Project Requirements

Written By: Team 38

Team Members: *Somay Jain ,Sriharsha Vogeti, Suggu Sai Praneeth,Arnav Sharma.*

Brief problem statement

Mathematics is an important subject in today's educational system. Hence it is very necessary to create interest in the minds of children at a tender age towards the subject. This can be done by making learning more interesting with the help of technologies. The application teaches regular arithmetic(multiplication, division) such as 7X8 in a really interactive manner.

System requirements

- The system must be implemented using web technologies like HTML5, CSS and Javascript for building the web application.
- The system is also implemented using the open-source framework Phonegap to create mobile applications for Android and iOS.
- The system must be capable of running on almost all browsers such as Firefox, Chrome, Safari, Opera, on all portable devices having Android version 2.2 and above and on iOS version 4 and above.

Users profile

- Children aged 12 or below.
- People eager to learn a new simple, efficient way of doing arithmetic.
- Basic computer skills
  - Can manipulate a mouse.
  - Can click and drag with a mouse.
  - Can respond appropriately to dialogs.
  - Can open a website
- Basic Mobile Handling skills
  - Can install mobile application
  - Can open and use a mobile application

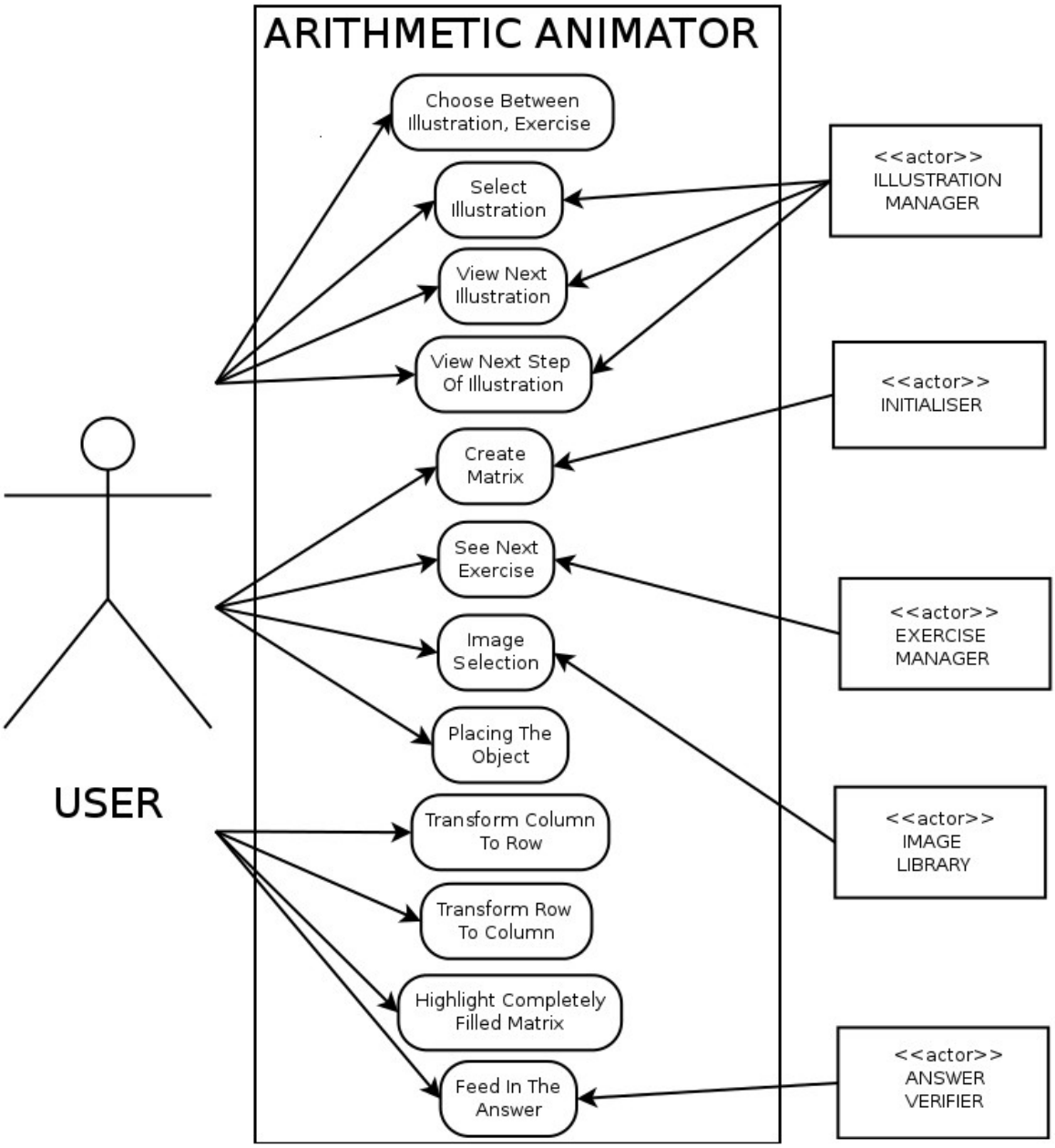
Use case description

No	Use Case Name	Description	Release
1	Choose between <Illustration> and <Exercise>	Our application presents user with two options i.e. Illustration and Exercise.	R1
2	Select Illustration.	The User has to choose between	R1

		Illustrations.	
3	View next illustration.	After he finishes one Illustration he will have the option to view other illustration.	R1
4	View the next step of illustration.	After the user is done with what the current step of illustration teaches, this step takes him to the next part.	R2
5	Create matrix.	When the user enters into the exercise initially an empty matrix of square boxes is created along with column number and row number displayed at the top and is returned to the output screen which is used further.	R2
6	See next exercise.	He will perform the whole process by himself. The question will be selected by him in list displayed by the exercise manager.	R2
7	Image Selection	After the user selects an integer he has to choose with what images he wants to fill the boxes with.	R1
8	Placing the object.	When the user comes to this screen he will be provided an empty matrix and he has to fill them with the image he has selected.	R2
9	Transform column to row.	If the best possible configuration requires a column to be transformed into a row, then this step is performed.	R1
10.	Transform row to column.	If the best possible configuration requires a row to be transformed into a row, then this step is performed.	R1
11	Highlight completely filled matrix.	He will have to highlight the completely filled matrix to calculate the number of tens. In illustration the matrix will be highlighted but in exercise the user has to do it.	R1

12	Feed in the answer.	He should highlight the completely filled matrix so that he can calculate the number of tens and the remaining matrix to count the change.	R2
13	Start Up Page	This Page is displayed as the User Open the application.	R1

Use Case Diagram



Use Case Description

Use Case Number:	UC-01
Use Case Name:	Select between <Illustration> and <Exercise>
Overview:	Our application presents user with two options i.e. Illustration and Exercise.
Actors:	User
Pre condition:	The must have been through the start up page.
Flow:	Main (success) Flow: 1.Illustrattion and Exercise option are displayed
Post Condition:	Player has to choose between Illustration and Exercise and based on his selection he is moved to a screen listing various illustrations and exercises.

Use Case Number:	UC-02
Use Case Name:	Select one <illustration> from <illustrations>
Overview:	The User has to choose between Illustrations.
Actors:	User, Illustration Manager
Pre condition:	He has to choose between Illustration and Example
Flow:	Main (success) Flow: 1.A list of activated (as the user completes more illustrations, more illustrations will be activated) illustrations will displayed.
Post Condition:	The Illustration Manager takes the user to the selected illustration.

Use Case Number:	UC-03
Use Case Name:	View the next illustration.
Overview	After he finishes one Illustration he will have the option to view other illustration.
Actors:	User, Illustration Manager.
Pre condi-	He has to choose the select the Illustration when the Illustration Manager

tion:	presents the option.
Flow:	Main (success) Flow: 1.He selects an Illustration. 2. The first part of the Illustration is displayed.
	Alternative Flow : He tries to select an inactivated illustration.
Post Condition:	After this illustration he can choose the next step of illustration till the illustration is exhausted.

Use Case Number:	UC-04
Use Case Name:	Choose to view the next step of illustration.
Overview	After the user is done with what the current step of illustration teaches, this step takes him to the next part.
Actors:	User, Illustration Manager
Pre condition:	He has to choose an Illustration.
Flow:	Main (success) Flow: 1.He chooses Illustration option 2. He successfully completes the first part of the illustration.
	Alternate (failure) Flow: 1. He does not complete the first step of the illustration.
Post Condition:	After finishing one illustration he will be presented with the option to do other activated illustrations.

Use Case Number:	UC-05
Use Case Name:	Create Matrix.
Overview:	When the user enters into the exercise initially an empty matrix of square boxes is created along with column number and row number displayed at the top and is returned to the output screen which is used further.
Actors:	Initializer,User

Pre condition:	Integer that can have a value of either 5,10,15 or 20.
Flow:	Main (success) Flow: 1. The system asks the user for either Exercise or Illustration. 2. Exercise is selected. 3. System asks for a number (N) from the user. 4. NxN matrix is made from it.
Post Condition:	User who selected the integer will next choose an image.

Use Case Number:	UC-06
Use Case Name:	See next exercise.
Overview:	He will perform the whole process by himself. The question will be selected by him in list displayed by the exercise manager.
Actors:	User, Exercise Manager.
Pre condition:	He will have to watch the illustration of the same level before doing this.
Flow:	Main (success) Flow: 1. Select the exercise in the exercise list displayed by the exercise manager. 2. The user performs the exercise by himself from what he has learned in the illustration.
	Alternate Flow after step 1: If he does not follow the correct order of steps, he has to redo his actions.
Post Condition:	The answer is sent to the answer verifier for verification.

Use Case Number:	UC-07
Use Case Name:	Image Selection
Overview:	After the user selects an integer he has to choose with what images he wants to fill the boxes with.
Actors:	User, Image Library
Pre condition:	User has to select the image from the Image Library.

Flow:	Main (success) Flow: 1.User has to select the image from the various options available in the Image Library. 2.He has to press the submit button to go to the next screen.
Post Condition:	After this the player will be directed to the next screen where the matrix has to be filled with the selected object.

Use Case Number:	UC-08
Use Case Name:	Placing the object
Overview:	When the user comes to this screen he will be provided an empty matrix and he has to fill them with the image he has selected.
Actors:	User, Exercise Manager
Pre condition:	After the user has selected the input number and chosen the image he will be directed to this screen.
Flow:	Main (success) Flow: 1.The user has to fill the boxes in a specified order 2. First he has to fill a row and then subsequently the next rows. 3. He can also select some area by dragging and the area in the matrix will be filled with the selection by the Exercise Manager.
	Alternate Flow after step 2: 2.1. If the user does not fill in the matrix in correct order and error is displayed and the user is requested to redo his actions.
Post Condition:	The user then has to transform rows and columns to get the most suitable configuration.

Use Case Number:	UC-09
Use Case Name:	Transform Column to Row.
Overview:	If the best possible configuration requires a column to be transformed into a row, then this step is performed.
Actors:	User
Pre condition:	He should fill the matrix with the object selected.
Flow:	Main (success) Flow:

	1. He selects the row he wants to transform and moves it right next to a column.
	Alternate Flow after step 1: If he does it wrong and he will be asked to redo it.
Post Condition:	1. On optimal configuration is achieved. 2. The user then has to highlight in tens to count.

Use Case Number:	UC-10
Use Case Name:	Transform Row to Column.
Overview:	If the best possible configuration requires a row to be transformed into a row, then this step is performed.
Actors:	User
Pre condition:	He should fill the matrix with the object selected.
Flow:	Main (success) Flow: 1. He selects the column he wants to transform and moves it right next to a row.
	Alternate Flow after step 1: If he does it wrong and he will be asked to redo it.
Post Condition:	1. On optimal configuration is achieved. 2. The user then has to highlight in tens to count.

Use Case Number:	UC-11
Use Case Name:	Highlight completely filled matrix.
Overview:	He will have to highlight the completely filled matrix to calculate the number of tens.
Actors:	User.
Pre condition:	He should first fill the matrix.
Flow:	Main (success) Flow: He will highlight the completely filled matrix.
	Alternate Flow after step 2: 2.1. If he fails to do so an error will be displayed and he will have to redo it.
Post Condi-	After this he has to calculate the final answer.

tion:	

Use Case Number:	UC-12
Use Case Name:	Feed in the answer
Overview:	Here the user will have to calculate the final answer.
Actors:	User, Answer Verifier, Exercise Manager
Pre condition:	He should highlight the completely filled matrix so that he can calculate the number of tens and the remaining matrix to count the change.
Flow:	Main (success) Flow: 1. He has to calculate the final answer based on completely filled matrix and the left over matrix.
	Alternate Flow after step 1: If he enters wrong answer error will be displayed and he will be asked to redo it.
Post Condition:	3. Some exercises will be activated 4. The user will be given an option to chose a new (next) exercise by the exercise manager.

Use Case Number:	UC - 13
Use Case Name:	Start up Page
Overview:	This is the first page and displays the name of the project and then redirects to the actual app.
Actors:	*User*
Pre condition:	This the first display Requires no pre condition.
Flow:	After this the user will be redirected to the app.
Post Condition:	After this he will be provided an option to choose between