A
PROJECT REPORT
ON

DSA Visualization

B.Tech (CE) Sem-VII
In fulfilment of all requirements for

Bachelor of Technology
In
Computer Engineering

Submitted by

Desai Kandarp D. (CE-022)
Gandhi Utkarsh S. (CE-025)
Ghanshani Vijay D. (CE-026)

Under the Guidance of
Prof. Purvi B. Soni

DEPARTMENT OF COMPUTER ENGINEERING
FACULTY OF TECHNOLOGY,
DHARMSINH DESAI UNIVERSITY
COLLEGE ROAD, NADIAD- 387001

DHARMSINH DESAI UNIVERSITY NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project carried out in the subject of System Design Practices entitled "DSA Visualization" and recorded in this report is a work of

1) Desai Kandarp D.	**ROLL NO: CE-022**	**ID: 13CEUOT021**
2) Gandhi Utkarsh S.	**ROLL NO: CE-025**	**ID: 13CEUOS115**
3) Ghanshani Vijay D.	**ROLL NO: CE-026**	**ID: 13CEUOS100**

Of Department of Computer Engineering, semester VII. They were involved in Project developing during academic year 2016 -2017.

Prof. Purvi B. Soni
(Project Guide),
Department of Computer Engineering
Faculty of Technology
Dharmsinh Desai University, Nadiad

Dr. C.K.Bhensdadia,
Head, Department of Computer Engineering
Faculty of Technology,
Dharmsinh Desai University, Nadiad

Abstract

Anyone who writes a large amount of code and deals with the efficiency of the code needs to have a proper data structure, that also organize the data efficiently. Whenever we view or process data on a computer system, there is always a data structure behind what we are doing.

Data structure, in simplest terms, is data organization for its efficient use. Data structures can be of various types, depending on the application. For example, databases use different data structures than compilers. Data structures allow information storage on hard disks, provides means for management of large dataset such as databases or internet indexing services. Many algorithms apply directly to a specific data structures. When working with certain data structures we need to know how to insert new data, search for a specified item, and deleting a specific item. So learning data structures is the most crucial part of a person dealing with the software development.

Contents

1. Introduction

1.1 Project details: Broad specifications

DSA Visualization is an Android application. When we are creating something (an application, for example), proper data structures and required algorithms needs to be implemented while making the application.

DSA Visualizations helps you learn various data structures and algorithms through animations and its abstract introduction. The animations help you understand every crucial concept of data structures and algorithms. It also allows you to take screenshots of the animation at any step. Once you are done with learning the data structures and algorithms you can also test yourself by having a quiz of different multiple choice questions related to the data structures and algorithms.

1.2 Technology Used

Front End Tool: Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android platform development. It is used to develop Android Applications. It internally uses Java programming language and XML.

Language: Java.

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.

Markup Language: XML

In computing, **Extensible Markup Language** (**XML**) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine readable.

Scipting Language: Java Script

JavaScript ("JS" for short) is a full-fledged dynamic programming language that, when applied to an HTML document, can provide dynamic interactivity on websites.

Diagram Tool: Star UML

StarUML is a UML tool by MKLab. It is used for creating UML diagrams and other various types of diagram.

2. Software Requirement Specifications

2.1 Introduction

The best way to understand complex data structures is to see them in action. We've developed an android application which contains interactive animations for a variety of data structures and algorithms. Our visualization tool is written in JavaScript using the HTML5 canvas element. User can learn data structures and algorithms through abstract introduction of each one and by different kinds of animations.

2.2 Purpose

Data Structures has been one of the most crucial subject to learn for any Computer Science or IT student. Data structure is a particular way of storing and organizing information in a computer so that it can be retrieved and used most productively. Data structures provide a means to manage large amounts of data efficiently for uses such as large databases and internet indexing services. Usually, efficient data structures are key to designing efficient algorithms. Some formal design methods and programming languages emphasize data structures, rather than algorithms, as the key organizing factor in software design. Data structures can be used to organize the storage and retrieval of information stored in both main memory and secondary memory. So we have developed an android application which helps anyone to learn every type of data structure from the basics through variety of animations.

2.3 Scope

- A Data Structure visualization application by the name “DSA Visualizations” is to be built.
- The application will allow the user to see and study animations for different kind of data structures and algorithms.
- After the completion of animation, user can take the screenshot of the animation and share it through different application.
- User can test his/her skills by going through a quiz of multiple choice questions.

2.4 Definitions, Acronyms and Abbreviations

DSA: Data Structures and Algorithms

2.5 Hardware Requirements

- Smartphones with android OS will be the target audience

2.6 Software Process Model

- The application would be developed using Agile method of software development.
- Agile software development (Agile) is a collection of software development methodologies that promote adaptive planning, evolutionary development and delivery, continuous improvement, and a time-boxed period of time to complete a body of work.
- This method is usually adopted when the project size is relatively small, experience of the developers in the domain is less and the requirements aren't clear as well as they are volatile.

Advantages of Agile model:

- New features are delivered quickly and frequently, with a high level of predictability.
- As the project is divided into manageable and small chunks the progress is visible and transparent.
- Changes can be easily adapted upon.

2.7 User Characteristics

- **End User**:
  User should have basic knowledge of Android operating system.
- **Administrator**:
  Administrator should have the knowledge about handling databases.

2.8 Nonfunctional Requirements

- Reliability:

  Users should be prompted with appropriate message when the application isn't able to process animation or isn't able to load JavaScript file due to change in the path of the file explicitly by the user.

- Performance:

  Response time will be kept as low as possible in order to ensure promptness for the user.

- Usability:

  User friendly user interface must be designed in order to ensure ease of use.

2.9 Installation Requirements

The system to be developed doesn't require any other software to be present on client's machine.

2.10 Functional Requirements

R1: Application should provide basic information about all data structures.

Input: Selected Data structure.

Output: Basic information of the data structure.

R2: Application should allow user to manipulate all data structures and visualize operations through animations.

Input: Selected Data structure.

Output: Animation of selected Data Structure.

R3: Application should allow to test skills through the quiz.

Input: Select Test skills option.

Output: Multiple Choice Question.

R4: Application should allow user to understand different algorithms using animations.

Input: Selected Data structure.

Output: Animation of selected algorithm.

R5: Application should allow user to take screenshot of the animation.

Input: Animation of the selected data structure and algorithm.

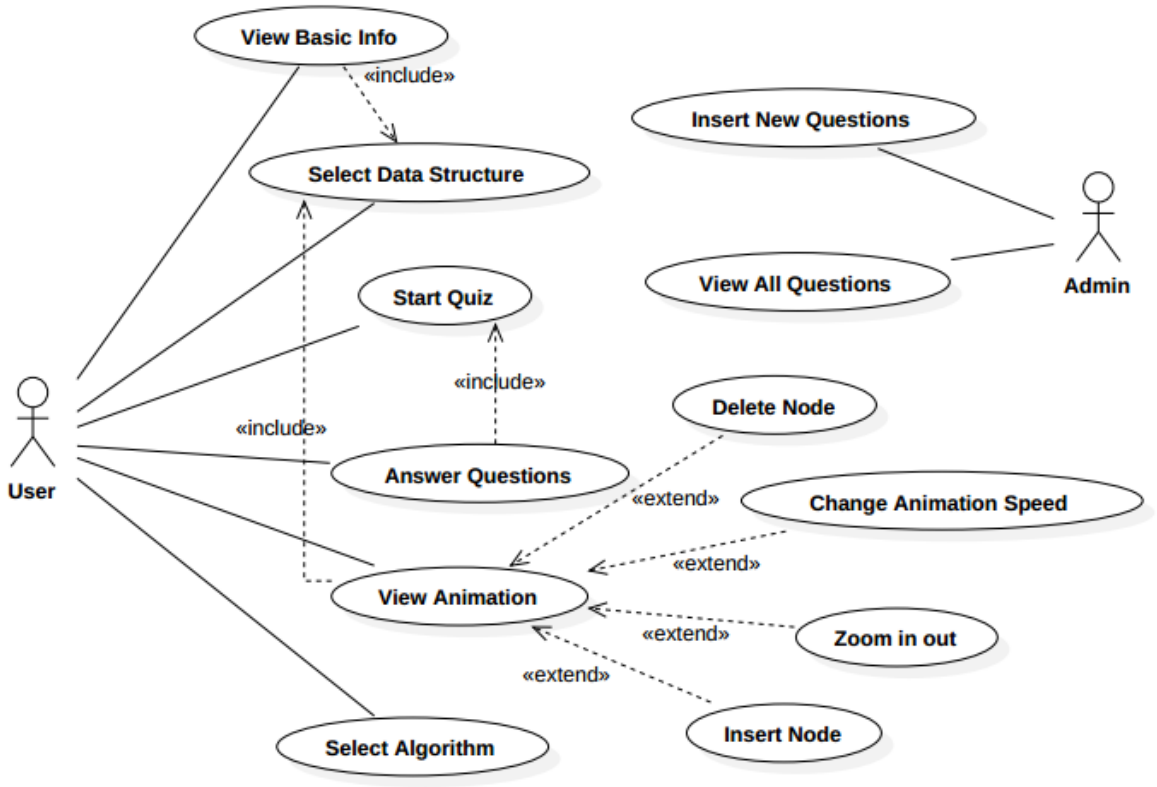
Output: Screenshot of the animation.

R6: Application allows user to share the screenshot.

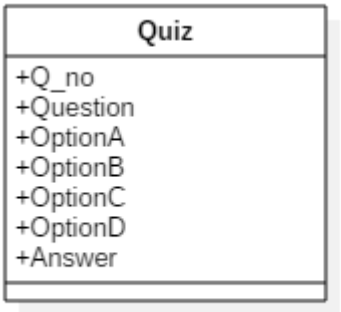
Input: Screenshot.
Output: Screenshot shared.

3. System Design

3.1 Use-case Diagram

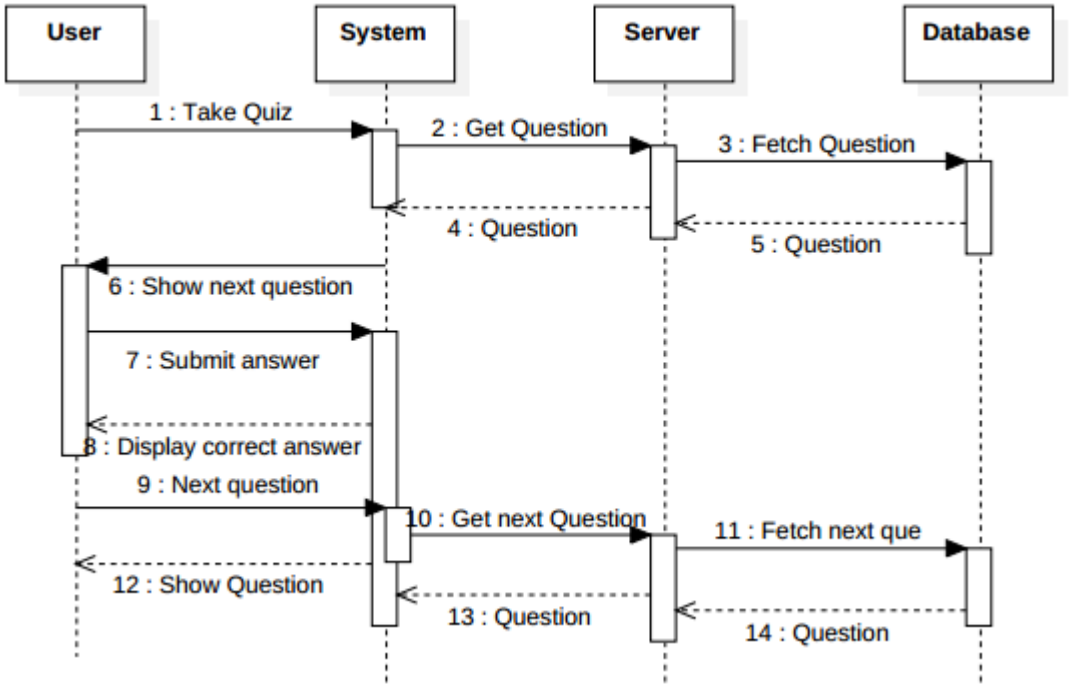


3.2 Class Diagram

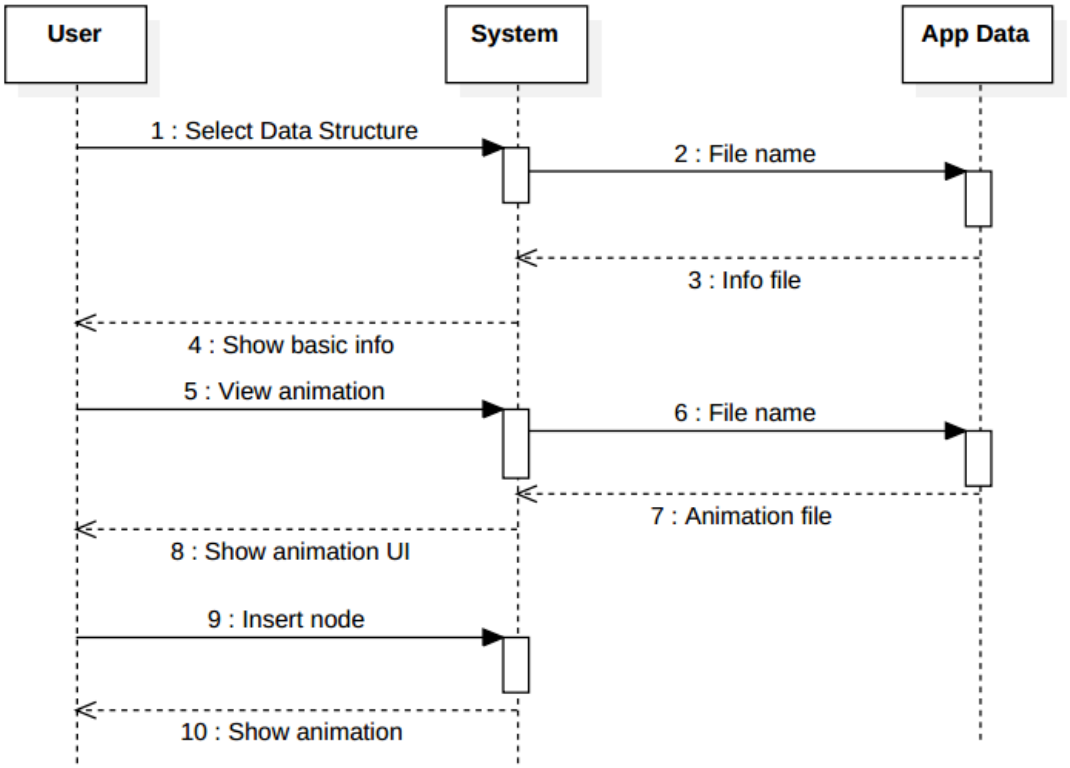


3.3 Sequence Diagrams

3.3.1 Diagram for Quiz

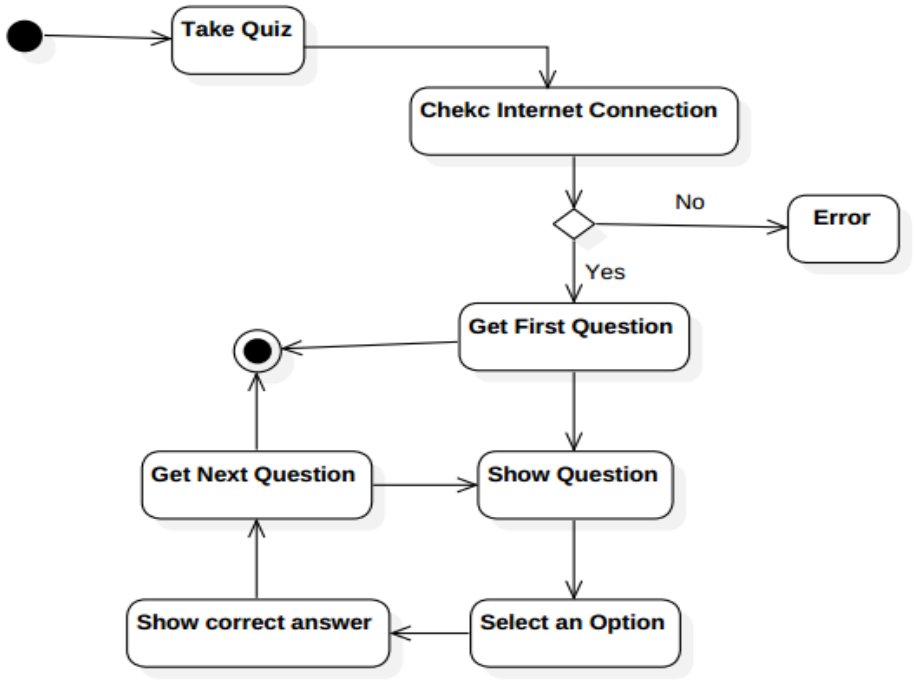


3.3.2 Diagram for View Animation

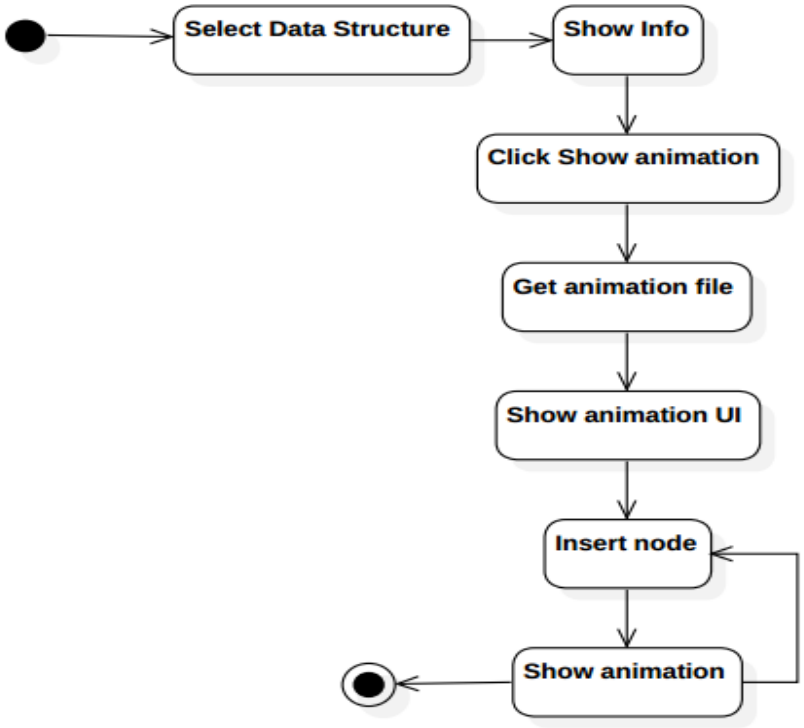


3.4 Activity Diagrams

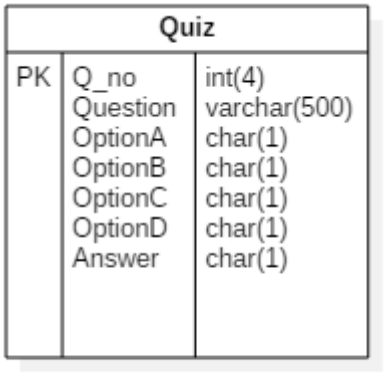
3.4.1 Diagram for Quiz



3.4.2 Diagram for View Animation



3.5 Table Relationship Diagram



3.6 Data Dictionary

3.6.1 Definition for Collaborator Table

#	Field	Type	Null	Key
1	ID	int(11)	NO	PRI
2	Question	varchar(200)	NO	
3	OptionA	varchar(50)	NO	
4	OptionB	varchar(50)	NO	
5	OptionC	varchar(50)	NO	
6	OptionD	varchar(50)	NO	
7	Answer	varchar(50)	NO	

4. Implementation

4.1 Implementation Environment

- Android Studio
- XML
- Java Script
- Java

4.2 Modules Description

- **Abstract Introduction:**

Abstract: It is used for an introduction to data structure or algorithm.

Input parameter: Data structure or algorithm name.

Output parameter: Introduction for the respective data structure or algorithm.

Processing logic: The introduction for the selected data structure or algorithm will be displayed in the webview.

- **Animation**:

  Abstract: It is used for seeing animation for a particular data structure or algorithm.

  Input parameter: Data structure or algorithm name.

  Output parameter: Animation window for the selected Data structure or algorithm.

  Processing logic: The animation for the selected data structure or algorithm will be displayed in the webview.

5. Testing

5.1 Testing Plan

The testing is a technique that is going to be used in the project is black box testing the expected inputs to the system are applied and only the outputs are checked.

5.2 Testing Strategy.

The development process repeats this testing sub process a number of lines for the following phases.

- Unit Testing
- Integration Testing

Unit Testing tests a unit of code after coding of that unit is completed. Integration Testing tests whether the previous programs that make up a system, interface with each other as desired. System testing ensures that the system meets its stated design specifications. Acceptance testing is testing by users to ascertain whether the system developed is a correct implementation of the software requirements specification.

Testing is carried out in such a hierarchical manner to that each component is correct and the assembly/combination of component is correct. Merely testing a whole system at end would most likely throw up errors in component that would be very costly to trace and fix. We have performed both Unit Testing and System Testing to detect and fix errors.

5.3 Testing Methods

We have performed Black-box testing for the testing purpose. A brief description is given below:

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or

workings. This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well.

5.4 Test Cases

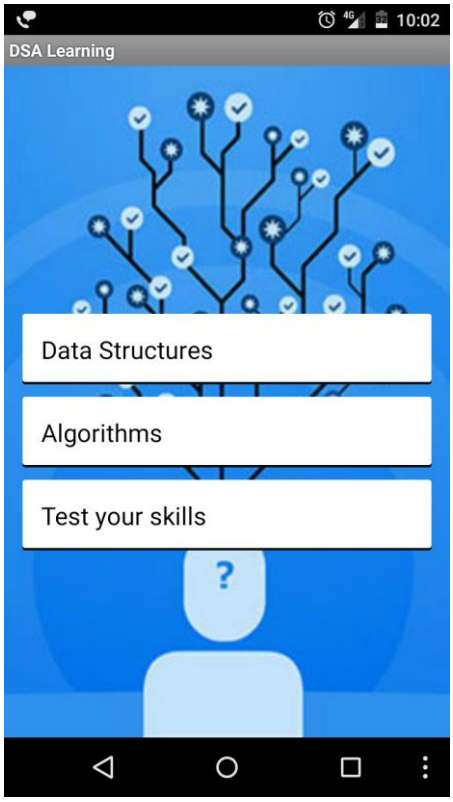
Test cases are described below:

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results
T01	View Basic Information	1. Open App 2. Select data structure	Data Structure Name	Information displayed	As per expected
T02	Insert Node	1.Select Data structure 2. Click View Animation 3. Type data to be inserted 4. Click insert	Insert :5	Node inserted	As per expected
T03	Delete Node	1. Type data to be deleted 2. Click delete	Delete :5	Node deleted	As per expected
T04	Change Animation Speed	1. Open Animation page 2. Move the slider to the right	Increase Speed	Successful	As per expected
T05	Change Animation Speed	1. Open Animation page 2. Move the slider to the left	Decrease Speed	Successful	As per expected
T06	Answer Question	1. Open quiz 2. Tap on an option	Option Number: A(Correct)	Green Background	As per expected
T07	Answer Question	1. Open quiz 2. Tap on an option	Option Number: B(Wrong)	Red Background & Correct Ans.	As per expected

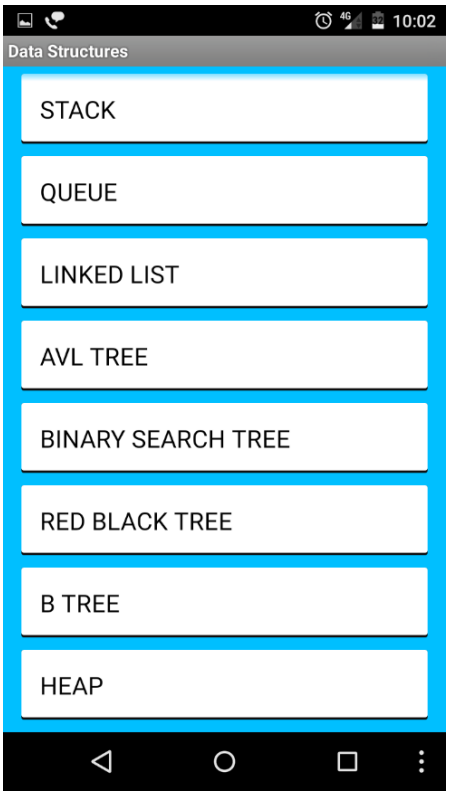
T08	Get Next Question	1. Open quiz 2. Click next button	Question number	Next question displayed	As per expected
T09	Get Next Question	1. Open quiz 2. Click previous button	Question number	Previous question displayed	As per expected

6. Screenshots

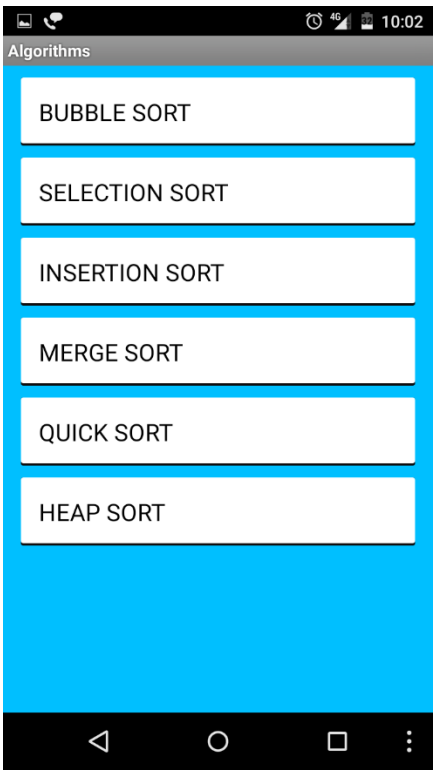
6.1 Home Page



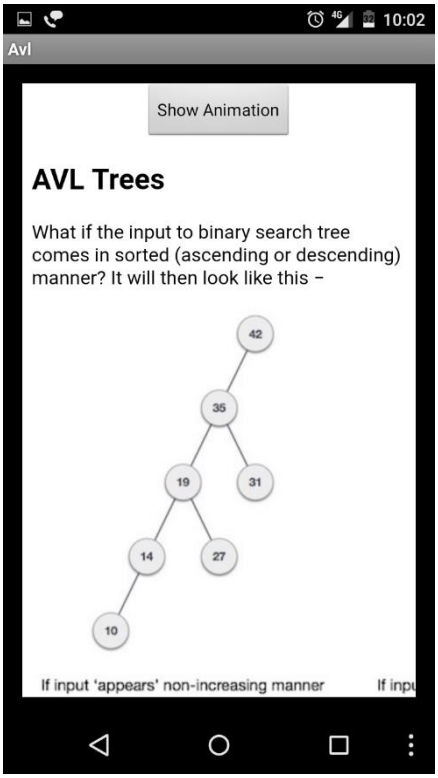
6.2 Data Structures List



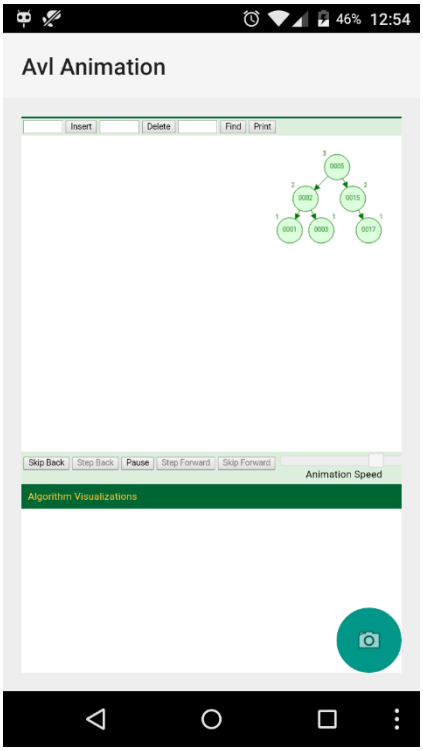
6.3 Algorithm List



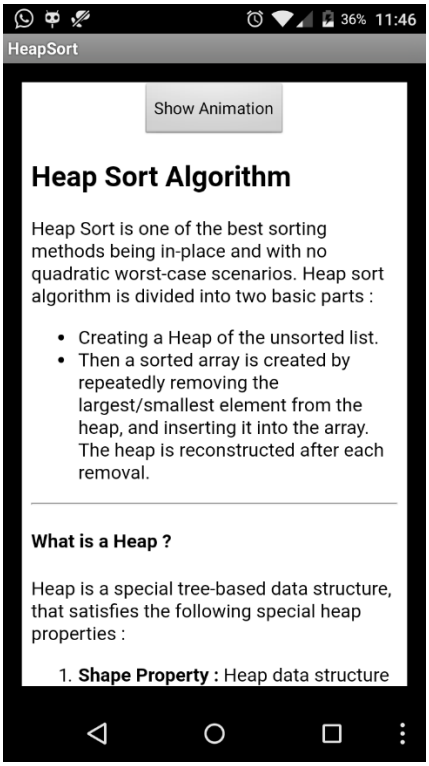
6.4 AVL Tree Introduction



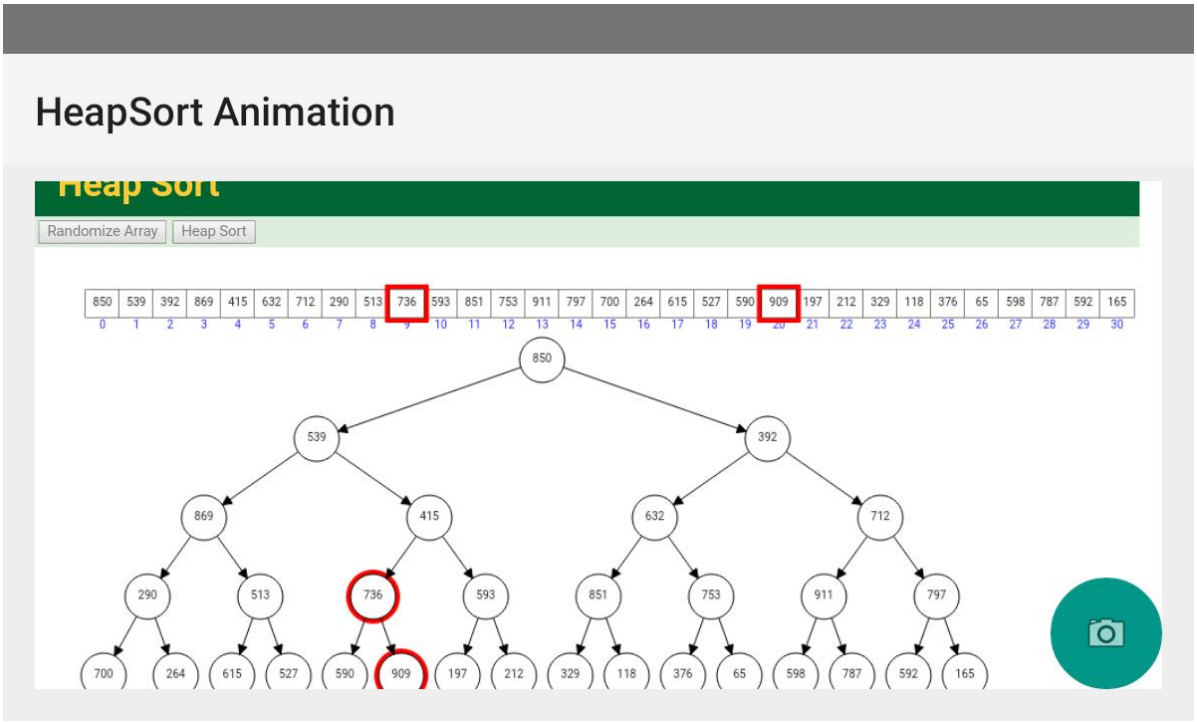
6.5 AVL Tree Animation



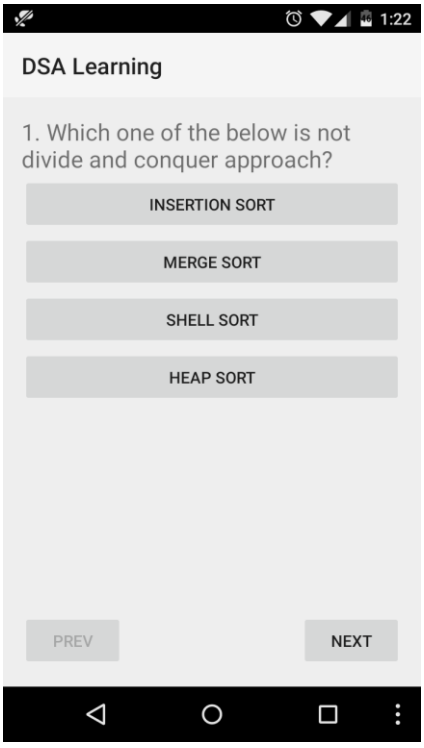
6.6 Heap Sort Introduction

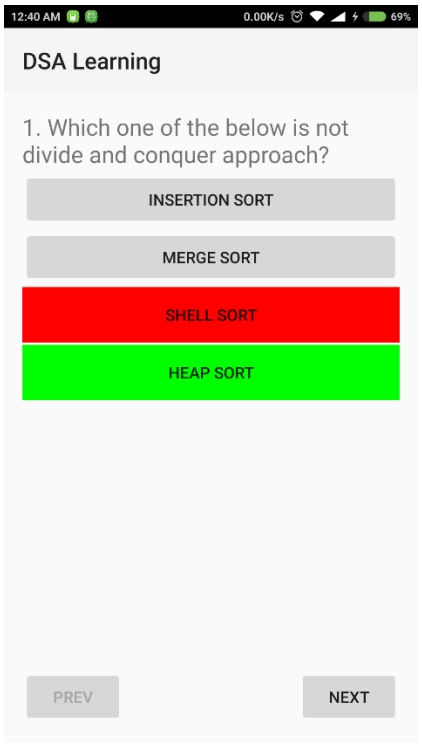


6.7 Heap Sort Animation



6.7 Test Your Skills





7. Limitations and Future enhancements

Limitations:

- Limited data structures are available.
- Only sorting algorithms and searching algorithms are available.

Future Enhancements:

- New Data structures will be added in future implementations.
- New Algorithms will be added in future implementations.
- Various settings for the nodes will be added in future implementation.

8. Conclusion

Here by we declare that we had performed a project by understanding all module of this project. We checked the feasibility and requirement for this system. Then we defined overall look and flow of control among modules in paper. After this, we started actual design of our modules of system in Java. All modules of system were developed separately. Then we integrated all modules by means of control flow among all modules.

After Coding and integrating of all modules done, then we tested all modules separately this is basically Unit Testing of an all modules.by completion of Unit Testing, whole system are then tested once again this called Integration Testing. Test cases were designed by performing black box testing.

9. Bibliography

Books:

- Android Programming: The Big Nerd Ranch Guide
  -By Bill Philips & Brian Harddy