Child Education Software

Arindam Ghosh

# Index

Contents

[Index 2](#_Toc351938867)

[Introduction: 4](#_Toc351938868)

[Objective 6](#_Toc351938869)

[System Analysis 6](#_Toc351938870)

[Identification of Need: 6](#_Toc351938871)

[Preliminary Investigation: 7](#_Toc351938872)

[Feasibility Study: 7](#_Toc351938873)

[Technical Feasibility 7](#_Toc351938874)

[Operational Feasibility 8](#_Toc351938875)

[Economic Feasibility 8](#_Toc351938876)

[Project Planning & Scheduling: 8](#_Toc351938877)

[Gantt chart 8](#_Toc351938878)

[Tracking Gantt 9](#_Toc351938879)

[Pert Chart 9](#_Toc351938880)

[Software requirement specifications (SRS): 10](#_Toc351938881)

[Functional Requirement 10](#_Toc351938882)

[Non-functional Requirements 12](#_Toc351938883)

[Data models 13](#_Toc351938884)

[Context Diagram 13](#_Toc351938885)

[0-Level DFD 13](#_Toc351938886)

[1-Level DFD 14](#_Toc351938887)

[2-Level DFD 16](#_Toc351938888)

[Sequence diagrams 17](#_Toc351938889)

[Entity Relationship Model, 17](#_Toc351938890)

[E-R Diagram 18](#_Toc351938891)

[Class Diagrams 19](#_Toc351938892)

[Activity Diagrams 20](#_Toc351938893)

[User Login 20](#_Toc351938894)

[Lesson Management 21](#_Toc351938895)

[Exercise Management 21](#_Toc351938896)

[View Report 22](#_Toc351938897)

[Sync 23](#_Toc351938898)

[Payment Management 24](#_Toc351938899)

[System Design 25](#_Toc351938900)

[Modularisation details 25](#_Toc351938901)

[Database & Table Details 25](#_Toc351938902)

[Complete Structure 26](#_Toc351938903)

[Module Description 26](#_Toc351938904)

[Child Education Software GUI: 26](#_Toc351938905)

[Child Education Software Engine: 30](#_Toc351938906)

[Tutorial Controller: 30](#_Toc351938907)

[User Interface Design 30](#_Toc351938908)

[Main window 30](#_Toc351938909)

[Open Video 30](#_Toc351938910)

[Play video 31](#_Toc351938911)

[Pause video 31](#_Toc351938912)

[Photo 32](#_Toc351938913)

[Typing 32](#_Toc351938914)

[Brush size 33](#_Toc351938915)

[Select color 33](#_Toc351938916)

[Erase 34](#_Toc351938917)

[Selection 34](#_Toc351938918)

[Test Cases (Unit Test Cases and System Test Cases) 35](#_Toc351938919)

[Coding 35](#_Toc351938920)

[Complete Project Coding 35](#_Toc351938921)

[Comments and Description of Coding segments 35](#_Toc351938922)

[Standardization of the coding 36](#_Toc351938923)

[Code Efficiency 36](#_Toc351938924)

[Error handling 36](#_Toc351938925)

[Parameters calling/passing 36](#_Toc351938926)

[Validation checks 36](#_Toc351938927)

[Testing 36](#_Toc351938928)

[Testing techniques and Testing strategies used 36](#_Toc351938929)

[Testing Plan used 36](#_Toc351938930)

[Test reports for Unit Test Cases and System Test Cases 36](#_Toc351938931)

[Debugging and Code improvement: 36](#_Toc351938932)

[System Security measures: 36](#_Toc351938933)

[Database/data security: 36](#_Toc351938934)

[Creation of User profiles and access rights 37](#_Toc351938935)

[Cost Estimation of the Project along with Cost Estimation Model 37](#_Toc351938936)

[Estimation of development effort 37](#_Toc351938937)

[Estimation of development time 37](#_Toc351938938)

[Reports 38](#_Toc351938939)

[Future scope and further enhancement of the Project 38](#_Toc351938940)

[Bibliography 39](#_Toc351938941)

[Website 39](#_Toc351938942)

[Books 40](#_Toc351938943)

[Appendices 40](#_Toc351938944)

[Mobile App Development 40](#_Toc351938945)

[Nokia SDK 2.0 for Java — for Series 40 apps 40](#_Toc351938946)

[Nokia Web - Tools 41](#_Toc351938947)

[Cacoo:: online drawing tool 45](#_Toc351938948)

[Creating Diagrams 45](#_Toc351938949)

[Collaboration 46](#_Toc351938950)

[Sharing Diagrams 46](#_Toc351938951)

[Managing Diagrams 46](#_Toc351938952)

[Languages and Time Zones 46](#_Toc351938953)

[Security 46](#_Toc351938954)

[API 46](#_Toc351938955)

[GitHub 47](#_Toc351938956)

[Description 47](#_Toc351938957)

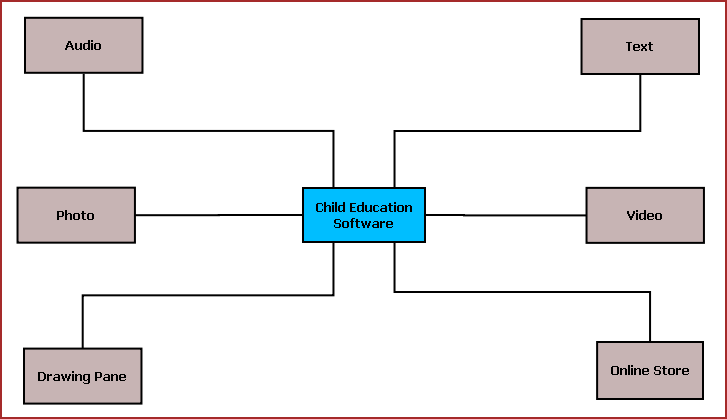
[Limitations and constraints 47](#_Toc351938958)

[Glossary. 47](#_Toc351938959)

# Introduction:

Child Education Software will enable a new aspect of teaching children with the help of modern technology. It will incorporate audio visual teaching method with user friendly interface. Almost every family is having a computer now. We are used to computerized system in every walk of our life. So kids also will be attracted towards computer. We can use computer as a mean of teaching new things. Because of rich multi-media content kids will be happy to use computer for learning new topics.

In Child Education Software users can use the existing tutorial as well as they can download new tutorial from online store. It will be an eco-friendly option. Children wastes paper a lot and they get bored as the colour of the fades gradually. The child education software will make learning more efficient, attractive and tech-savvy for the children.



**Figure 1:** Overview of Child Education Software

The main features of this software are listed below:

* + Textual display of teaching material.
  + Video display for video tutorial.
  + Sound system to play audio clips.
  + Drawing Pane or Handwriting practice area.
  + Photo viewer.
  + Easy navigation.
  + Interactive UI.
  + Printing of selected content.
  + Online store for buying new tutorials and upgrades.

## Objective

The objectives of the projects are given below:

* To provide a new kind of medium for educating children
* To cop up with technological revolutions happening around us and to involve the kids with new technology
* To make education process more interesting to the kids
* To utilize our electronic gadgets to do more meaningful work like teaching children
* To save paper waste and to take more green initiatives
* By accomplishing this project I could learn new technologies like .NET, C#, XML and I am able to be involved with the complete software development lifecycle.

# System Analysis

## Identification of Need:

Now we are used to computerized system in every walk of our life and kids also will be attracted towards computer. We can use computer as a mean of teaching new things. Because of rich multi-media content kids will be happy to use computer for learning new topics. Since we are also dedicated to the education we are always searching for ways to create educational material which starts from "simple" alphabets to complex grammar exercises and study material for other subjects such as science, mathematics, history etc. Our primary focus is on schoolchildren aged 3 to 18, and the specialized user interface needs of young users. However, we will also have programs to aid teachers in planning lessons, and others that are of interest to university students and anyone else with a desire to learn.

CES will allow participants to have lessons from offline resources as well as online study materials. They can also do a hands-on with the exercises and evaluate their progress. Additional study material and tool can be availed by purchasing modules from various online resources.

## Preliminary Investigation:

From everywhere I tried to spoke with the parents and guardian about this modern system of education. All they agree with may think about this application. I tried to collect opinion from software experts and took their opinion to develop the child education system.

## Feasibility Study:

Feasibility study is an essential requirement of any proposed system. It proposes one or more conceptual solutions to the problem set for the project. It is the review of the findings analyzed so far. Before proceeding directly into the design phase I had to check whether this project is worth doing. For this I carried out feasibility study of the proposed system in 3 different ways

* To check whether the project is technically feasible that is whether I have the necessary skills and know-how to complete the project.
* To check whether the proposed system is easy to use and that it satisfies the user objectives (operational feasibility) and can be fitted into current system operation.
* To determine whether the project’s goal can be achieved within the resource limits allocated that is to find out whether the project is economically feasible.

### Technical Feasibility

It has been already mentioned that ‘**Child Education System’** is purely a desktop based project with some mobile application and related web-based project. A lots of such types of projects have already been made or been running to make every day. So it is not even technically impossible rather difficult to build such a software. The technical software knowledge that is required for desktop is C#, Mysql, WPF for mobile apps is nokia web tool and some anyone who has a basic knowledge of computer software and has a good sense of design can almost be an expert in handling these software. As the project will proceed one can easily make himself informative about ASP to do the project. So not very much technical capability is required by the project. We see that the project is technically feasible and worth to do because of the reasons specified above.

### Operational Feasibility

To know whether the proposed system is acceptable to the end user, they were subjected to a mini prototype. The users were asked to respond as to how they feel about the system. To a large extent the users were satisfied with the prototype, which I have shown to them, minor modifications were also done to closely match with the users’ requirement.

To implement the user requirements in a software system is the true goal of a systems analyst. I rigorously followed the requirements of the user, what they want from the system and how it should help them. Prototyping results show that the proposed system is acceptable to the user.

### Economic Feasibility

If it is seen from real life view then MY PROJECT IS ECONOMICALLY FEASIBLE as this sort of software is going to get the market. As these types of engineering colleges emerge they all will try to communicate with at most people they can. When this communication is required portal will be the most wanted communication medium that they rely for. So this type of web-based project will always have a commercial aspect. So I am not wasting my time in a non-economic feasible project.

Technology is growing up day by day. The parents can easily afford to buy various software for their children’s education. If we keep the cost in comfortable label with the basic feature, which help to maintain the child education system, we can sell the application. And according to their need we can modify the applications and customize it for their help.

## Project Planning & Scheduling:

### Gantt chart



### Tracking Gantt



### 

### Pert Chart



## Software requirement specifications (SRS):

### Functional Requirement

#### Audio visual display of study materials:

**Introduction**:

User should be able to watch animated video lessons as well as audio lessons.

**Input**:

Click on show lesion button.

**Processing**:

System will find and fetch a predefined lesion and display that.

**Output**:

Kids can watch them and learn new things.

#### Exercise for students:

**Introduction**:

There should be relevant and user friendly exercise topics.

**Input**:

User will click on start practice button.

**Processing**:

System will open relevant practice set for kids from database.

**Output**:

Kids can interact with the graphical practice sets and solve them.

#### Drawing pane for practice

**Introduction**:

There should be a drawing pane for kids with an attractive GUI.

**Input**:

User will click on start drawing button and kids will start interacting with the GUI.

**Processing**:

Drawn shapes will be saved temporarily in memory and could be permanently kept in database.

**Output**:

Stored drawings will be shown to the user whenever is asked.

#### Download new lesions:

**Introduction**:

New lesions could be downloaded from the web if user wants.

**Input**:

User clicks on download new lesions button.

**Processing**:

System searches for relevant data in the previously stored online database and copies them in the local system.

**Output**:

System shows the newly downloaded lesions to the user.

#### Learn about various topics like nature, economy etc.

**Introduction**:

There should be various GK tutorials with relevant images and examples.

**Input**:

User clicks on the learn button.

**Processing**:

Various options provided to the user and as per his choice a new image and description gets opened.

**Output**:

Students get to see topics with images and examples that help them remember the topics better.

#### Online payment

**Introduction**:

For purchasing lesions, user should be able to pay online using debit card, internet banking, credit card etc.

**Input**:

User opts for online payment option in the application GUI.

**Processing**:

After getting required details, system takes user to his banker’s website and deducts the required amount. Generates a online bill as well.

**Output**:

User gets the digital bill’s copy in his email inbox and gets the new lesion in his system as well.

#### Easily sharable downloaded lesions:

**Introduction**:

People should be able to share the lesions to other computers.

**Input**:

User clicks on share the lesions button

**Processing**:

System takes the tutorials from database and sends them to the portable device as per user choice for sharing.

**Output**:

User gets the data and could easily copy them to the other portable device.

#### Generate grade card

**Introduction**:

After completing exercises, a grade card should be generated.

**Input**:

Kids complete the exercises within the given time.

**Processing:**

After matching the answers with predefined answers, system generates a report card and saves them in database.

**Output**:

User can see the report card after the exam and see them in future as well.

## Non-functional Requirements

* The application will be self-dependent and no dependency on other parties required.
* There will be a digital backup and restore system.
* There will be more opportunity to extend the application in various type of device in future.
* The response time will be low and the system will response fast.
* It will be very user friendly and usable by any person with minimal computer knowledge.
* In terms of security unauthorized access will be denied and register user will be able to change as necessary.
* It will be efficient as it reduces manual labor and searching.
* DNBSN will have user manual and help documents.
* It is designed such a way that it can be maintained with minimal effort.

## Data models

### Context Diagram

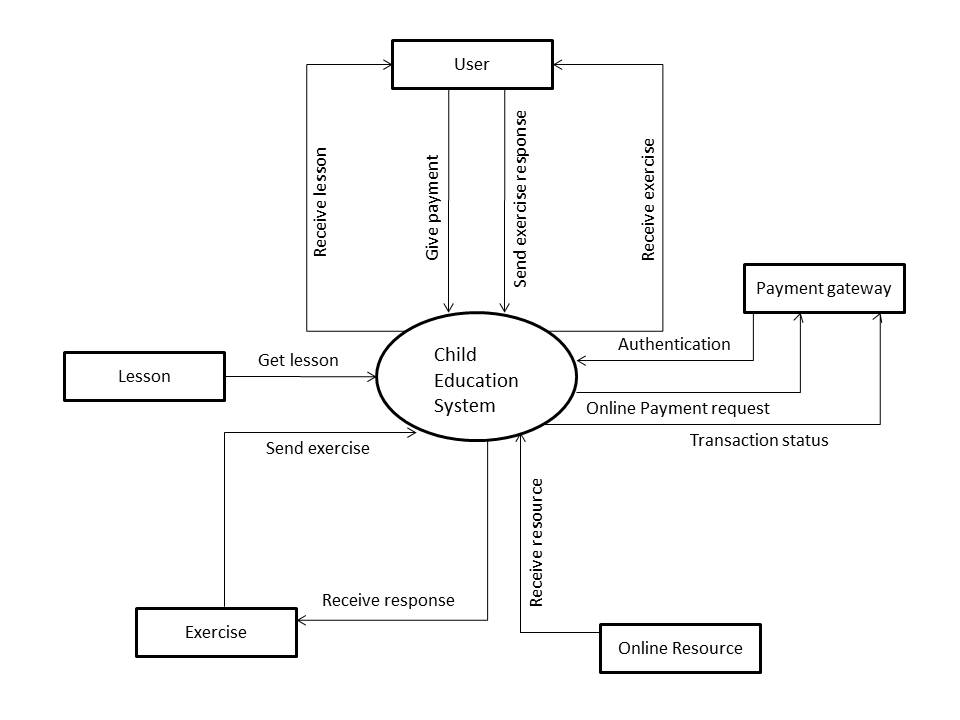


Figure 1: Context Diagram of CES

### 0-Level DFD

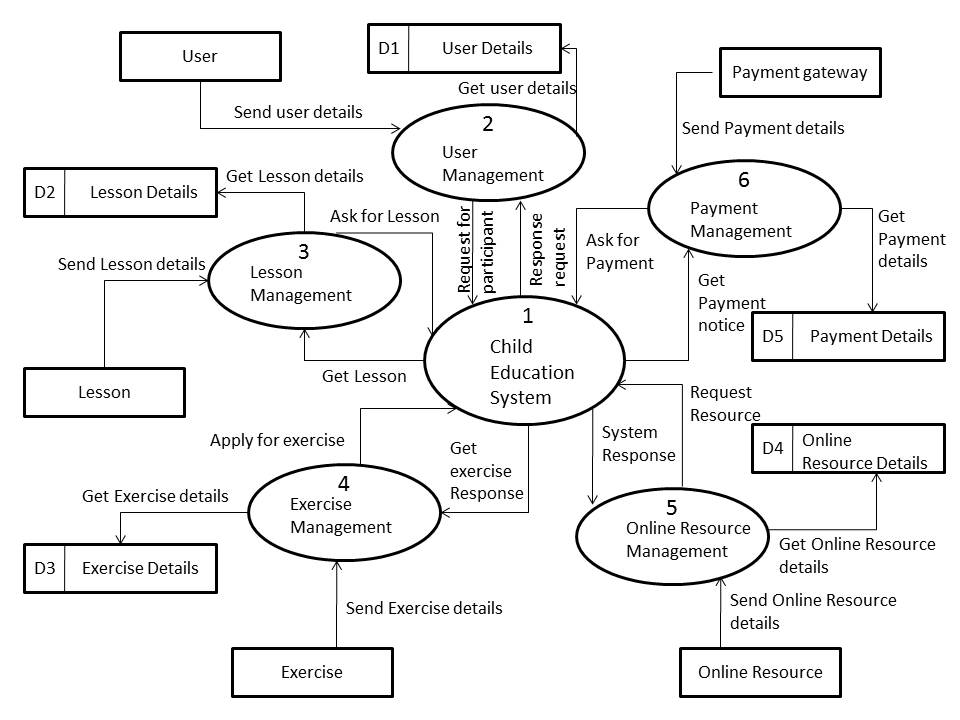


Figure 2: DFD level 0 of CES

### 1-Level DFD

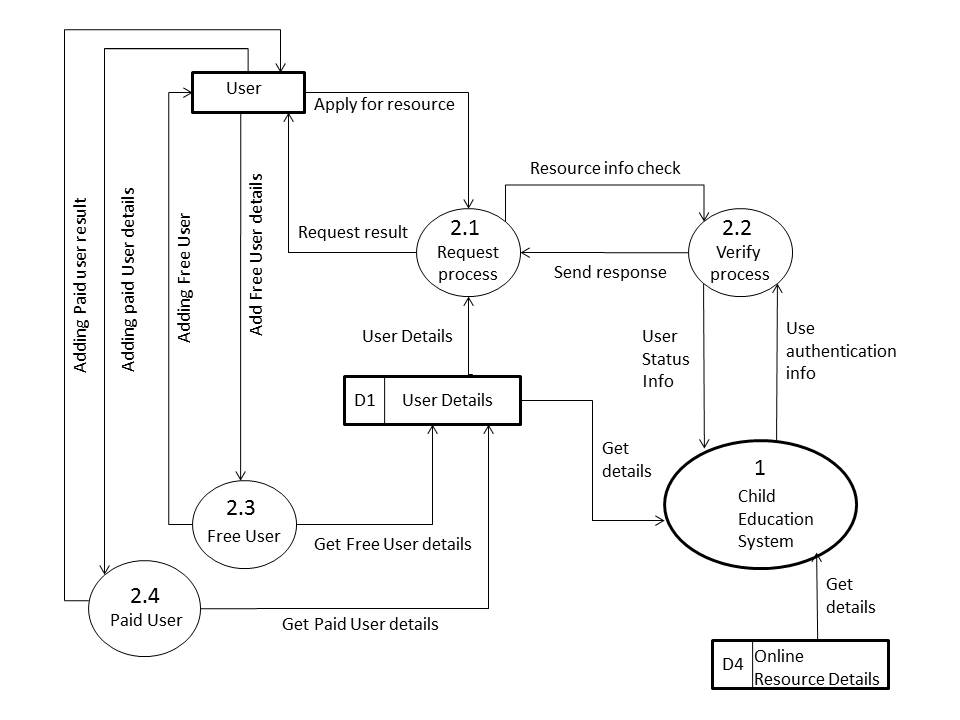


Figure 3: DFD level 1 of CES

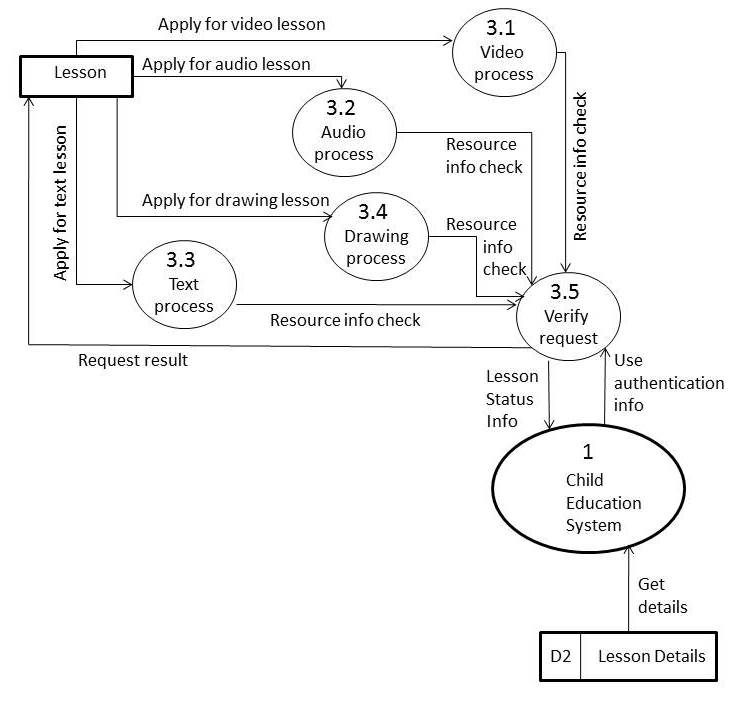


Figure 4: DFD level 1 of CES

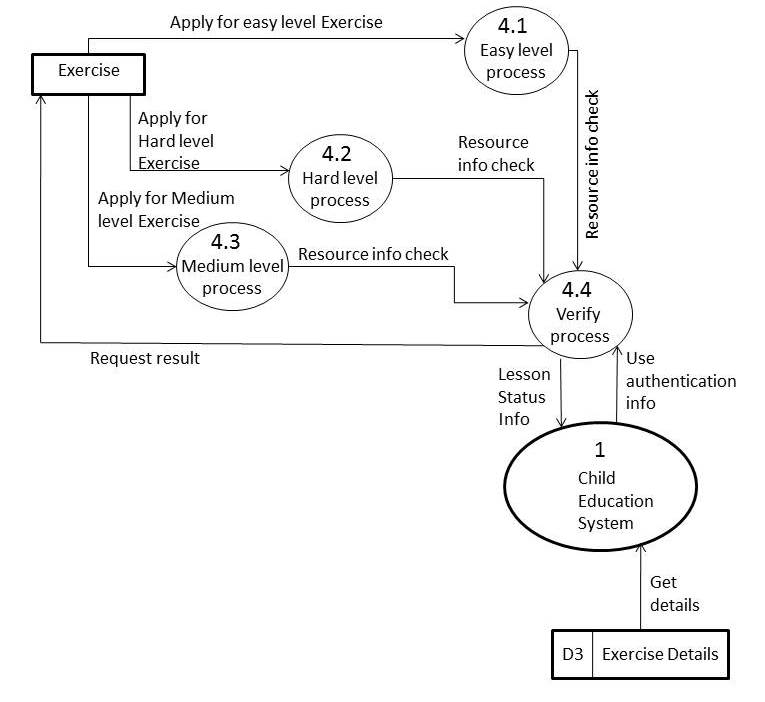


Figure 5: DFD level 1 of CES

### 2-Level DFD

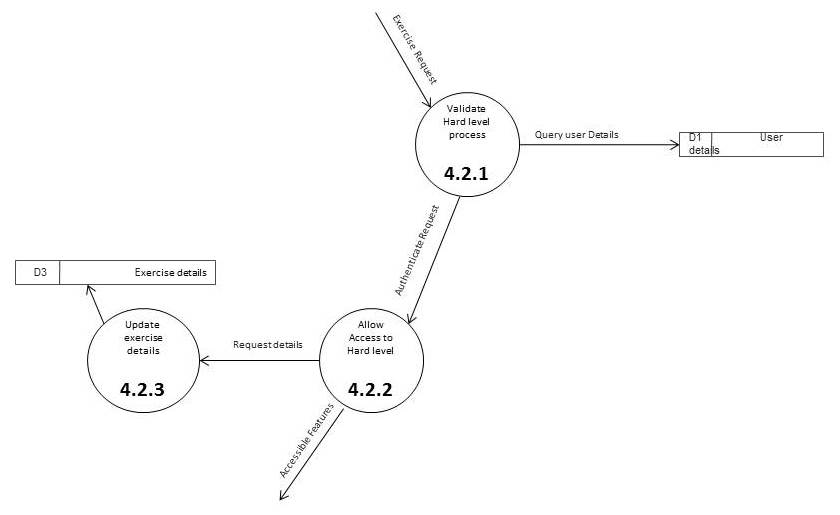


Figure 6: DFD level 2 of CES

## Sequence diagrams

## Entity Relationship Model,

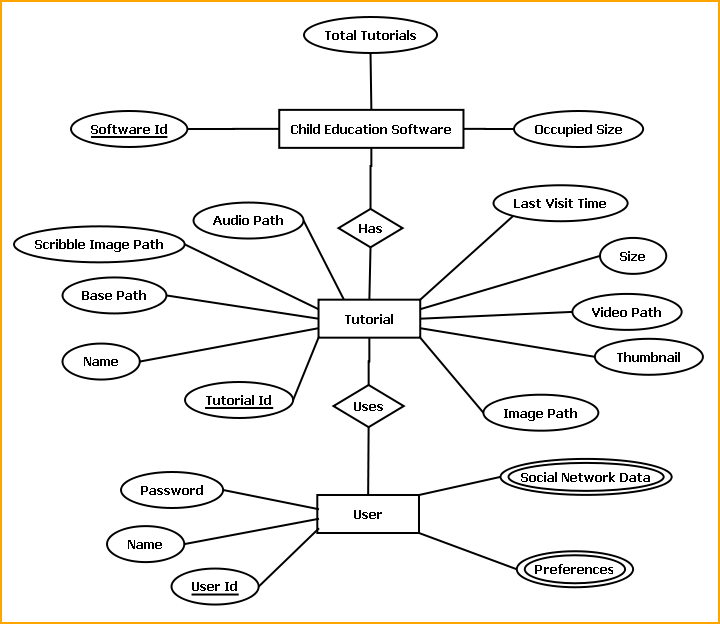
We will design a RDBMS for Child Education Software. The entities and their attributes are listed below. Attributes in Bold letter is the unique key.

|  |  |
| --- | --- |
| **Entities** | **Attributes** |
| Tutorial | **Tutorial Id**, Name, Base Path, Last Visit Time, Size, Thumbnail, video path, image path, audio path, scribble image path |
| Child Education Software | **Software Id**, Total tutorials , Occupied Size |
| User | **User Id**, Name, Social Network Data, password, Preferences. |

**Relationship between Entities:**

* Child Education Software has Tutorials 🡪 1 : N
* User uses tutorials 🡪 M : N

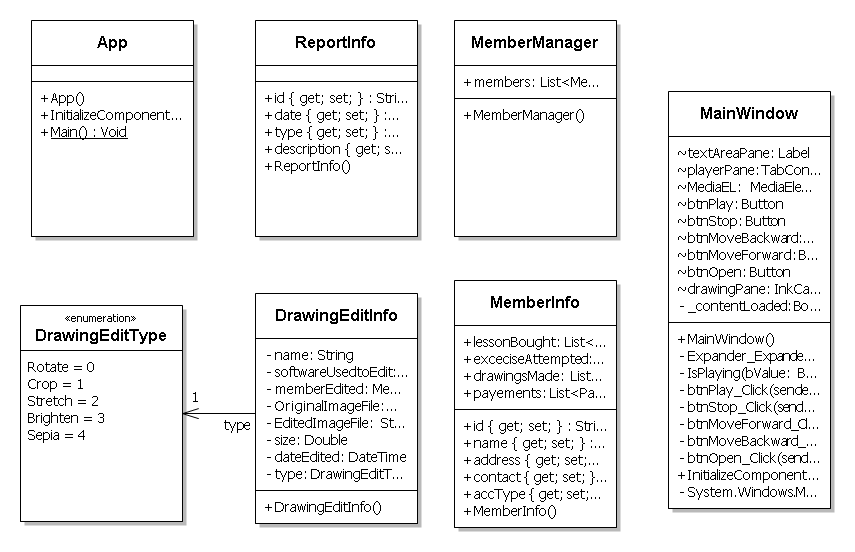
## E-R Diagram

****

E-R Diagram of Child Education Software

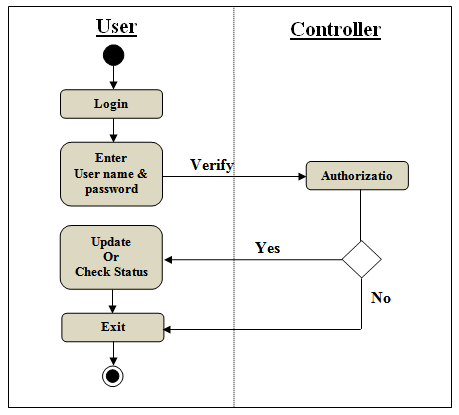
## Class Diagrams

****

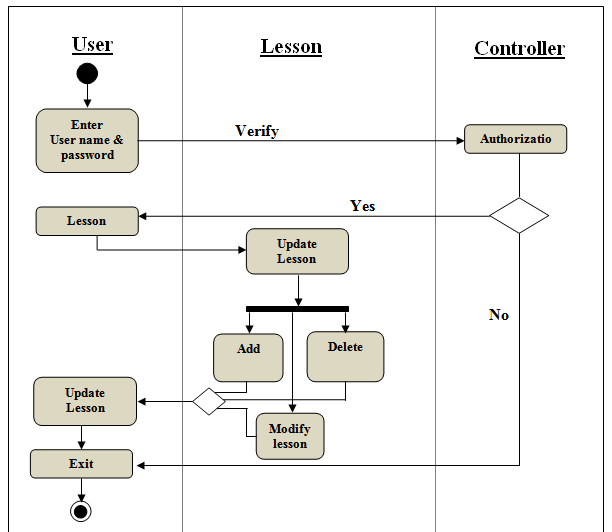


## Activity Diagrams

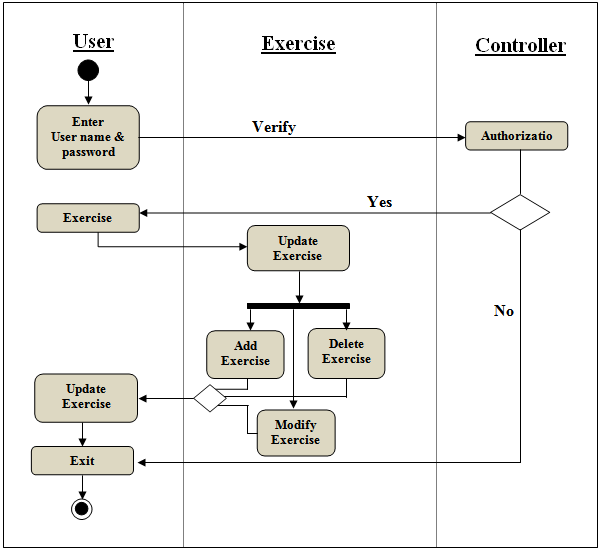
### User Login



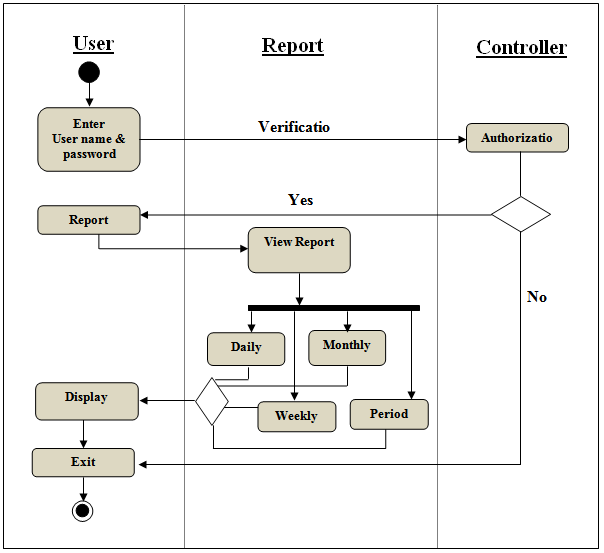
### Lesson Management



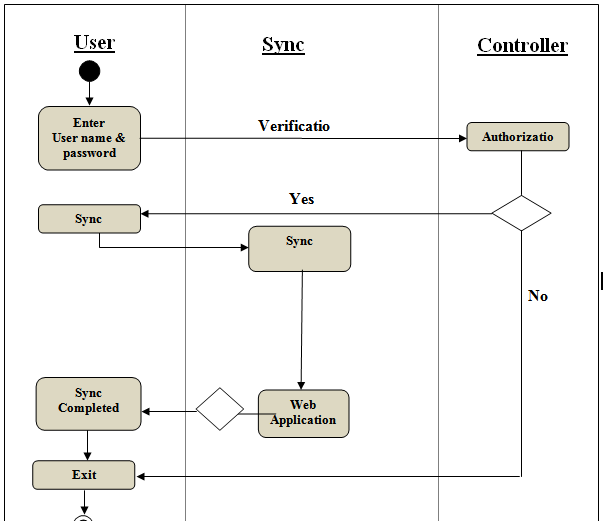
### Exercise Management



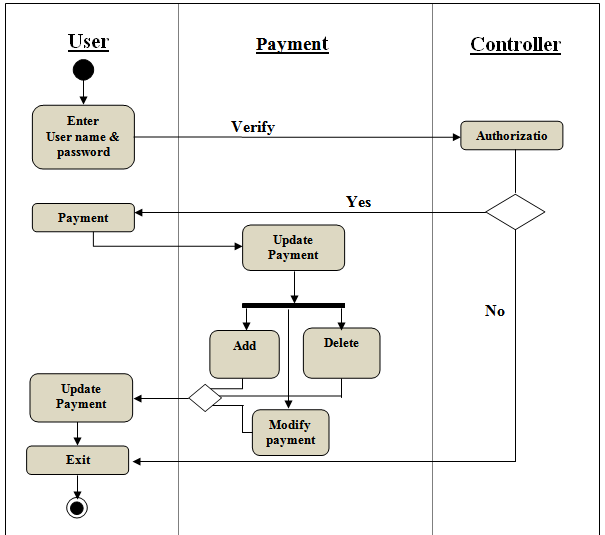
### View Report



### Sync



### Payment Management



# System Design

## Modularisation details

# Database & Table Details

The database used for this software is called CESdb. Database tables and corresponding keys are shown in tabular form. It shows the tables and its columns. A key in Bold is the primary key.

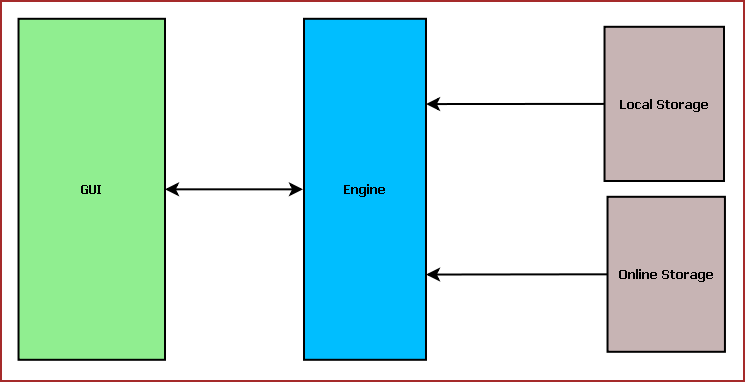
|  |  |
| --- | --- |
| **Tables** | **Keys** |
| Tutorial | **Tutorial Id**, Name, Base Path, Last Visit Time, Size, Thumbnail, video path, image path, audio path, scribble image path |
| Child Education Software | **Software Id**, Total tutorials , Occupied Size |
| User | **User Id**, Name, Social Network Data, password, Preferences. |

# Complete Structure

## Module Description

Child Education Software is divided into three main modules.

* GUI
* Engine
* Local Storage & Online Storage



**Figure:** Child Education System Components

### Child Education Software GUI:

Child Education Software GUI will display tutorial, navigator, online store interface. The main components of GUI are:

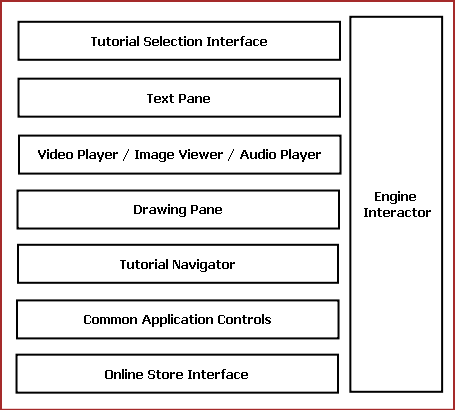
* Tutorial Selection Interface
* Text Pane
* Video Player / Image Viewer / Audio Player
* Drawing Pane
* Tutorial Navigator
* Common Application Control
* Online Store Interface
* Engine Interactor

#### Tutorial Selection Interface:

This interface will display the available tutorials from local storage and the tutorials bought from Online Store. From here the user will select the tutorial to be viewed. The tutorials will be displayed according to different category such as: Age, Subject, culture etc.

#### Text Pane:

This area will display the text portion of the tutorial. This will contain the description about the subject if different colours. Users can select text, copy and paste the text.



**Fig:** Components of Child Education Software GUI

#### Common Controls:

Common controls include several controls of the application. Such as: Menu Bar, Toolbar, status bar, Context Menu and progress bar.

#### Engine Interaction:

This module handles the interaction between GUI and Engine. This module defines all callback for the Engine events.

#### Video Player / Image Viewer / Audio Player:

This Pane displays the image, video & audio artwork associated with the tutorial content. This will have two components such as: Display Area & Control Pane. Display Area will display the content. Control Pane contains the control like Play, Pause, Previous, Next for controlling the displayed content.

#### Drawing Pane:

Drawing Pane allows user to draw, sketch & scribble. This will help the kids to draw and practice while learning. User can save the drawing as a Image or load a image to scribble.

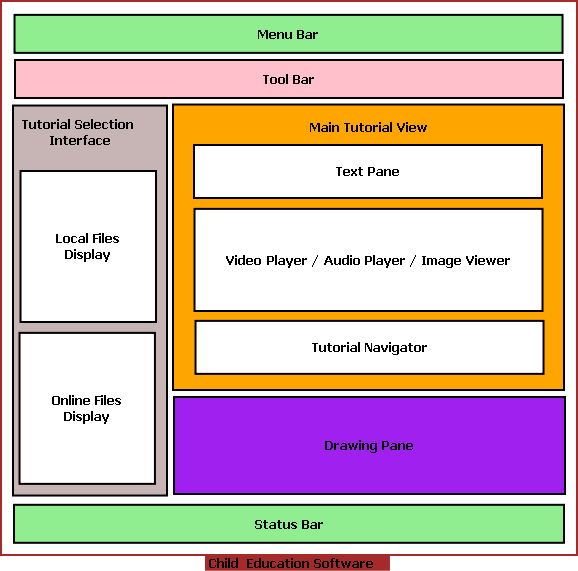
#### Tutorial Navigator:

This pane contains the controls for navigating within the tutorial. It will have “Next”, “Previous” and “Go to Home” options.

#### Online Store Interface:

This pane allows user to login into Online Store. Then it will display the available tutorial online for download. User may need to purchase or download it for free.

The GUI layout of Child Education Software is shown below:



### Child Education Software Engine:

Child Education Software Engine provides multimedia framework for displaying Video/Image, tutorial controlling logic. It is mainly divided into following components:

* Tutorial Controller
* Image Controller
* Archive Manager
* Video /Audio Controller
* Online Store Controller
* GUI Interactor
* Database Manager

### Tutorial Controller:

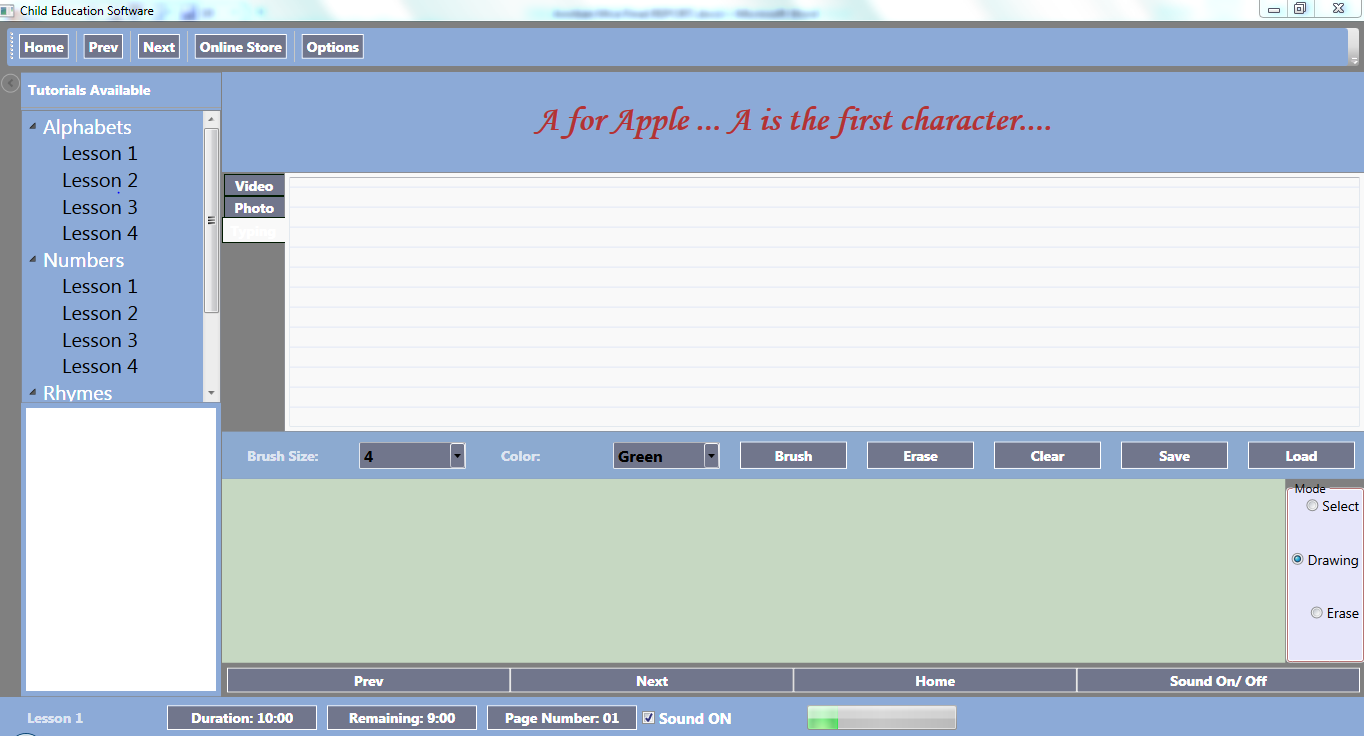
This module defines the tutorial format. It controls tutorial flow, content to be displayed. Every tutorial is internally a XML file mentioning the tutorial contents.

Tutorial controller has a XML parser to read the XML and load tutorial contents accordingly. Tutorial Controller controls the other engine components to coordinate tutorial display.

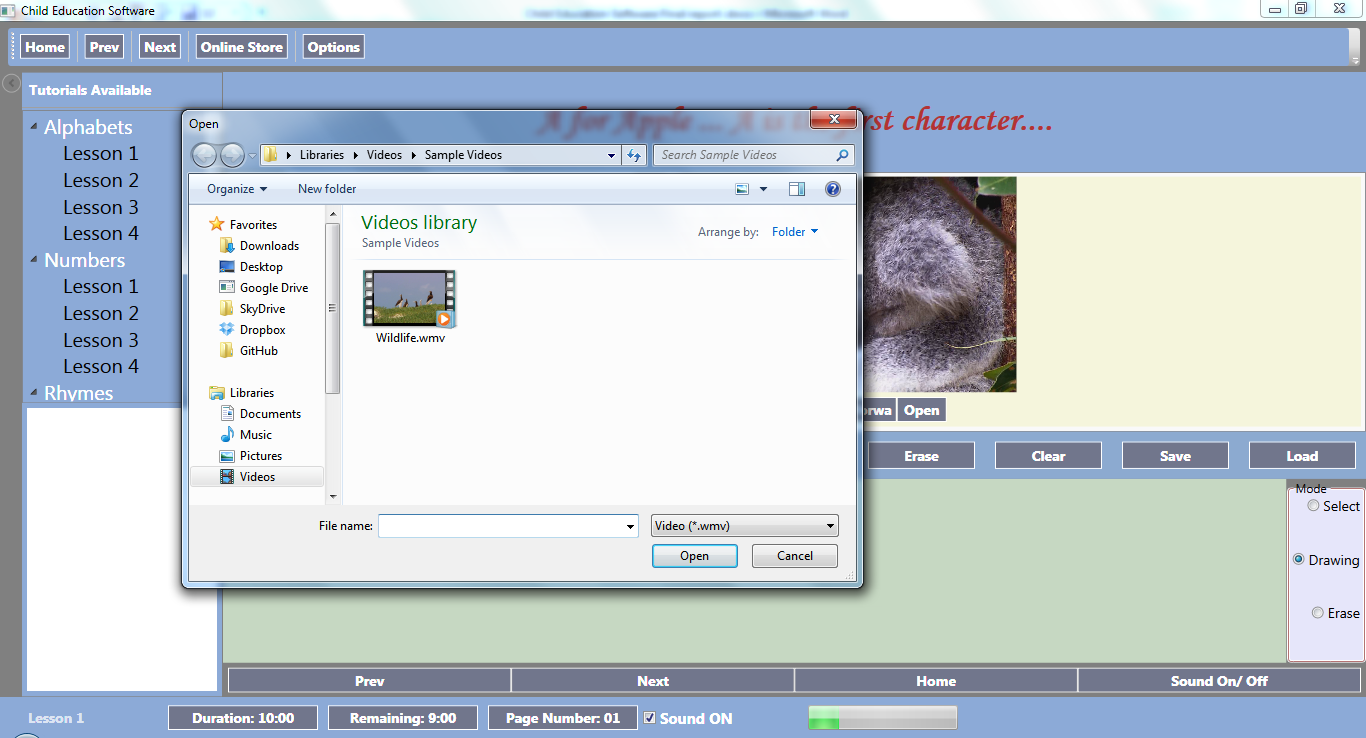
Different components of Child Education Software Engine and their interaction with other modules are displayed in the following diagram.

## User Interface Design

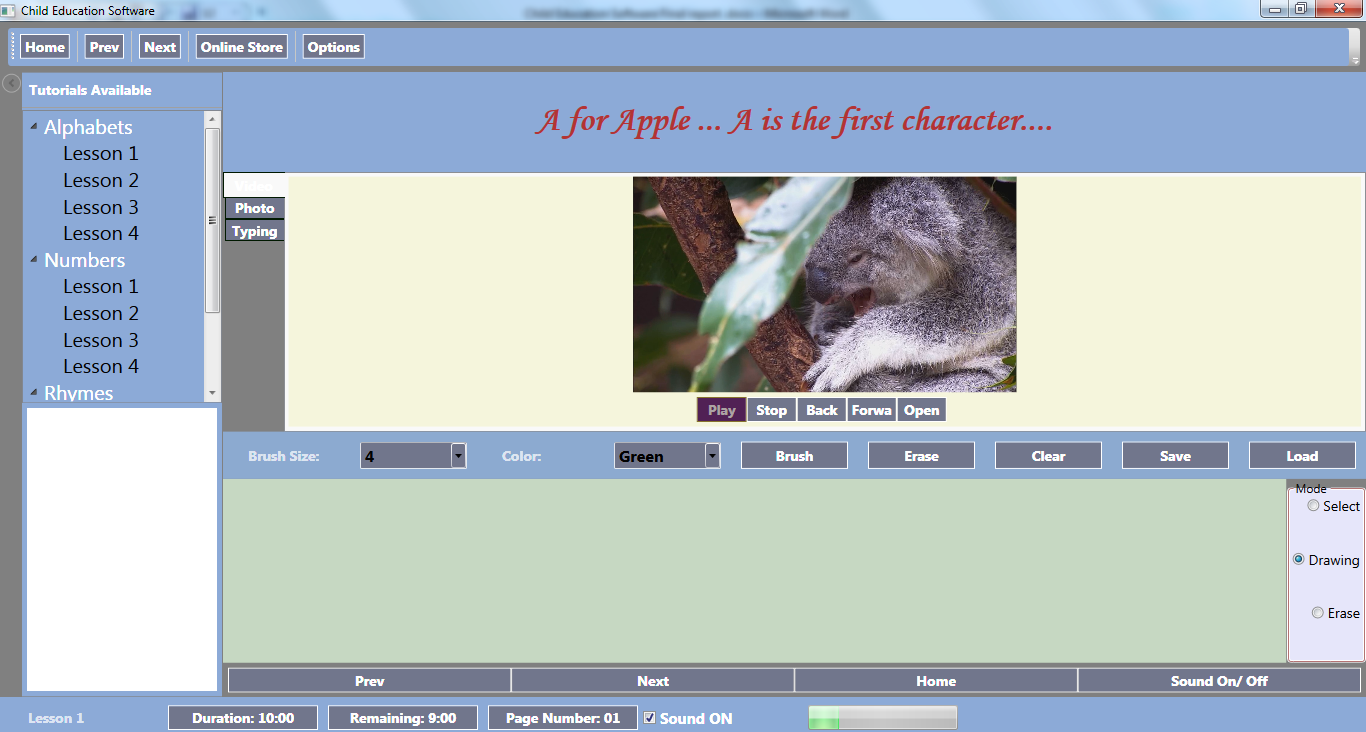
### Main window



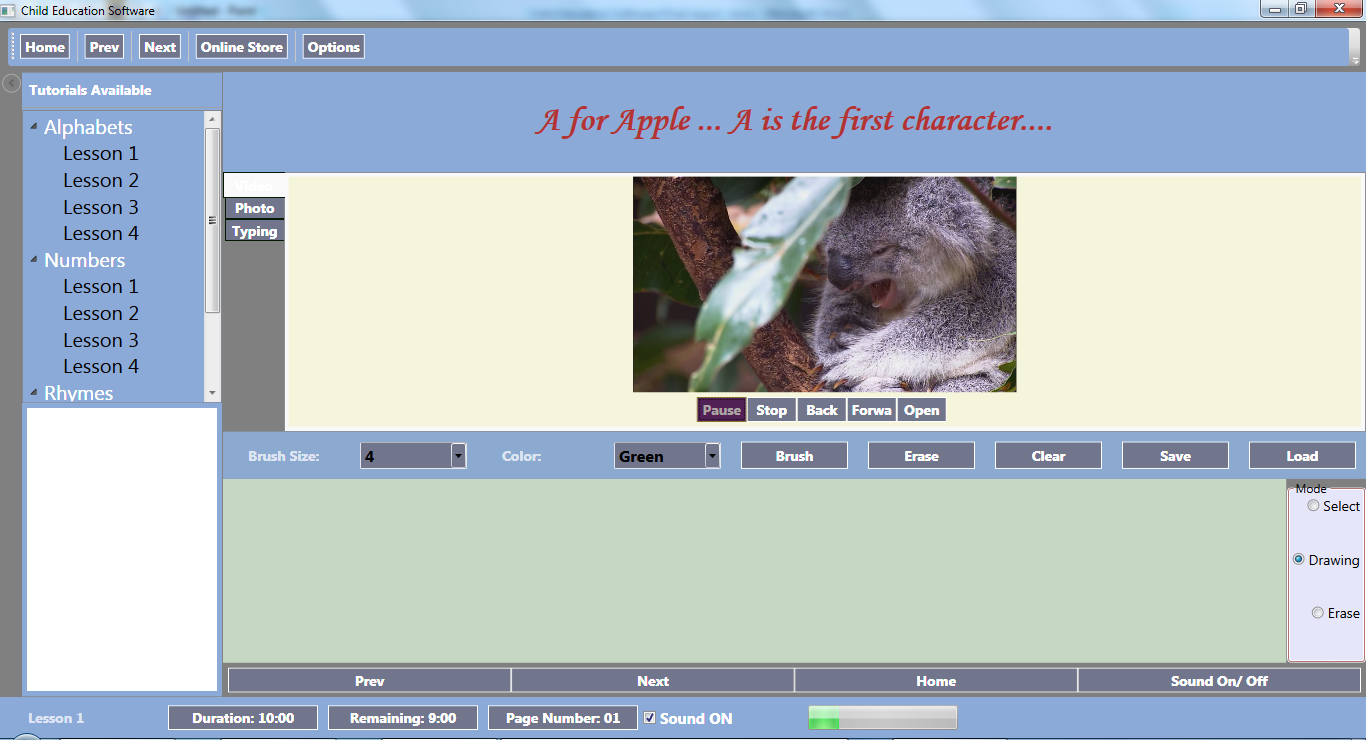
### Open Video



### Play video



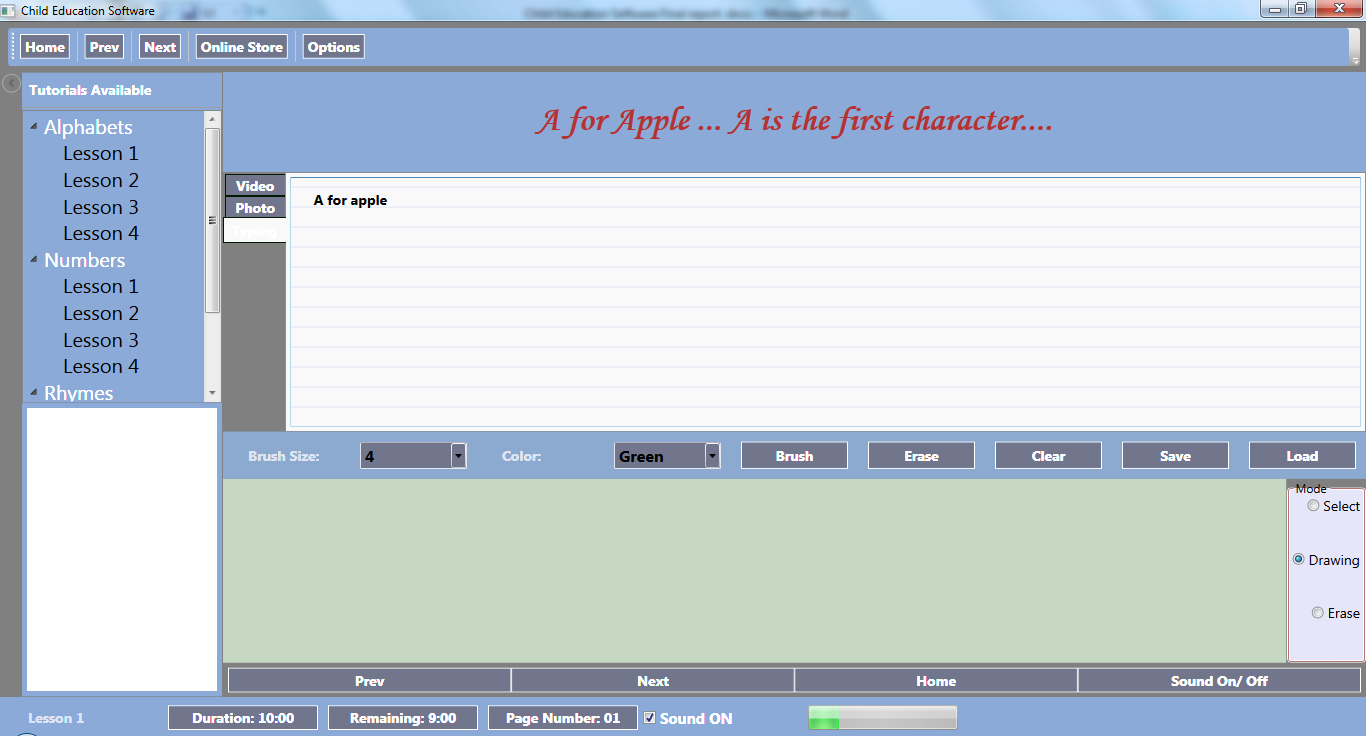
### Pause video



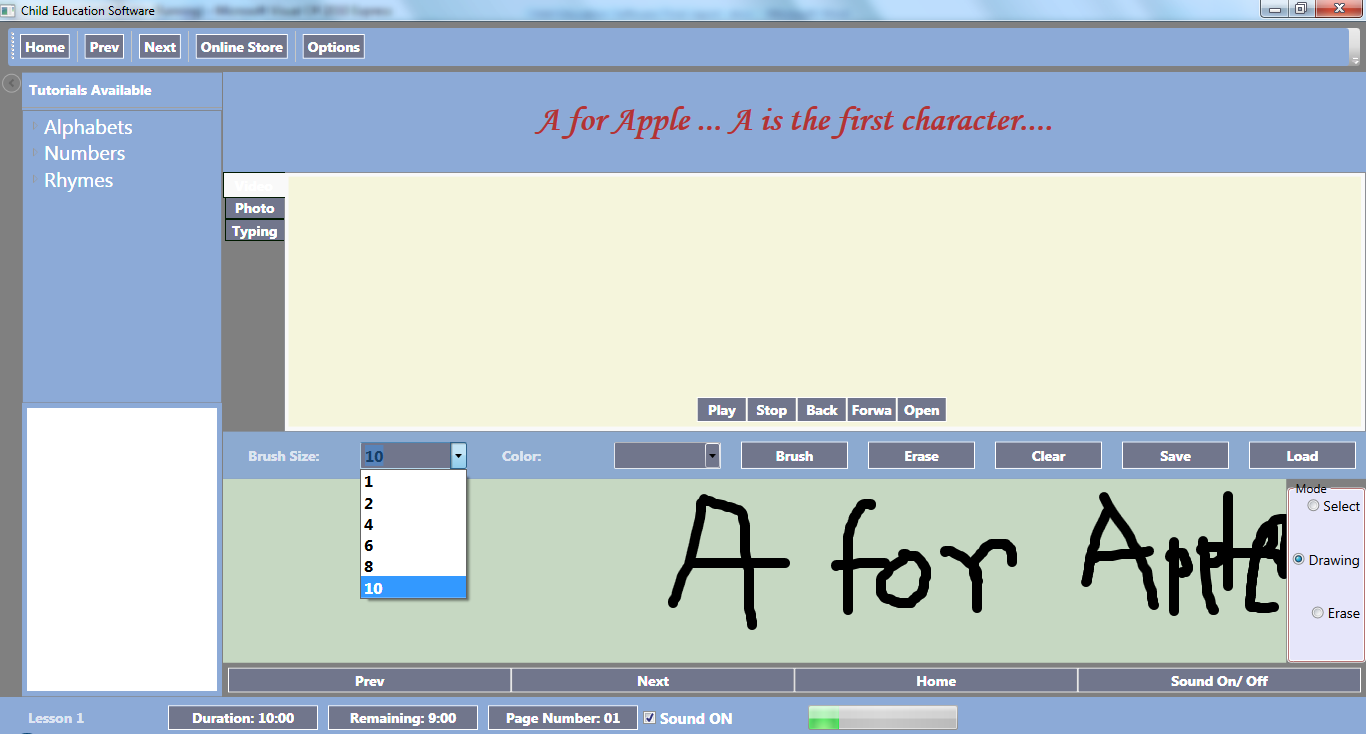
### Photo



### Typing



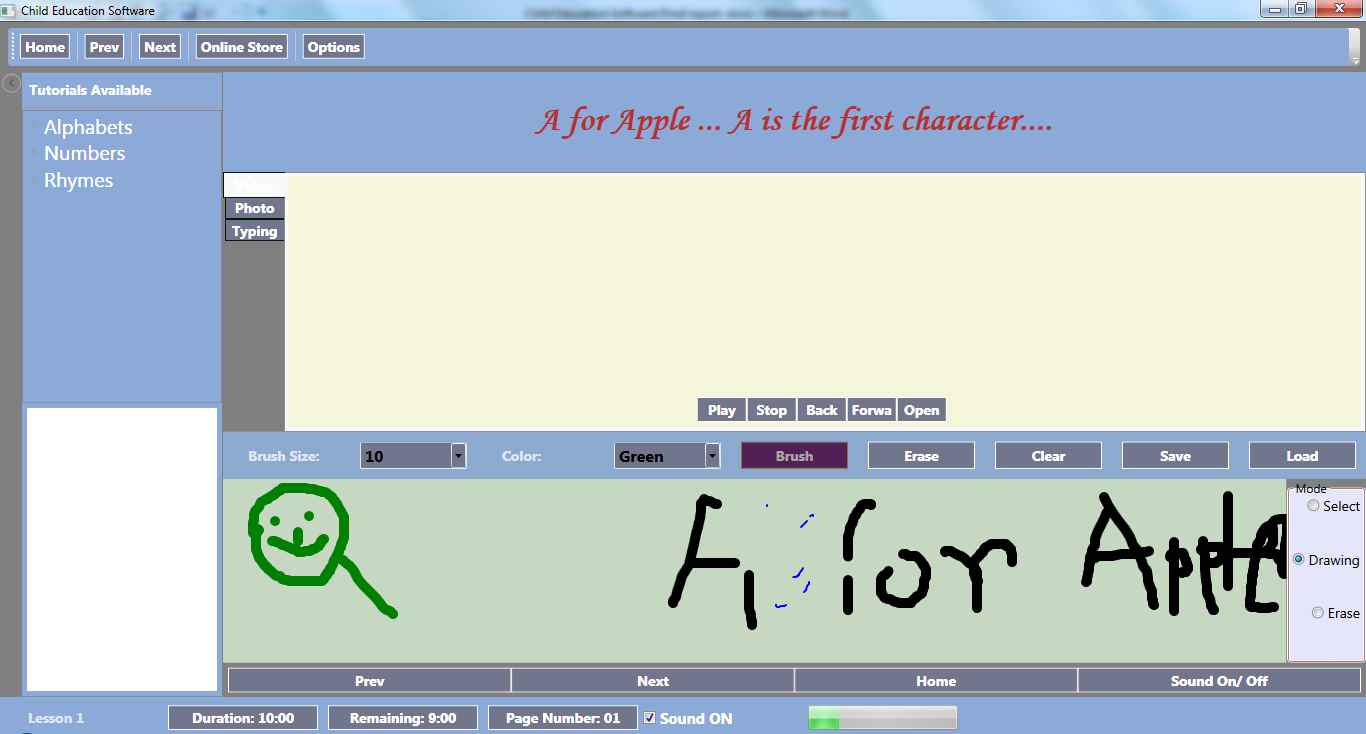
### Brush size



### Select color



### Erase



### Selection



## Test Cases (Unit Test Cases and System Test Cases)

# Coding

## Complete Project Coding

Code

## Comments and Description of Coding segments

Various types of comments and description we use in our coding section. Some of them are:

//open the connection

This comment is use at the data interaction section where the code to open the MySql connection.

//define the command reference

To define a command reference in MySql.

//define the connection used by the command object

To define the connection, which is used by the comment object.

//always close the connection

It is indicating to close connection after code is executed.

Manu Unused code in our project we did comment them also like :  
<!--<Condition Property="Password" Value="c" />-->

## Standardization of the coding

12

## Code Efficiency

We started working on the project keeping in mind that we must develop it in a way that it not only provides a very easy to use GUI but also provide a fast and flexible service to the users. We know that a particular work can be done in more than one ways. We have tried all the options and then chose the one which provides the fastest and most secure performance. First of all, we have used the latest technologies of Microsoft like visual studio 2010 as IDE and WPF as GUI to keep our application’s performance few steps ahead. We have studies all the rules of software development life cycle and applied them to keep our application flexible. We have given special attention to the storage related codes. We have avoided all the unnecessary database codes and kept them as short as possible without harming our purpose so that insertion, updating, deletion and fetching of data take place flexibly. You can see the result as a user; our application does all the works very smoothly.

## Error handling

## Parameters calling/passing

## Validation checks

# Testing

## Testing techniques and Testing strategies used

## Testing Plan used

## Test reports for Unit Test Cases and System Test Cases

## Debugging and Code improvement:

# System Security measures:

## Database/data security:

It encrypts the data stored in the database so that even if someone succeeds to hack the database still not much harm could be done.

The application will use Google open-id authentication for web interface.

## Creation of User profiles and access rights

The software requires a predefined username and password to login.

It allows a guest login as well which lets a guest user user this application with very limited access to the user data.

# Cost Estimation of the Project along with Cost Estimation Model

We used the basic COCOMO model, which gives an approximate estimate of our **CES** project parameters. The basic COCOMO estimation model is given by the following expressions:

Effort = a1 \* (KLOC)a2 PM

Tdev = b1 \* (Effort)b2 months

Where

KLOC is the estimated size of the software product expressed in Kilo Lines of Code a1, a2, b1, b2 are constants for each category of software products.

Tdev is the estimated time to develop the software, expressed in months.

Effort is the total effort required to develop the software product, expressed in person-month (PM).

Our project is semidetached type, because the development team consists of a mixture of experienced and inexperienced staff like my guide and me. Team members may have limited experience on related system but may be unfamiliar with aspects of the system being developed.

## Estimation of development effort

For our Semi-detached class software product **CES**, the formula for estimating the effort based on the code size is shown below:

Semi-detached **CES**: Tdev = 3.0\*(KLOC)1.12 PM

## Estimation of development time

For our Semi-detached class software product **CES**, the formula for estimating the development time based on the effort is given below:

Semi-detached **CES**: Tdev = 2.5\*(Effort)0.35 months

Assume that the size of a Semi-detached CES product has been estimated to be 4,000 lines of source code. Assume that the average salary of software engineer(me) is Rs. 15,000 per month.

Assume that the size of our

The basic COCOMO estimation formula for CES semidetached software:

Our Effort = 3.0 \* (4)1.12 PM

= 14 PM

Normal Development time = 2.5 \* (14)0.35 months

= 6 months

Cost required to develop the product = Rs. 6 \* 15,000

= Rs. 90,000

# Reports

* List of Lesion updates could be downloading.
* Hand writing could be generated.
* A list of events could be generated.
* Typing test could be generated.
* List of google plus update could be generated.

# Future scope and further enhancement of the Project

* Now it will display the text based RSS feeds and link of the multimedia contents. We will display the Multimedia contents like Video, Audio & Image in future.
* To support UNIX / Linux Based Operating systems.
* To Support Mobile Operating systems for Symbian, Meego & Android.

# Bibliography

## Website

* <http://en.wikipedia.org>
* http://en.wikipedia.org/wiki/Windows\_Presentation\_Foundation
* <http://msdn.microsoft.com/en-us/>
* http://www.c-sharpcorner.com/beginners/
* <http://www.microsoft.com/en-us/default.aspx>
* <http://www.codeplex.com/>
* <http://stackoverflow.com/>
* <http://www.codeguru.com/>
* http://www.csharpcourse.com/
* <http://www.w3schools.com>
* <http://blogs.technicise.com/>
* http://connect.technicise.com/
* <http://www.mysql.com/>
* http://www.mysql.com/support/
* <http://dev.mysql.com/>
* http://dev.mysql.com/support/
* <http://www.homeandlearn.co.uk/csharp/csharp.html>
* <http://www.wpftutorial.net/Home.html>
* <http://www.youtube.com/?gl=IN>
* <http://blendinsider.com/>
* https://github.com/
* <https://github.com/anirban-nandy>
* [https://github.com/anirban-nandy/DailyNoteBook](https://github.com/%20anirban-nandy%20/DailyNoteBook)
* <http://learn.github.com/p/intro.html>
* <http://www.vogella.com/articles/Git/article.html>
* <http://try.github.com/levels/1/challenges/1>
* <https://enterprise.github.com/support>
* <https://support.enterprise.github.com/home>

## Books

* Fundamentals of software engineering by Rajib Mall
* Pro C# 2010 and the .NET 4.0 Platform by Andrew Troselen
* C# Programming by Rob Miles

# Appendices

## Mobile App Development

Quickly and efficiently create and test Java™ applications with the Nokia SDKs for Java and the Series 40 platform SDKs.

The Nokia SDKs for Java provide the development tools for phones containing a Java Runtime for Series 40. Each release of the Java Runtime has a corresponding Nokia SDK.

For earlier platform releases Series 40 platform SDKs offer versions to support specific editions and feature packs.

Within both families of tools, each SDK includes Java APIs, an emulator, documentation, code examples, and emulator based debugging tools. The SDKs can be used with either the NetBeans or Eclipse IDEs to create, compile, and package applications and content. Applications can be tested using the emulator.

### Nokia SDK 2.0 for Java — for Series 40 apps

|  |
| --- |
|  |

Create apps for Series 40 phones with the Java Runtime 2.0.0, including the full-touch UI equipped Nokia Asha 305, Nokia Asha 306, and Nokia Asha 311 using the Nokia SDK 2.0 for Java. Then test your apps in an emulator based on the Nokia Asha 305. In addition to the features of the Nokia SDK 1.1 for Java, the 2.0 SDK offers:

#### LWUIT for Series 40 arrives at 1.0

LUWIT for Series 40 has graduated beta to a full productised release. With a number of new APIs — such as PopUpChoiceGroup, ContextMenu, and NokiaListCellRenderer — the 1.0 release includes significant improvements in performance, particularly in lists, themes loading, and the HTMLComponent. Compatibility with the native full-touch UI has been fine-tuned and many bugs fixed, particularly in command handling and text input. A LWUIT Developer's Library has also been released, providing full technical and design guides. There are many new examples too.  
  
This release is delivered to the Nokia SDK 2.0 for Java through the SDK Manager, while a download for the Nokia SDK 1.1 for Java is available from LWUIT for Series 40 project. 

#### The Nokia SDK 2.0 for Java adds new features

The Nokia SDK 2.0 for Java delivers everything you need to develop apps for the exciting new full-touch UI equipped Nokia Asha 305, Nokia Asha 306, and Nokia Asha 311 phones. And now the SDK has graduated from beta.

The SDK delivers the updated Nokia UI API for advanced touch interaction, the Mobile Sensor API (JSR-256) to take advantage of the orientation sensors on the latest Series 40 phones, and the in-app purchase APIs. Now LWUIT is included as a plug-in, so you can create slick UIs, faster. In addition, the emulator gains features for the simulation of multipoint-touch gestures, such as pinch-to-zoom, and PC keyboard input. Based on the Nokia Asha 305, the emulator also provides improved sensor and location support so you can test more of your app on a PC.

The Nokia IDE for Java ME (Eclipse) has also been enhanced with improved searching in the Device SDK Manager and a tool that lets you pull code examples directly into the IDE. Building on the power of the Eclipse platform for Java development, the Nokia IDE makes delivering your Series 40 Java apps easier with features such as an editor for Nokia specific JAD attributes.

#### Testing your Java apps for Series 40 using Remote Device Access

Testing your Java apps on several Series 40 phones is easy and cheap with remote Device Access. Remote Device Access offers a range of Series 40 phones that you access over the internet free-of-charge. So when you need to test your app you can simply pop-on the internet, book a phone, install your app, and you will be testing in minutes.   
  
Right now, the Nokia Asha 311 is available for you to test your apps.

#### Explore in-app purchasing in your Java apps

Using the Nokia SDK 1.1 for Java or later you can explore adding in-app purchase features to your apps, a feature available on phones with Java Runtime 1.1.0 for Series 40 or later. Now you can generate revenue by offering users digital assets and content as part of the app experience — and allow them to purchase these items without leaving your application.

#### Add a new dimension to location with the Maps API

Leveraging the location information provided by the Location API for J2ME™ (JSR-179) on Series 40 phones, you can add rich maps to your apps with the Maps API for Java ME. With exciting features, such as custom overlays, you can create a unique experience. And with the release of the Nokia SDK 2.0 for Java, you now get the Maps APIs delivered ready for your use without additional downloads. Find out more about creating location aware applications with Java technology ›

### Nokia Web - Tools

#### Series 40 web apps tools

##### Nokia Web Tools 2.3

###### **Description**

Nokia Web Tools provides a set of tools that enable the creation of Series 40 web apps. The tools included are:

* Web Developer Environment (WDE) — enabling web apps to be created, edited, packaged and deployed.
* Web App Simulator (WAS) — enabling web apps to be previewed and debugged on a computer.
* Web Developer Channel (WDC) — included in Web Developer Environment, to deliver information and tools to facilitate web app development.

###### **What's new**

This new version of Nokia Web Tools provides:

* UI Designer in WDE offering drag-and-drop population of the web app’s UI.
* the ability to deploy a web app over a USB connection from a PC running Microsoft Windows.
* automatic reloading of the simulator for locally previewed web apps as code changes are saved.
* additional templates upon which to base new web apps, including web apps for trivia games, shopping, and video browsing among others.
* more sample web apps and snippets to help developers use the platform capabilities easily.

###### **Symbian WRT widget development**

This version of Nokia Web Tools no longer supports Symbian WRT widget development. If you wish to continue using [Nokia Web Tools 1.2](http://www.developer.nokia.com/info/sw.nokia.com/id/054b94d5-4cdf-4b17-b268-bedcfc421ba6/Nokia_Web_Tools_1_2_for_Symbian.html) for Symbian WRT widget development, please refer to the Installation Guide for details on the setup requirements.

###### **App publication**

Web apps created with Nokia Web Tools can be submitted for publication in Nokia Store.

###### **Versions available**

Nokia Web Tools are available in versions for:

* 32- or 64-bit Microsoft Windows XP Service Pack 2, Windows Vista, or Windows 7.
* 32-bit Ubuntu Linux 10.04.
* 64-bit Apple Mac OS X 10.6.

To make your Series 40 web apps development as straightforward as possible Nokia Web Tools, Bluetooth Launcher, and the Nokia Xpress Browser are available in the Series 40 web apps section.

Alternatively, Xpress Web App Builder is an online tool that enables content owners to create web apps from clipped content, RSS feeds, and social media content using a wide selection of formats.

You create Series 40 web apps using Nokia Web Tools. Based on Eclipse, Nokia Web Tools builds on the powerful web development features of the Eclipse Web Tools Platform to create the Web Developer Environment (WDE). WDE includes features to create, edit, validate, test, package, and deploy Series 40 web apps. Testing is supported by the Web Apps Simulator (WAS) that enables web apps to be run and tested on a computer. WAS includes an implementation of Web Inspector, so you can perfom debugging and examine of a web app's content and performance. This getting started guide takes you through installing Nokia Web Tools, creating a web app from a template, testing it on your computer, and running it on a phone, before providing links to the resources you need to build great web apps and deliver them to Nokia Store.

WDE offers a number of templates you can use to create Series 40 web apps easily. These templates range from the Basic web app template, which contains the core web app files with no functionality, through a selection of templates offering basic UI constructions to fully functional web apps, such as the Videos browsing project template that offers a working web app to browse videos. You can work with web app examples or a web app project you have already created as well. For more details on importing web apps, see [Importing a web app or web app project](http://www.developer.nokia.com/Resources/Library/Series_40_web_apps_library/#%21tools-library/creating-a-web-app-project/importing-an-existing-web-app-or-web-app-project.html) in the Series 40 Web App Developer's Library.

During development, transferring a web app onto a phone each time you make code changes isn’t a practical way of previewing and testing your web app. To simplify testing of a web app, WDE integrates with the Web App Simulator (WAS) to enable testing on your computer.

You have two options for running your web app in WAS:

* a server (cloud) based preview, this option provides a simulation that is very close to the experience that will be seen on a phone.
* a local preview, which is useful when you are working offline or want to debug your web app.

Having tested your web app in the simulator the next stage is to run it on a Series 40 phone. You have three ways to do this::

* deploying the web app to a phone from WDE over a Bluetooth connection.
* deploying the web app from WDE to a phone over a USB connection (but only if you are working on a Microsoft Windows PC as [Nokia Suite](http://www.nokia.com/nokiasuite) or [Nokia PC Suite](http://www.nokia.com/global/support/nokia-pc-suite/) is required).
* running the web app by entering a short URL into the Nokia Xpress Browser on a phone.

##### Series 40 Web App Developer’s Library

The Series 40 Web App Developer’s Library describes the Series 40 web apps development environment for Series 40 phones that run the Xpress Browser for Series 40, the tools for developing Series 40 web apps, and the design considerations for Series 40 web apps.

##### Nokia design principles

These design principles are common for all Nokia platforms and represent Nokia’s idea of good design. Keep these things in mind to make sure your application will be a success.

###### **Less is more.**

* Keep it simple, easy, and intuitive to use. Make it obvious what your application does and how it works.
* The application is lean. There is no extra content. The application works fast and is powerful.
* The structure is flat. No deep hierarchy and endless scrolling.
* The design is elegant and simple and works without exceptions.

###### **Every pixel counts.**

* Remember that mobile devices have limited screen real estate.
* Consider ergonomics and the size of peoples’ fingers when laying out your application and designing controls. The minimum dimensions for touch areas are 8 mm for thumb, 7 mm for finger, and 5 x 4.5 mm for pen.

###### **Natural interaction.**

* The interaction is familiar, clear, and trustworthy.
* Basic interaction should be achieved with touch.
* Be consistent, logical, and coherent both within the application and within your target platform.
* Make sure that you use terms consistently.

###### **Remember the user.**

* What does the user want to do with the application? Keep the big picture in mind when designing the application.
* Mobile devices are used in different situations. Check that the application can be used in those conditions where it should be used, e.g. in a bus, outdoors, in a noisy environment...
* You are creating the application for the end user, not for yourself.
* The application is intuitive and fun to use.
* The application makes the user smile but not laugh.

###### **Use fonts and colours wisely.**

* The fonts you use are clear and easy to read from small screen.
* Check that the contrast is clear enough.
* Use a limited number of colours.
* Remember the colour metaphors and cultural differences in perceiving colours.

###### **Don’t be offensive.**

* You have only one chance to make the first impression. Make the most of it.
* Check that the application is in line with the [Ovi Store content guidelines](https://publish.ovi.com/account/content_guidelines).

#### Xpress Web App Builder

Xpress Web App Builder is an online tool that guides you through the process of creating rich web apps, with no coding required. Select from a variety of templates, customise your theme, and then add clipped web content, RSS feeds, and social media information. The key features of the tool are:

* layout templates to present content, including single pane, tabbed view, and accordion view, as well as focused templates for news, pictures, and video content.
* a wide range of content widgets for clipped web content; RSS feeds; video from YouTube; pictures from Flickr, Picasa, and other photo sharing sites; and blogs from Tumblr and WordPress.
* the ability to add PSMS and call capabilities, static HERE Maps, and in-app advertising from [Nokia Ad Exchange](http://www.developer.nokia.com/NAX).
* the option to customise your app's colour scheme, including header and font colours.
* static and dynamic previews of your app, for all supported screen resolutions.

When you've completed your web app, the tool provides a short URL for testing the app on your phone, and lets you submit the app to Nokia Publish to start the process of publication in Nokia store. However, if you want to customise your web app further, you can download the source code and import it into Nokia Web Tools.

#### Test your Series 40 web apps

If you don't have access to a Series 40 phone, you can test your web content and apps by making use of the Remote Device Access service. This service provides you with access to ten Series 40 phone models, more than 30 phones, over an internet connection. The service is available free to all Nokia Developer members.

#### Mobile Web Components

Make the most of the latest HTML5 feature in Nokia browsers.

Add rich, HTML5 based components to your web pages and web apps for Symbian Anna phones and the Nokia N9 Smartphone. Included are components for collapsible content blocks, scrollable large content item windows, pop-up menus, expandable sliding menus, slideshows, and others.

#### Leverage the power of QtWebKit

Using Qt WebKit technology, Web developer can easily transform web apps and web services into powerful native applications. Qt offers HTML5 and CSS3 support today. The quick and powerful way to use web assets and skills to produce apps for smartphones and mobile computers.

#### Device APIs::API Bridge

APIBridge is a component for Nokia Symbian devices that enables WRT widgets, Adobe Flash Lite content, and Java applications to access device features through a plug-in architecture. The APIBridge package ships with a set of plug-ins and the components to enable the features of the plug-ins to be used. Developers can extend the APIBridge component with their own plug-ins.

## Cacoo:: online drawing tool

 Cacoo is a diagram creation tool that runs in your web browser.Multiple people can work together on the same diagram in real time.Diagrams can be published directly to websites, wikis, and blogs.

### Creating Diagrams

* Elements can be dragged and drop to easily create diagrams.
* Elements can be linked together with connectors.
* Connectors automatically move when elements are repositioned.
* You can use a text box and put text anywhere you like.
* You can upload images from your PC and include them in Diagrams.
* You can take screenshots of your computer from within Cacoo.
* Smart styles can easily be applied to stencils.
* You can have multiple sheets in a diagram and use them as backgrounds or layers.
* When you move the objects on your canvas, they will be snapped at the objects or grids nearby and align automatically.
* Copying, pasting and other functionality of basic drawing software is also built in to Cacoo.
* All actions are stored so there are unlimited levels of undo.
* You can import an image from the other websites by indicating the URL.
* The imported image can be easily trimmed only using your mouse.
* According to your editing status, tips will be shown on the right bottom corner of the canvas.

### Collaboration

* You can invite collaborators to work with you in Cacoo.
* Multiple people can edit a diagram in real time.
* There is a chat function in the editor so people can communicate while creating diagrams.
* People can leave comments about the diagrams.
* Each user can set their own user icon.
* When editing with multiple people, users icons appear on selected objects.
* Sharing diagrams become much smoother. Diagrams in the shared folders can be accessible and editable by people who you have shared the folder with.

### Sharing Diagrams

* If you keep the diagram private then other users can't see it.
* If you make the diagram URL public, then anyone who knows the URL can see it.
* Publishing a diagram to a blog can be useful in various ways.
* You can place code into blogs to create a slideshow
* Published images always display the most recent version.
* Diagrams can be exported to SVG format (Plus Plan users only) and PNG format. (More formats will be available in the future.)
* Diagrams can be posted to Twitter/Facebook/GoogleBuzz
* Diagrams can be displayed in SVG format for printing. (Plus Plan users only. A few browsers are not supported.)

### Managing Diagrams

* Diagrams can be placed into folders.
* Diagrams can be copied.
* Diagrams can be displayed as thumbnails or as a list.

### Languages and Time Zones

* All pages and notification e-mails support English and Japanese
* Users can enter text from almost all languages.
* Dates are displayed relative to your local time zone.

### Security

* Private diagrams can only be seen by users you select.
* URLs which you do not share can not be found by other users or search engines.
* All editing and management is protected by SSL.
* In order to access information about diagrams a Cacoo ID and password are requited.
* User passwords are encrypted on Cacoo's server.

### API

* You can access Cacoo using the API.
* The Cacoo API supports OAuth and an API Key.

By using the Cacoo API you are able to interact with Cacoo from other services and applications.

Authorization Methods

There are two ways to access the Cacoo API.

#### 1. API Key

The API key allows you make requests to the Cacoo API. You can make an API key here.

#### API Key

Append your API key to requests to the API to return data from your account.(Parameter name "apiKey")

Example: https://cacoo.com/api/v1/diagrams.json?apiKey=abcdefghijklmn

#### 2. OAuth

OAuth 1.0a is supported as an authorization method for Cacoo. You can register applications here.

You can get your Access Token from the following links.

#### applications

Access Token:https://cacoo.com/oauth/access\_token

Authorize:https://cacoo.com/oauth/authorize

Request Token:https://cacoo.com/oauth/request\_token

## http://t1.gstatic.com/images?q=tbn:ANd9GcS-CmbHGLD4MH83JH1oNIr_acREqblVhrcFuvQfYZR8HFi1UpaqlgGitHub

GitHub is a web-based hosting service for software development projects that use the Git revision control system. GitHub offers both paid plans for private repositories, and free accounts for open source projects. As of May 2011, GitHub was the most popular open source code repository site.GitHub Inc. was founded in 2008 and is based in San Francisco, California.

### Description

The site provides social networking functionality such as feeds, followers and the network graph to display how developers work on their versions of a repository.

GitHub also operates other services: a pastebin-style site called Gist that provides wikis for individual repositories and web pages that can be edited through a Git repository, a slide hosting service called Speaker Deck, and a web analytics platform called Gauges.

As of January 2010, GitHub is operated under the name GitHub, Inc.

The software that runs GitHub was written using Ruby on Rails and Erlang by GitHub, Inc. (previously known as Logical Awesome) developers Chris Wanstrath, PJ Hyett, and Tom Preston-Werner.

### Limitations and constraints

According to the terms of service,if an account's bandwidth usage significantly exceeds the average of other GitHub customers, the account's file hosting service may be immediately disabled or throttled until bandwidth consumption is reduced. In addition, while there is no hard limit, the guideline for the maximum size of a repository is one gigabyte.

# Glossary.