

Software Engineering Project Report

Submitted by panduranga (121003036) and vikas reddy (121003130)

Date of submission - 4/20/2020

CONTRIBUTIONS :

Panduranga - UI, authentication, database connection, animations, empirical estimation, methodology, modules and its functionalities, test cases, automated testing , aim and objectives, limitations, competitive software.

vikas reddy - UML diagrams, data flow diagrams, shuffling functions, math functions, question generator, random sequence generator functions in implementation, unit testing, usability testing, functional testing, database testing.

Contents

- a. Problem statement
- b. Aim and objectives
- c. Existing/competitive software
- d. Limitations of existing software
- e. Methodology –modules and their functionalities
- f. Design Engineering
 - UML diagrams
 - Data flow diagrams
- g. Development
- h. GUI and Back end screen shots
 - Desktop view
 - Mobile view
 - Back end
- i. Testing
 - Test cases
 - Automated test results
- j. Empirical estimation
 - LOC
 - FP

a. Problem Statement

In this rapidly growing world, computer became an essential and learning computer basics at young age is a must to catch up. So pre-school management and Microsoft as stakeholders, joined to develop an e-learning application for kids that introduce them to computer basics.

b. Aim and Objectives

Developing an e-learning application that helps children to learn computer basics along with fun activities like coloring, quizzes and story reading modules is the aim of this project.

The Objectives of this e-learning website

- Interaction and learning modules should be clear and simple.
- Parents should be able to track the progress if they want to.
- To make all the basic learning modules available to children without requiring any price to pay.
- Introduce all the different ways of interaction with a computer to kids

c. Existing/competitive software

- **e-learningforkids.org**

e-learning for kids is an educational site with free access to science, math, computer, language arts, and life skills

- **studentsuk.com**

Stories for Kids is a website dedicated to providing free and available stories for children, young adults and teenagers

- **funbrain.com**

FunBrain is an educational browser game website for children and adults. Funbrain offers hundreds of games, books, comics, and videos that develop skills in math, reading, problem-solving and literacy.

d. Limitations of existing solutions

- **Interaction types**

Existing software that are mentioned above, primarily focus on learning alone rather than introducing computer to kids. So their interaction includes clicking alone whereas our software include all possible kind of interactions such as click, scroll, drag, drop, slide, zoom in and zoom out along with basic learning modules that are essential for kids of age 3 – 7.

- **Limited no of questions**

All the existing software have very limited no of questions mostly 5 -10 per game. So we focus on generating huge no of questions which are random, sensible and correct. Odd man out module has a capacity to generate *60 lakh different questions* & both Identify pic , Identify word module can generate *10 lakh different questions* and there's no limit in math module for no of questions.

- **Focus on single type of learning**

There are separate software for coloring, reading story books, quizzes, math but story modules, coloring modules, math modules, quizzes, numbers, alphabets are not available in one software alone which will be extra work for parents and students to search for different applications and switch between them.

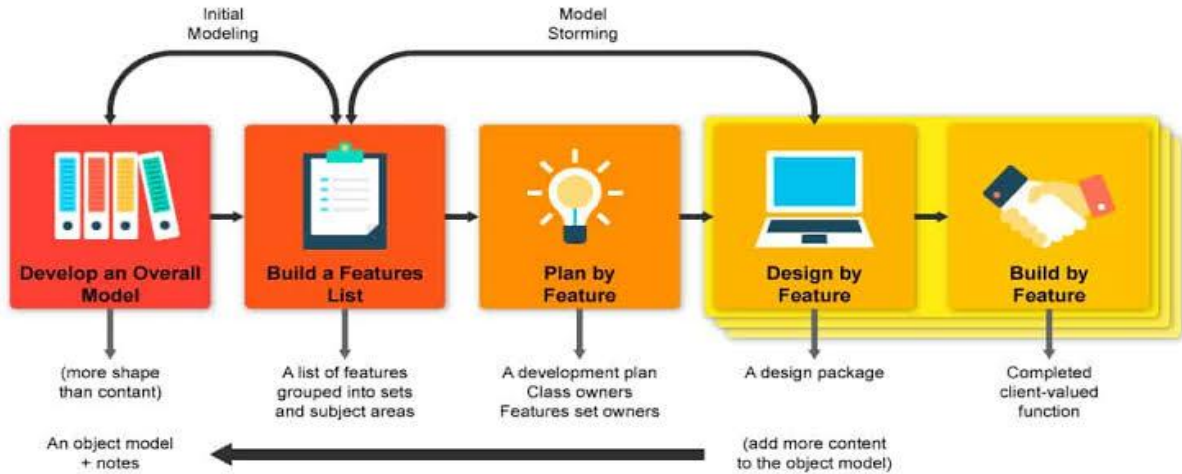
- **Some software come with cost**

Some Existing e-learning applications require the parents to pay for their preschool kids, which is treated as potential reason for children not showing up on these websites. So we focus to develop simple and free software for kids to have fun.

e. Methodology – modules and their functionalities

Feature driven process model is chosen to develop this software as new features can be added on the stake holders request easily. The general objective of FDD is to deliver concrete and flexible software in a short time. FDD or feature-driven development is an Agile framework - a certain process that offers businesses feature-rich systems that support them in controlling their ever-growing nature. Even from its name, we may immediately guess that this framework organizes

software development around making progress on features.



FDD has following steps

1. Developing an Overall Model
2. Building a Feature List
3. Planning by the Feature
4. Designing by the Feature
5. Building by the Feature

Advantages of FDD

1. Minimal Complexity of the System
2. Maximum Quality
3. Useful when team members experience varies
4. Well defined progress tracking

Disadvantages of FDD

1. Costly for steady developing environment
2. Assume that changes will be made in future

Modules and their functionalities

Modules	Functionalities
Coloring	<ul style="list-style-type: none">• Coloring by picking colors from<ul style="list-style-type: none">◦ color palette◦ RGB sliders• Zoom in and out for accuracy
Stories	<ul style="list-style-type: none">• Light & dark theme• Change font size• Change font style
Numbers	<ul style="list-style-type: none">• Generate random sequence of numbers to display• Arrange numbers by drag and drop• Result score
Alphabets	<ul style="list-style-type: none">• Generate random sequence of numbers to display• Arrange alphabets by drag and drop• Result score
Odd man out	<ul style="list-style-type: none">• Fetch questions and answers randomly and correctly• Options display animations• Validate• Update and calculate scores and high scores
Identify pic	<ul style="list-style-type: none">• Fetch questions and answers randomly and correctly• Options display animations• Validate• Update and calculate scores and high scores
Identify words	<ul style="list-style-type: none">• Fetch questions and answers randomly and correctly• Options display animations

	<ul style="list-style-type: none"> • Validate • Update and calculate scores and high scores
Math	<ul style="list-style-type: none"> • Operation selection • Random question generation according to score (easy -> hard) • Validate • Update & calculate hi score and score • Game over display with answers
Progress	<ul style="list-style-type: none"> • Name, email, profile pic • All high scores
Authentication	<ul style="list-style-type: none"> • Login • Signup • Errors if any during authentication

f. Design Engineering

1. Use case diagram

Identifying actors

Actor Id	Actor Name	Description
A1	Student	Actual user of the application. He interacts with the system to play games
A2	Auth manager	Authenticates users with email id and password
A3	Admin	Updates games data, stories and drawings
A4	Data base Admin	Stores and gives game data like high scores

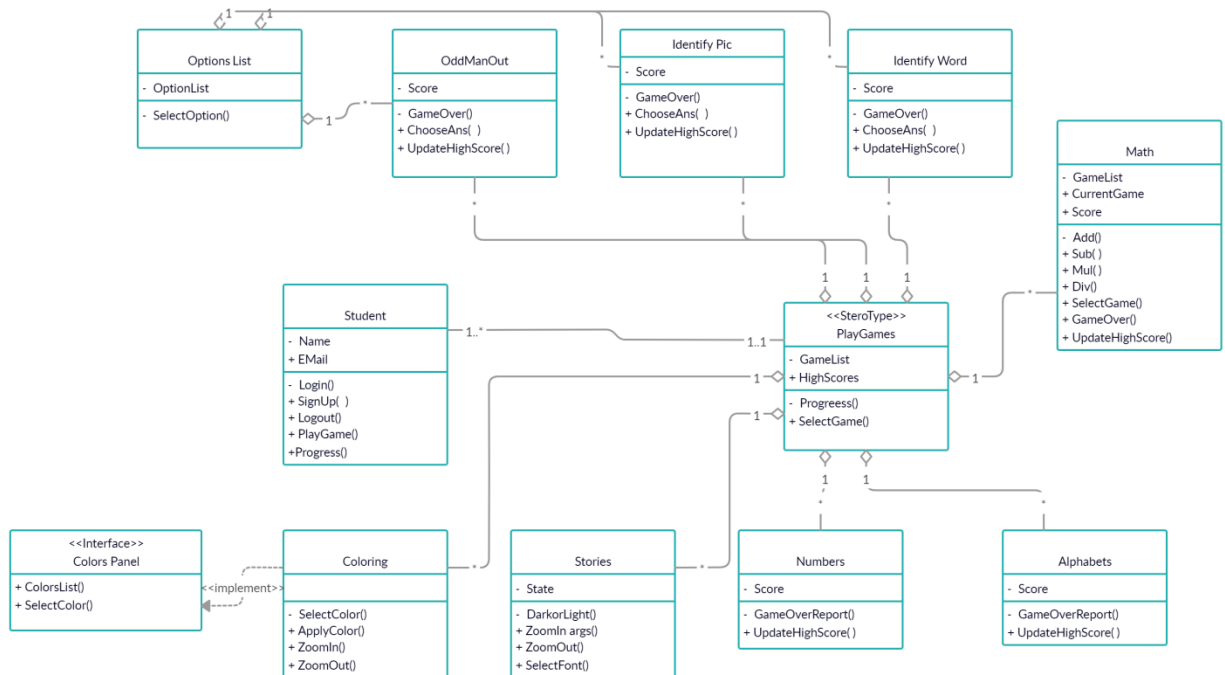


Identifying Use cases

Use case ID	Name	Description
UC1	Signup	New Users signup with email, name and password
UC2	Login	Registered user logs in with their credentials
UC3	Logout	Users logout to end session
UC4	Update data	Admin updates data required for games
UC5	Track progress	Progress is monitored and updated
UC6	Play games	Includes coloring, math, stories, Odd man out, numbers, alphabets

2. Class Diagram

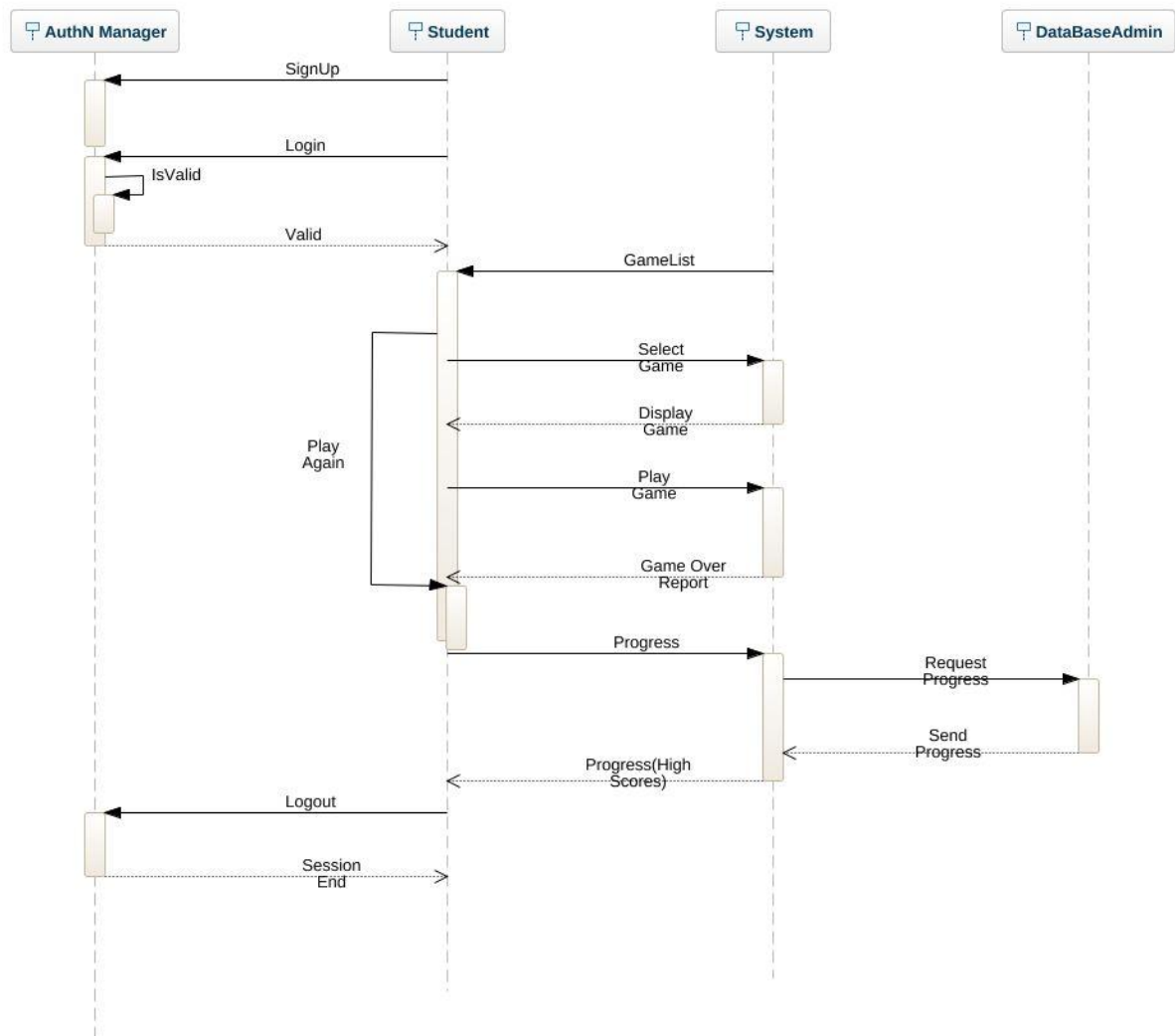
Class	Class	Association
Student	Play games	One-many association
Play games	coloring	Inherit
Play games	Stories	Inherit
Play games	Odd man out	Inherit
Play games	Identify pic	Inherit
Play games	Identify word	Inherit
Play games	Numbers	Inherit
Play games	Alphabets	Inherit
Play games	Math	Inherit
Odd man out	Options panel	Implements
Identify pic	Options panel	Implements
Identify word	Options panel	implements



3 .Sequence diagrams

This logical sequence diagram contains the interactions (messages)among the actors.

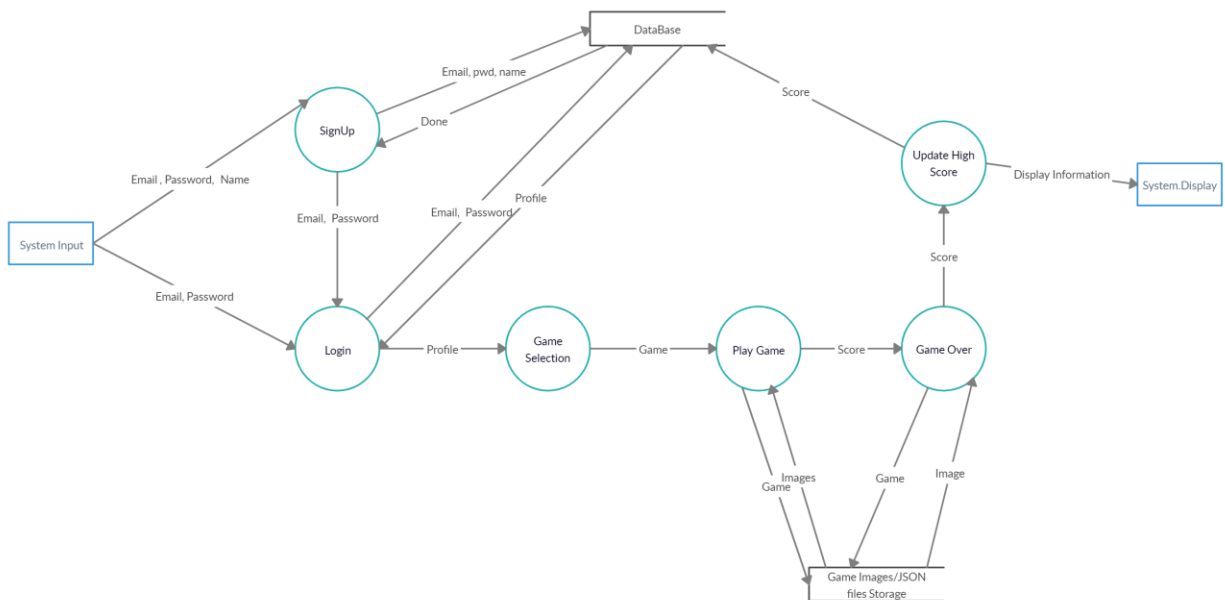
Student first sign up with email, name and password .Then by using those credentials user can login select any game or view his/her progress in the profile tab or log out.



4. Dataflow diagram

The shown DFD is a level-1 DFD which shows all the higher level processes.

It shows the flow of data that happens in the application. A process processes the input data and gives output data. It shows the data repositories which are the actual storages of the data.



g .Development

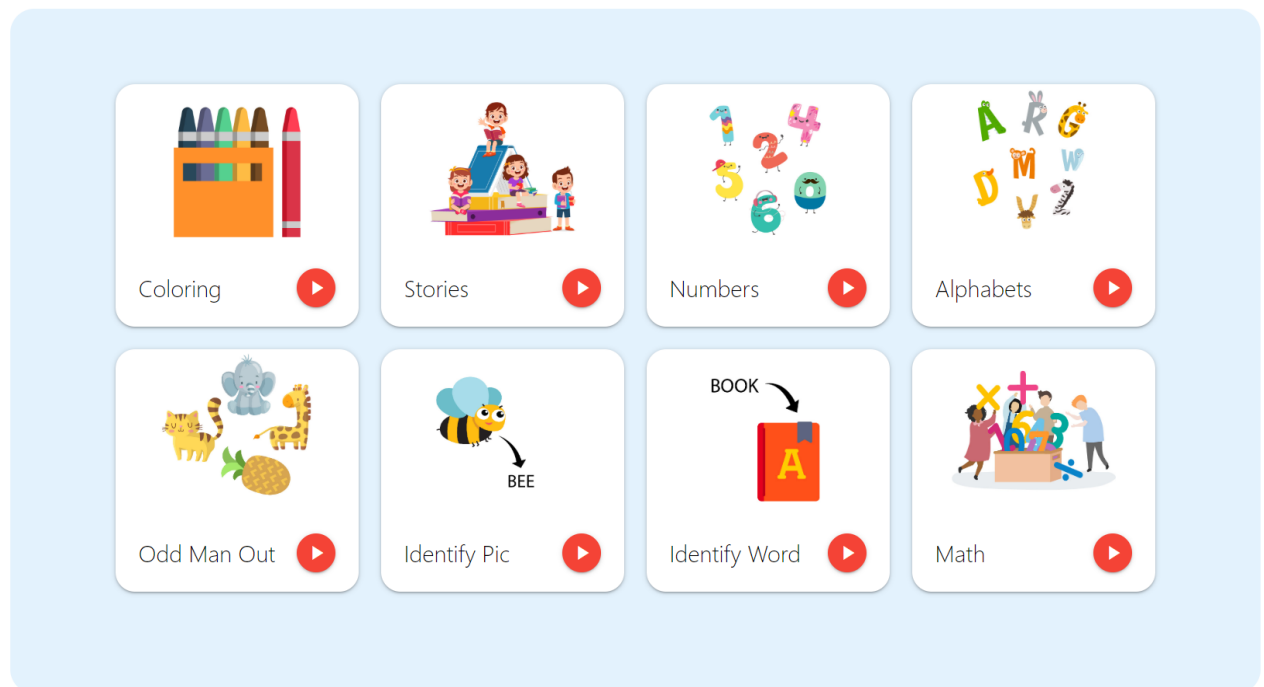
The project's code is in the following github repository.

<https://github.com/1panduranga/Scribble>

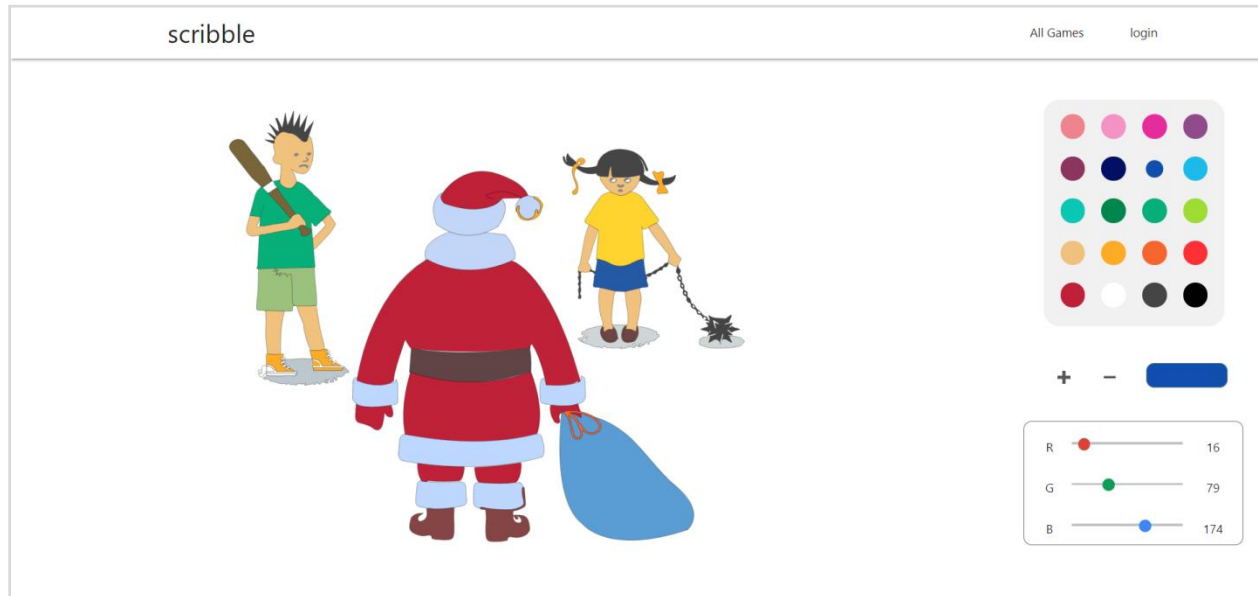
h.GUI and Backend Screen shots

1. Desktop view

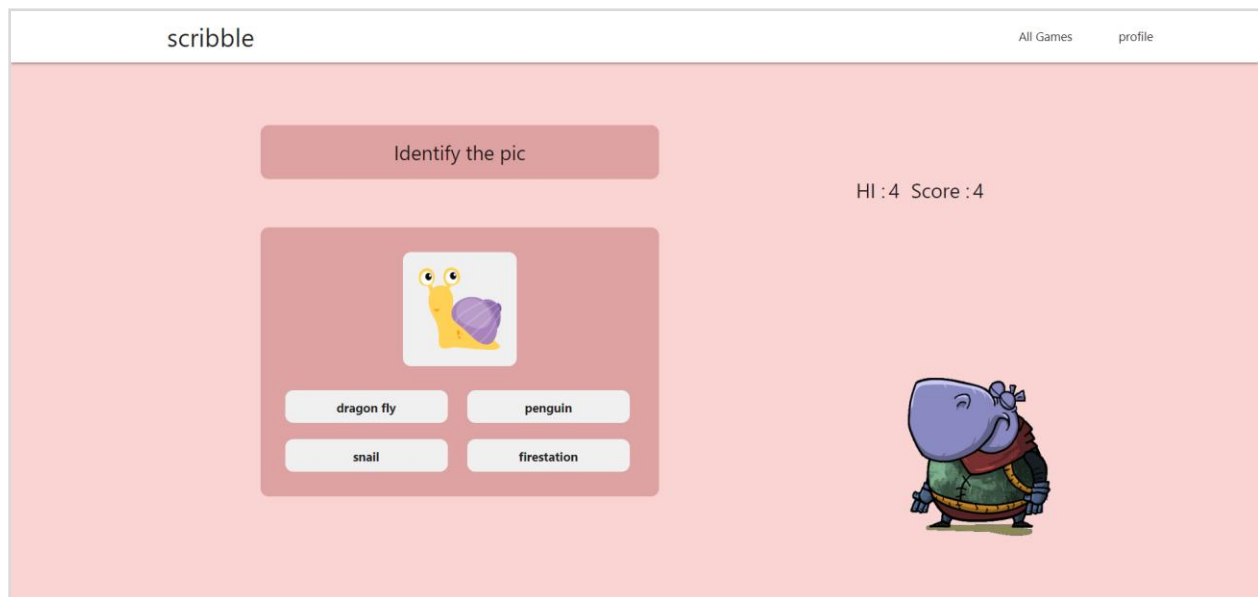
All modules on homepage



Coloring module with color palette and rgb sliders



Identify pic module



Identify word module


scribble


All Games


profile


Identify the word

hand










HI : 13

Score : 9




Odd man out module


scribble


All Games


profile

find the odd one










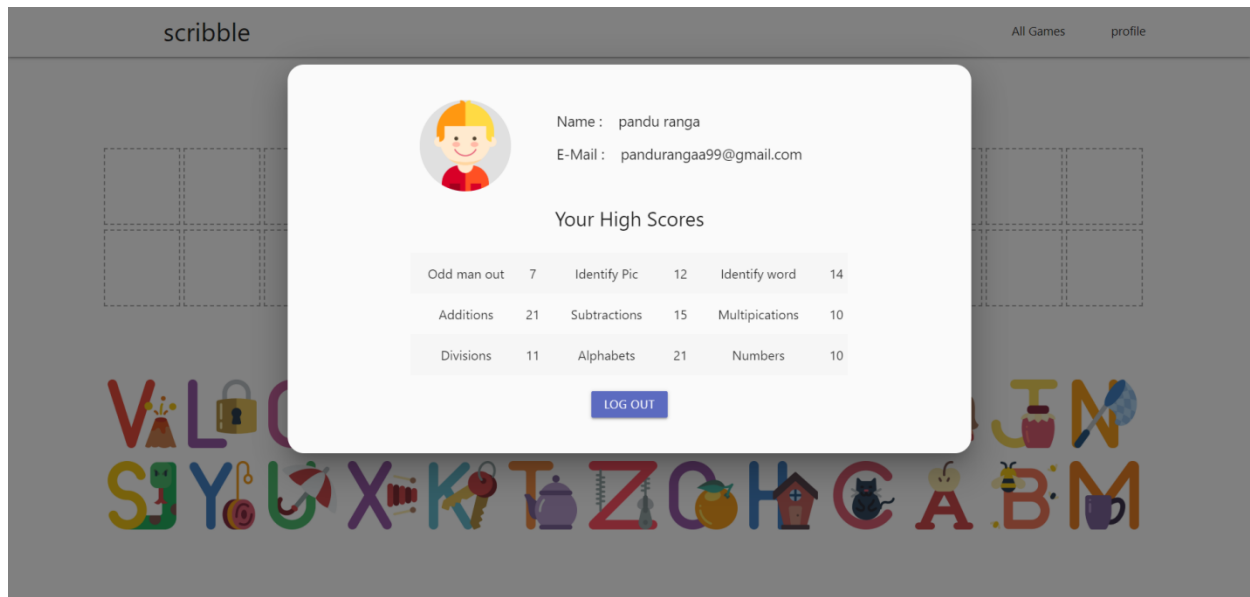
HI : 10

Score : 1

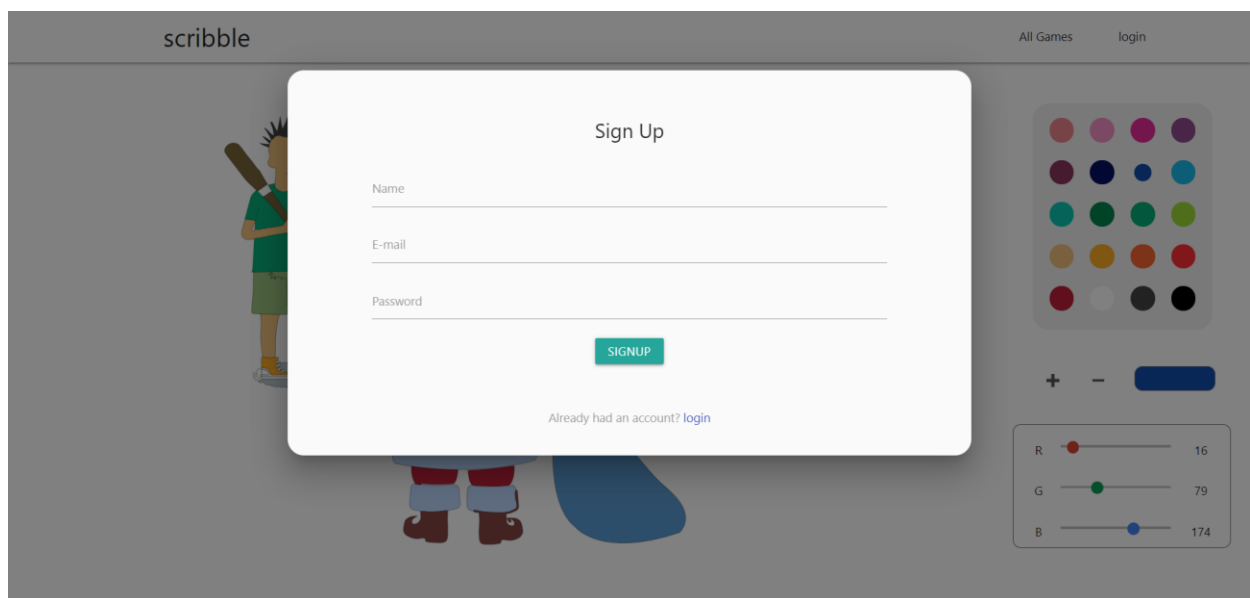
WOAH !!
It's correct answer






Profile module with account details, high scores and log out option




Signup and login module



Stories module in light mode

Little Red Riding Hood





By William son


10 mins read

IN a great wide forest, full of beautiful trees, and green glades, and thorny thickets, there lived a long time ago a wood-cutter and his wife, who had only one child, a little girl. She was so pretty, and so good, that the sun seemed to shine more brightly when its light fell upon her rosy little face, and the birds would seem to sing more sweetly when she was passing by.

Her real name was Maisie; but the neighbors round about all called her "Little Red Riding-Hood," because of a scarlet riding-hood and cloak that her kind old grandmother had made for her, and which she nearly always wore. She was a happy, merry little child, with a smile and a gentle word for everybody, and so you may easily believe that everybody loved her, and was glad to catch a glimpse of her golden curls and her scarlet cloak as she tripped

Stories module in dark mode



Little Red Riding Hood

By William son

10 mins read

IN a great wide forest, full of beautiful trees, and green glades, and thorny thickets, there lived a long time ago a wood-cutter and his wife, who had only one child, a little girl. She was so pretty, and so good, that the sun seemed to shine more brightly when its light fell upon her rosy little face, and the birds would seem to sing more sweetly when she was passing by.

Her real name was Maisie; but the neighbors round about all called her "Little Red Riding-Hood," because of a scarlet riding-hood and cloak that her kind old grandmother had made for her, and which she nearly always wore. She was a happy, merry little child, with a smile and a gentle word for everybody, and so you may easily believe that everybody loved her, and was glad to catch a glimpse of her golden curls and her scarlet cloak as she tripped

Alphabets module

scribble

All Games

profile

Arrange the Alphabets



Numbers module

scribble

All Games

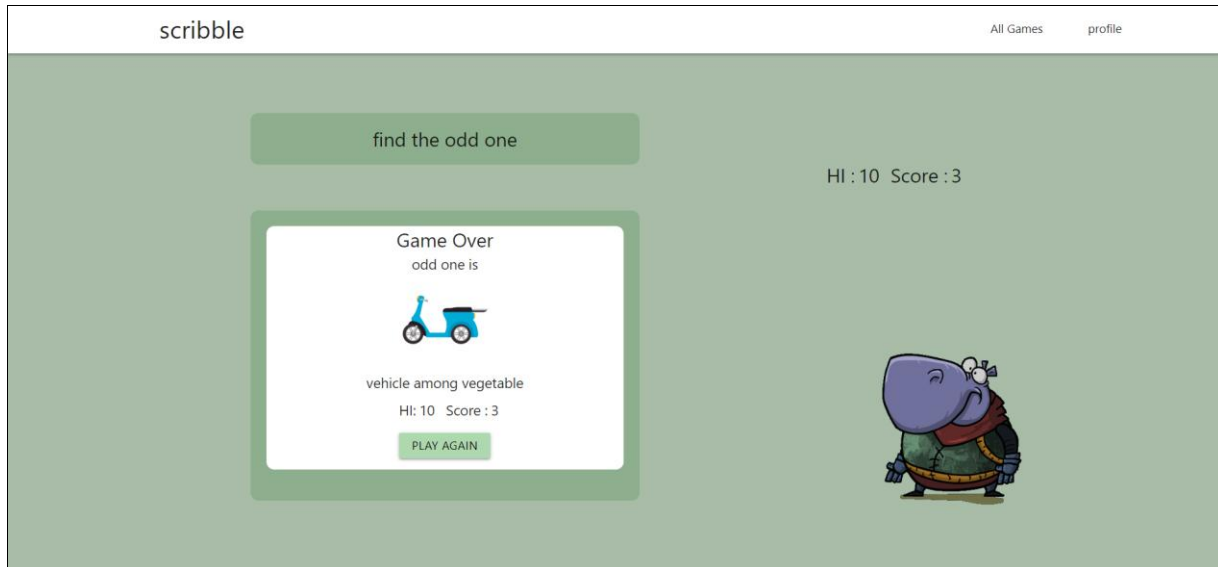
profile

Arrange the numbers in sequence

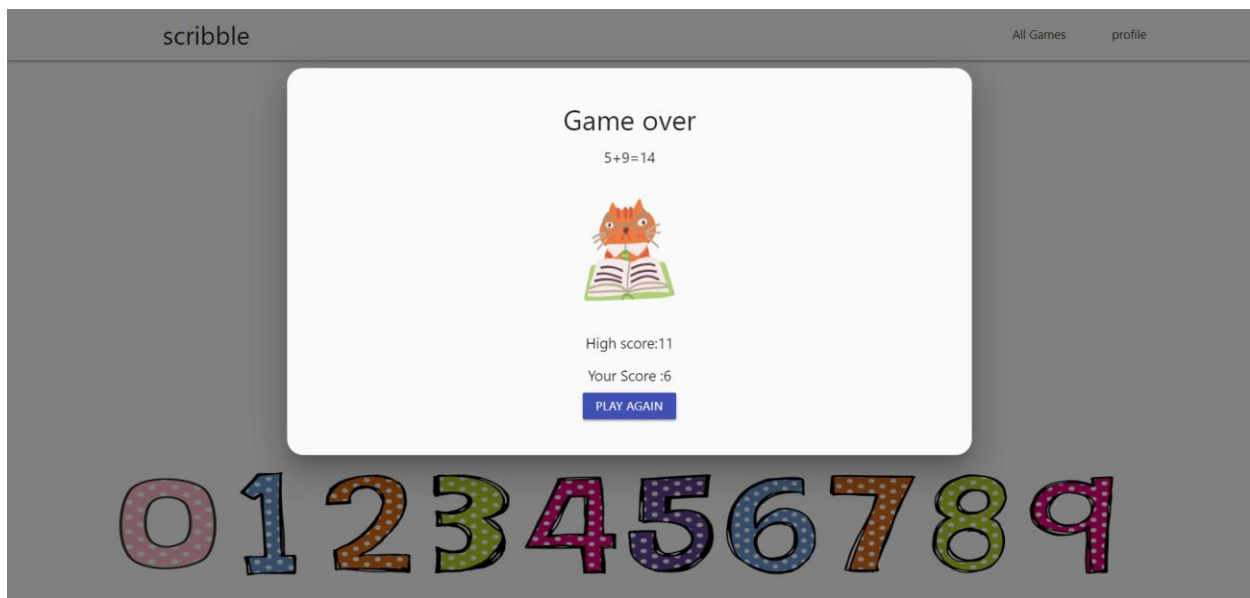
1	2								
---	---	--	--	--	--	--	--	--	--



Game over function displaying scores and correct answer



Game over in math module displaying scores and correct answer



Navigation to all modules from any module

scribble	Stories	profile
<p>Arrange the numbers in sequence</p> <div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div> <p>3 6 8 9 4 2 7 1 5</p>		
<p>127.0.0.1:5500/add/</p>		

Addition in Math Module

scribble	All Games	profile
<p>Addition</p> <p>HI : 11 Score : 21</p> <p>24 + 34</p> <div><div></div><div></div></div> <p>0 1 2 3 4 5 6 7 8 9</p>		


2. Mobile view

Stories – light mode


A+
A-
Aa




Little Red Riding Home


 By William son

 10 mins read


IN a great wide forest, full of beautiful trees, and green glades, and thorny thickets, there lived a long time ago a wood-cutter and his wife, who had only one child, a little girl. She was so pretty, and so good, that the sun seemed to shine more brightly when its light fell upon her rosy little face, and


stories – dark mode


A+
A-
Aa



Little Red Riding Home

 By William son

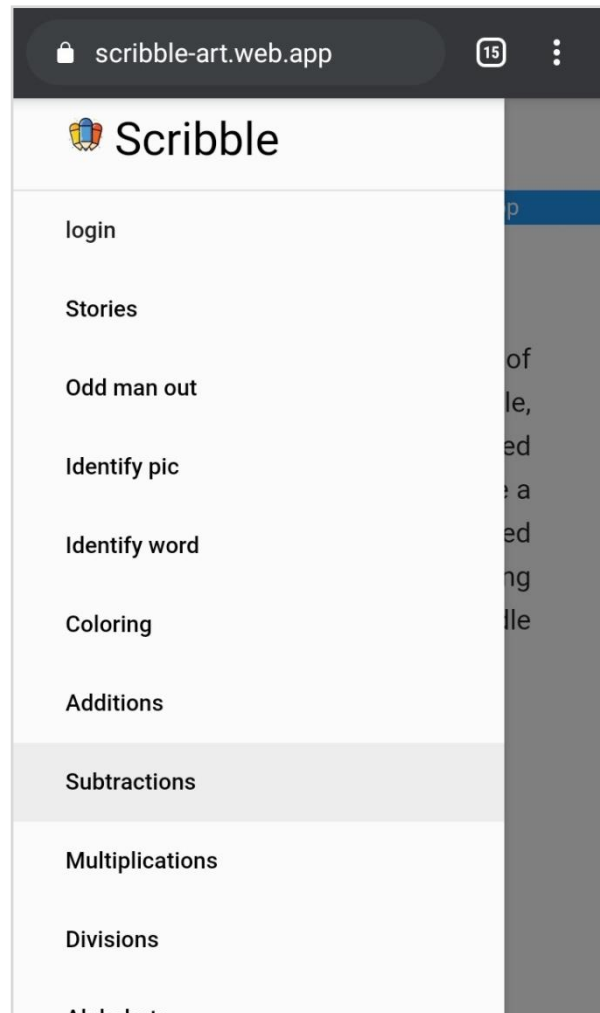
 10 mins read

IN a great wide forest, full of beautiful trees, and green glades, and thorny thickets, there lived a long time ago a wood-cutter and his wife, who had only one child, a little girl. She was so pretty, and so good, that the sun seemed to shine more brightly when its light fell upon her rosy little face, and

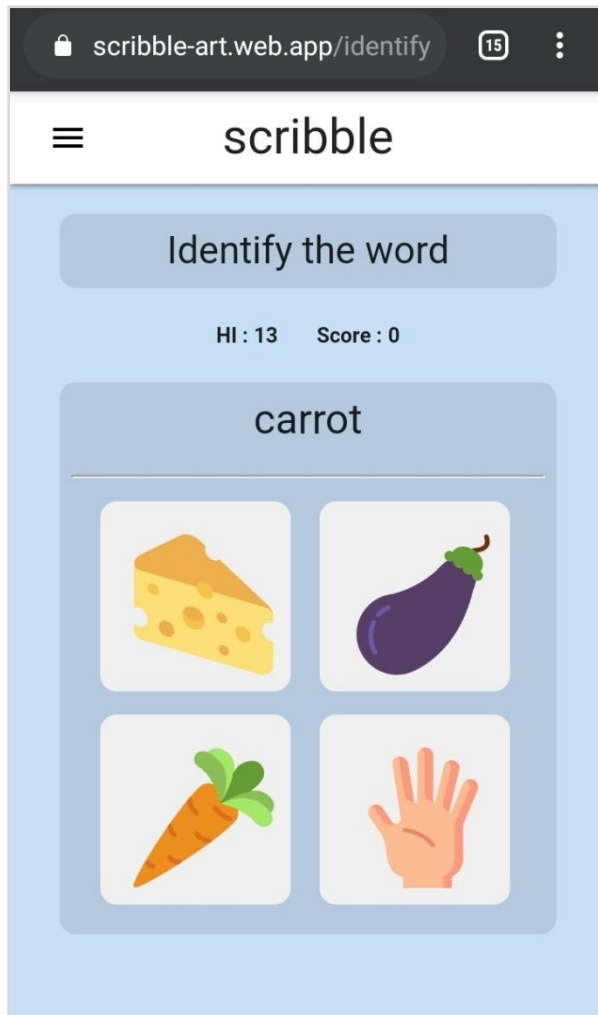
Coloring module



Mobile Navigation



Identify word module



Game over display with answers & scores



3. Backend

Database

The screenshot shows the Firebase Database console for a project named 'scribble'. The left sidebar contains navigation links for Project Overview, Develop (Authentication, Database, Storage, Hosting, Functions, ML Kit), Quality (Crashlytics, Performance, Te...), Analytics (Dashboard, Events, Conversi...), Grow (Predictions, A/B Testing, Clo...), Extensions, and Spark (Free \$0/month, Upgrade). The main content area is titled 'Database' and shows the 'Cloud Firestore' database. The 'Data' tab is selected, displaying a collection named 'users' with a document ID 'DfkrOIHU4wXAWABNWGp8WRuZ5073'. The document contains the following fields:

Field	Value
a	21
alpha	21
d	11
ipic	12
word	14
m	10
num	10
odd	7
s	15

Authentication control

The screenshot shows the Firebase Authentication console for the same project. The left sidebar is identical to the previous screenshot. The main content area is titled 'Authentication' and shows the 'Users' tab. A search bar at the top allows searching by email address, phone number, or user UID. Below the search bar is a table listing users:

Identifier	Providers	Created	Signed In	User UID ↑
pandurangaa99@gmail.c...	📧	Apr 14, 2020	Apr 14, 2020	DfkrOIHU4wXAWABNWGp8WR...
humanstar09@gmail.com	📧	Apr 9, 2020	Apr 14, 2020	QKueLg9p7uebjeSzYakpwTSnf...
fakeis607@gmail.com	📧	Mar 31, 2020	Mar 31, 2020	lh5kji79MkSablUCzgZRLRS...

At the bottom of the table, it shows 'Rows per page: 50' and '1-3 of 3'.

i. Testing

The web application testing consists of

- Usability Testing
- Functional Testing
- Compatibility Testing
- Database Testing
- Performance Testing

Usability Testing

Usability testing is nothing but the User-friendliness check. In Usability testing, the application flow is tested so that a new user can understand the application easily. Basically, system navigation is checked in Usability testing.

Test cases

- ✓ Web page content should be correct without any spelling or grammatical errors
- ✓ All the text should be properly aligned.
- ✓ Enough space should be provided between field labels, columns, rows, and error messages.
- ✓ All the buttons should be in a standard format and size.
- ✓ Home link should be there on every single page.
- ✓ Disabled fields should be grayed out.
- ✓ Scroll bar should appear only if required.
- ✓ Title should display on each web page

Functional Testing

Testing the features and operational behavior of a product to ensure they correspond to its specifications.

Test cases

- ✓ Test all the mandatory fields should be validated.
- ✓ Test the random functionality.
- ✓ Test the Shuffling functionality.
- ✓ Test the functionality of the buttons available
- ✓ Test the java script is properly working in different browsers (IE, Firefox, Chrome, safari and Opera).

Compatibility Testing

The purpose of Compatibility testing is to evaluate how well software performs in a particular browser, Operating Systems, hardware or software.

Test cases

- ✓ Test the website in different browsers (IE, Firefox, Chrome, Safari and Opera) and ensure the website is displaying properly.
- ✓ Test the HTML version being used is compatible with appropriate browser versions.
- ✓ Test the images display correctly in different browsers.
- ✓ Test the fonts are usable in different browsers.
- ✓ Test the java script code is usable in different browsers.
- ✓ Test the Animated GIF's across different browsers.

Database Testing

In Database testing backend records are tested which have been inserted through the web or desktop applications. The data which is displaying in the web application should match with the data stored in the Database.

Test Cases

- ✓ Verify the Tables, columns, column types and defaults: All things should match with the specifications.
- ✓ Verify whether the column allows a null or not.
- ✓ Verify the parameter names, types and number of parameters.
- ✓ Test the parameters if they are required or not.
- ✓ Test when the output is zero, the zero records should be affected.
- ✓ Verify the data gets properly saved into the database after each page submission.

Performance Testing



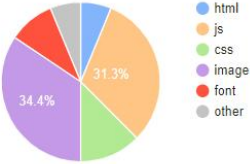
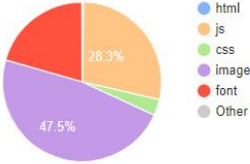
Performance Testing is conducted to evaluate the compliance of a system or component with specified performance requirements.

Test Cases

- ✓ To determine the performance, stability and scalability of an application under different load conditions.
- ✓ To determine if the current architecture can support the application at peak user levels.
- ✓ To determine which configuration sizing provides the best performance level.
- ✓ To identify application and infrastructure bottlenecks.
- ✓ To determine if the new version of the software adversely had an impact on response time.
- ✓ To evaluate product and/or hardware to determine if it can handle projected load volumes.

Automated Testing results

Run 2:

	Waterfall	Screenshot
First View (3.177s)		
<u>Content Breakdown</u>	<p>Requests</p> 	<p>Bytes</p> 

Performance Results (Median Run - SpeedIndex)

	Load Time	First Byte	Start Render	First Contentful Paint	Speed Index	Last Painted Hero	First CPU Idle	Document Complete			Fully Loaded			
								Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View (Run 2)	3.177s	0.725s	1.700s	1.770s	1.701s	1.700s	> 1.700s	3.177s	31	1,029 KB	3.440s	32	1,029 KB	\$\$\$- =

j. Empirical Estimation

1. LOC

Lines of code in our project = 12,439

Let,

LOC a person can code efficiently in a month = 6000 LOC

Average salary for a person per month = \$1000

Then,

No of persons required to complete the project = $12439/6000 = 6.21 \approx 6$ persons

Cost required to develop the software = $6 \times 1000 = \$6000$

2. FP

I/p - Information entering the system = 23

O/p - Information leaving the system = 11

Enquires- Request for instant access of information = 52

ILF - In held files within system = 16

ELF - In held by other systems that is analyzed = 04

By comparing we can assume the system is complex

So,

Total count = $23 \times 6 + 11 \times 7 + 52 \times 6 + 16 \times 15 + 4 \times 10 = 807$

CAF = $0.65 + 0.01 \times \sum f_i = 1.07$

FP = CAF x Total count = 863