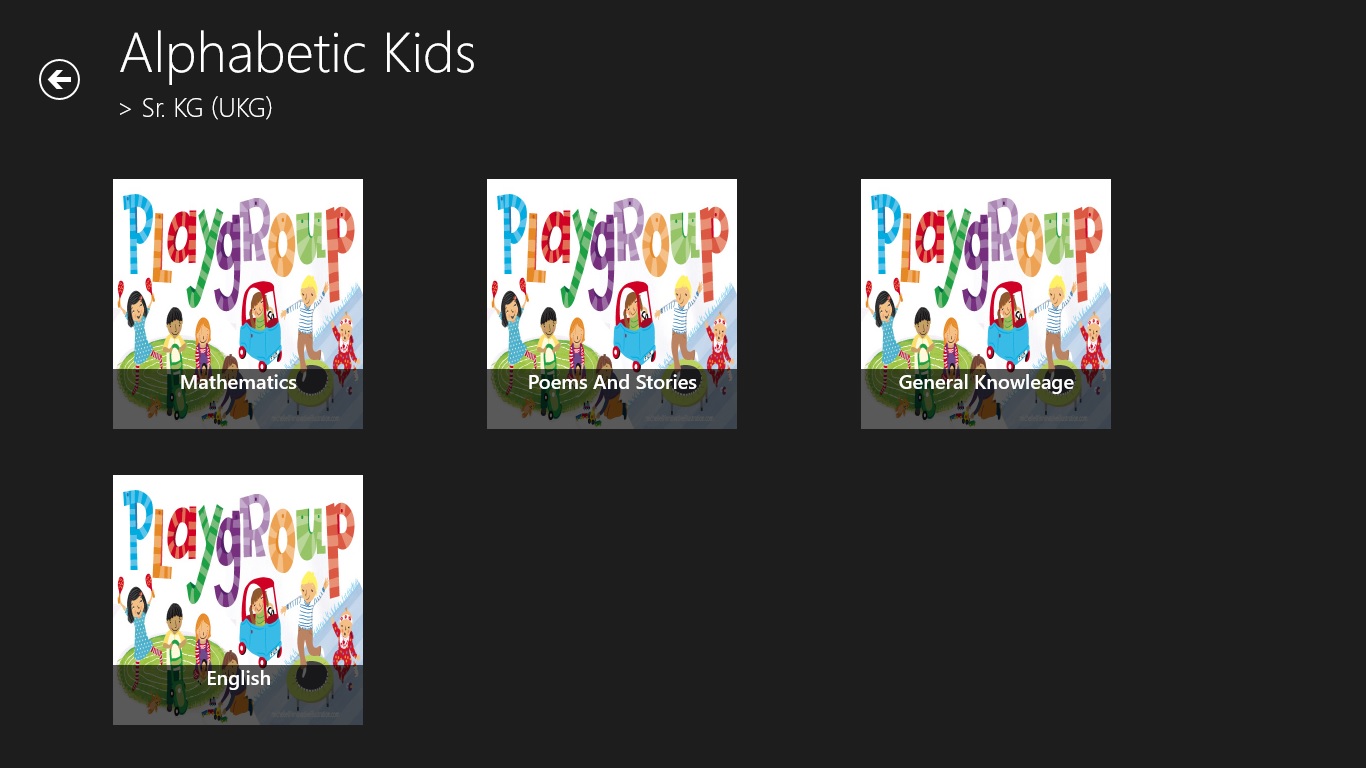


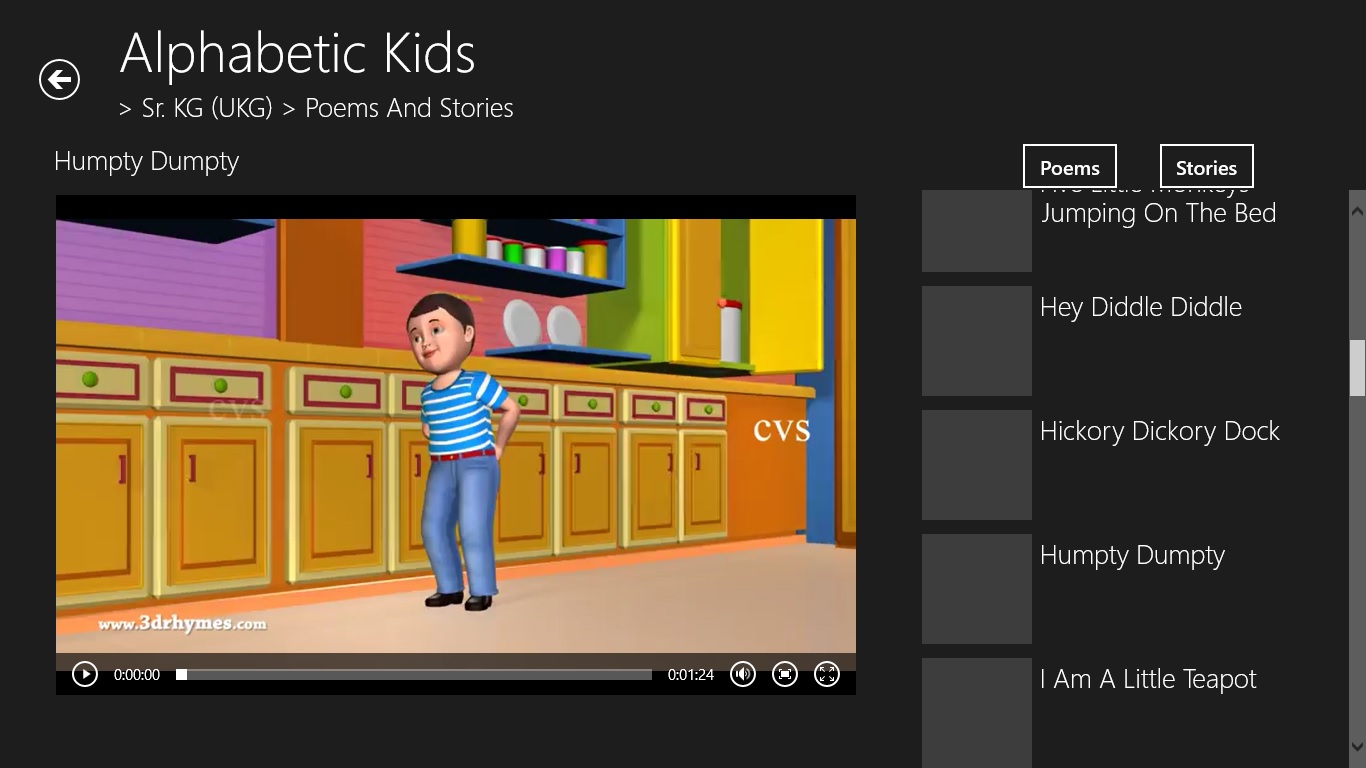












# 6. TESTING AND RESULTS:

## 6.2 Types of Testing

### 6.2.1 Unit Testing:

Unit testing means the verification and validation of software. In unit testing programmer tests each units of source code for fit to use.

Unit testing focuses on verifying the effort on the smallest unit of software module. Unit tested area to ensure that the module operates properly at boundaries established to limit.

The unit testing is while box oriented, and the step can be conducted in parallel for multiple components.

### 6.2.2 Integration Testing:

Integration testing takes input as a module that means, it is nothing but the set of unit tested module. It collects them in bigger aggregates and implements tests explained in an integration test plan.

Afterwards, distribute as it’s outcomes to the integrated system which is prepared for system testing. In the Integration testing (modules) the data can be tested across an interface. Conducting tests to uncover errors associated with interring while integration testing is a technique for constructing a program structure.

It having two types of testing:

1. Top-Down Integration Testing
2. Bottom-Up Integration Testing

# 7. SYSTEM DEVELOPMENT:

## 7.1 Analytical Model

The main objective of this research is to develop an Intelligent System using data mining modeling technique, namely, Naive Bayes. It is implemented as system based application in this user answers the predefined questions. It retrieves hidden data from stored database and compares the user values with trained data set. It can answer complex queries for diagnosing heart disease and thus assist healthcare practitioners to make intelligent clinical decisions, which traditional decision support systems cannot. By providing effective treatments, it also helps to reduce treatment costs.

Poor clinical decisions may lead to disasters and hence are seldom entertained. Besides, it is essential that the hospitals decrease the cost of clinical test. Appropriate computer-based information and/or decision support systems can aid in achieving clinical tests at a reduced cost. Owing to the accessibility of integrated information through enormous patient repositories, there is a swing in the insight of clinicians, patients and payers from qualitative visualization of clinical data to demanding a finer quantitative analysis of information with the assistance of all supporting clinical and imaging data.

## 7.2 Introduction of SDLC Model

* A typical Software Development life cycle consists of the following stages:



***Figure 7-1.****Software Development Life Cycle*

### Stage 1: Planning and Requirement Analysis:

* Requirement analysis is the most important and fundamental stage in SDLC. The senior members of the team perform it with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational, and technical areas.
* Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

### Stage 2: Defining Requirements:

* Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through ‘SRS’ – Software Requirement Specification document which consists of all the product requirements to be designed and developed during the project life cycle.

### Stage 3: Designing the product architecture:

* SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification. This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints , the best design approach is selected for the product.
* A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

### Stage 4: Building or Developing the Product:

* In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.
* Developers have to follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers etc are used to generate the code. Different high level programming languages such as C, C++, Pascal, **Java**, and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

### Stage 5: Testing the Product:

* This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However this stage refers to the testing only stage of the product where products defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

### Stage 6: Deployment in the Market and Maintenance:

* Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometime product deployment happens in stages as per the organizations’ business strategy. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).
* Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

## 7.3 Waterfall Model

### 7.3.1 Introduction:

* The Waterfall Model was first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.
* Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.



***Figure 7-2.****Waterfall Model*

* The sequential phases in Waterfall model are:

1. **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.
2. **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
3. **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
4. **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
5. **Deployment of system:** Once the functional and non functional testing is done, the product is deployed in the customer environment or released into the market.
6. **Maintenance:** There are some issues, which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

### 7.3.2 Applications:

* Requirements are very well documented, clear and fixed

1. Product definition is stable.
2. Technology is understood and is not dynamic.
3. There are no ambiguous requirements.

# CONCLUSION AND FUTURE SCOPE:

In this project, we have designed such a software which helps kids for better Learning. This project provides friendly environment to kids.

## Future Scope:

Our project also useful for Windows phone in future and also can be in Android. We could add more contents in our project in future.

# 9. REFERENCES:

1. Pete Brown, “Windows Store App Development: C# And XAML”, Manning Publications, June 2013.
2. For downloading images - <https://images.google.com/>
3. For get help In coding - <http://www.stackoverflow.com/>
4. Adam Freeman, “Windows 8 Apps Revealed Using XAML and C”, Apress Publications.
5. For tutorials of C# / XAML language - <https://channel9.msdn.com/>