CALIFORNIA STATE UNIVERTSITY, LONG BEACH
CECS 575 Object Oriented Analysis and Design

PROJECT NAME : AlgoGuru

Student Name and Number
Katrin Raab

Ronit Kumar

Sujeeth Panicker 014828682

Vansh Bajaj

Contents

Abstract

This project is an effort to ease the process of learning algorithms by facilitating the student with adequate study resources, engaging animations and exercises. For the instructor it as an effective teaching aid to suggest to students, for class presentations and visualisation for complex algorithms and data structures. The current method of learning algorithms through paper and board examples lacks effective visualization and often leaves the student with an half-baked idea of the algorithm or worse yet she loses interest in the topic altogether owing to its complexity.

The mobile application aims to be a go-to reference for computer scientists to delve into concepts of algorithms and data structures. An engaging application is one of our core goals to enhance the learning experience.

The project plan gives a brief idea of what to expect in the future and how the project needs to be implemented. A generic mobile architecture is explained and its adoption to our requirements. Early mock ups of the functional scenarios of the application is also available for critique.

1. Introduction

1.1 **Algorithms:**

An algorithm in compting terms is a set of steps taken to carry out a particular taskor tasks.

- But why are they so important?

Algorithms today are employed in almost every facet of technology from sophisticated astronomical devices and supercomputers to simple appications like your email app.

They are primarily used for their ability to

a) Carry out tasks correctly
b) Do tasks efficiently

By ”correctly” we mean the ability to get the expected result on completing the required steps and by ”efficiently” we mean the optimal use of resources including time.

It is vital for computer scientists to understand and apply algorithms to problems to solve real world problems and contribute to the betterment of mankind.

2. Purpose of the project

- The problem we aim to solve is the dearth of a quality application that would demystify algorithms to a budding computer scientist through a simple but engaging interface.

- Instructors still use paper or the board to show examples and workings of an algorithm to students.

- It is difficult even for the most imaginative of students to visualize the steps involved in complex algorithms and their subsequent impacts on a data structure.

3. Solution

- The objective of this project is to develop a mobile application that aids any user interested in algorithms to learn the various kinds of algorithms popularly used in computing technologies and studied at university level computer science programs.

- Links to prerequisite material will also be available in the app for users to better understand the concepts explained.

- The app aims to break down the algorithms and show how the input data is manipulated at each step so that the user has a deeper and concise understanding of the algorithm.

- Each algorithm will have a quiz associated with it to ensure that the user has understood the concept completely.

- **Why this solution?**
  This solution not only engages the user but also provides her with a comprehensive understanding of the inner workings of an algorithm. Moreover, she can revisit the material at any given time and revise her knowledge of any particular algorithm.

- **Who shall benefit?**

  Budding computer scientists and instructors who can use this application as a reference or teaching aid.

- **Are there any competing solutions?**

  Yes there are competing solutions like Algorithms : Animated and Explained and Algorithm Visualizer.

- **What makes our solution unique?**

  None of these solutions engages the user to the degree we intend to. In addition, there are no prerequisite materials available to ease the user's learning process. The applications also do not ensure that the user's grasp of algorithms through questionnaires or feedback mechanisms.

- **Constraints:**
  Timespan: A project of this magnitude developed by a team of four amateur developers would take around 8-10 months including comprehensive testing. We intend to develop the core functionality of the app including the interface and algorithm analysis and animated explanations within the course timeline. We shall develop add-on features like badges and quizzes for users through incremental releases.

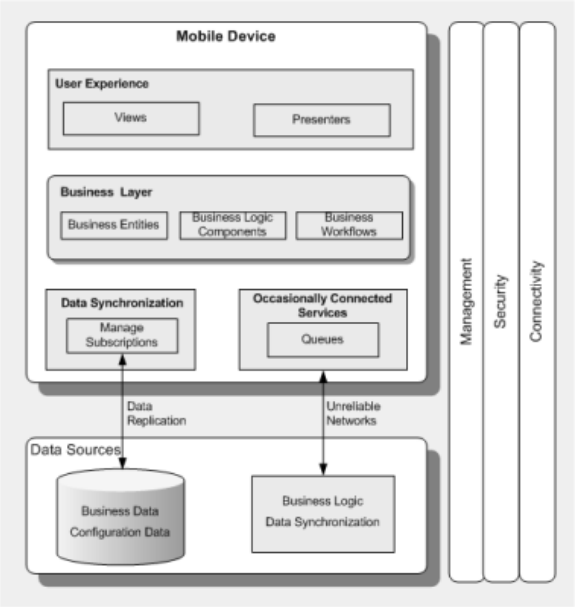
4. Roles & Responsibilities

Sujeeth	Manage project schedule, Delegate tasks Design back end system, Test and verify UI design, Documentation
Ronit	Develop UI, Test and verify back end systems, Documentation
Vansh	Develop UI, Test and verify back end systems, Documentation
Katrin	Design back end system, Test and verify UI design, Documentation

5. Proposed Architecture

Generic Mobile Application architecture:

[3]



App architecture design is a process, which has to be executed in a defined flow. The flow includes three different layers. Namely,

1. Presentation Layer

This layer comprises UI components and UI process components (Views and Controllers). At this layer, the team has to define the way the mobile app will present itself in front of the end users.

This layer is primarily focused on deciding the features and their location. However, simultaneously, the team also decides other aspects like the theme, font size, etc.

2. Business Layer

As the name suggests, the layer focuses on the business front. In simple language, it focuses on the way business will be presented in front of the end users. This includes workflows, business components, and entities beneath the hood of two sub-layers named service and domain model layers.

While the service layer focuses on defining a common set of application functions that will be available to client and end users, the domain model layer represents expertise and knowledge linked to the specific problem domain. The entire plan is formulated in a way to explore and enhance the future of application.

3. Data Layer

At this third stage, data-related factors are kept in mind. This includes data access components, data helpers/utilities, and service agents. Here, the one thing that you need to keep in mind is that the three components sit under the two subheads, precisely, persistence layer and network layer.

While the former provides simplified access to data, which can be stored in a persistent storage or backend, the latter is responsible for networking calls.

Adapting the architecture to AlgoGuru's needs:

The database layer shall consist of MongoDB's database as a service. A caching layer over it using Memcached.

Java as backend server language as well as for android UI development.

6. Solution Qualities

Adaptability Requirements

The application will run on Android 7.0 Nougat and other google supported older OS versions and on an android emulator on the PC.

Scalability Requirements

Before the application is available on the playstore we will be adding features like:

1) Load Balancing to reduce the impact of usage surges and increase availability

2) Caching to improve database performance using frameworks like Memcached

There will be no technical debt as these features will be layers added onto the existing repository with minimal changes to already developed code.

Security Requirements

We intend to develop AlgoGuru keeping in mind the best practices for android application security. We will be keeping in mind the following points: [2]

- Inbound SMS listeners (command and control)
- Unsafe file creation
- Improper database storage
- Unsafe use of shared preferences
- Storage of sensitive data on mass storage device
- Content provider SQL injection
- APN or proxy modification

Privacy Requirements

The user shall be signing on using one's google account or a guest user profile, which has minimal user data requirement. The application does not intend to retain user information at any time including multimedia and location. If the user signs in through a google account, her data is then subject to Google's data security and usage policy.

Usability Requirements

> **Minimalistic in view**

Since the app is very specific and provides a very specific role we intend to have a minimalistic UI which caters to the required functionality.

>**Easy navigation**

The application shall allow ease of movement through different panels and frames

>**Intuitive usage**

We intend to add movements and responses that users are used to over time in apps and thus make their experience seamless.

>**Relevant Content**

The content on display shall be related to algorithms and will only stick to that particular topic except for prerequisite material.

The algorithmic breakdown will also be shown in phases through different animations.

7. Project Timeline:

Aug 31st to Sep 5th	Project consensus and planning Ascertaining requirements.
Sep 6th to Sep 12th	UML design and use case design
Sep 13th to Sep 19th	Back end skeletal coding and deciding on design patterns
Sep 20th to Sep 26th	Coding core entities and defining service layers
Sep 27th to October 4th	Complete back end design and selecting ui frames and colour schemes.
October 5th to October 11th	Back end testing and UI design
October 11th to October 18th	Complete UI design and carry out system tests
October 19th to November 3rd	System tests, verification and documentation

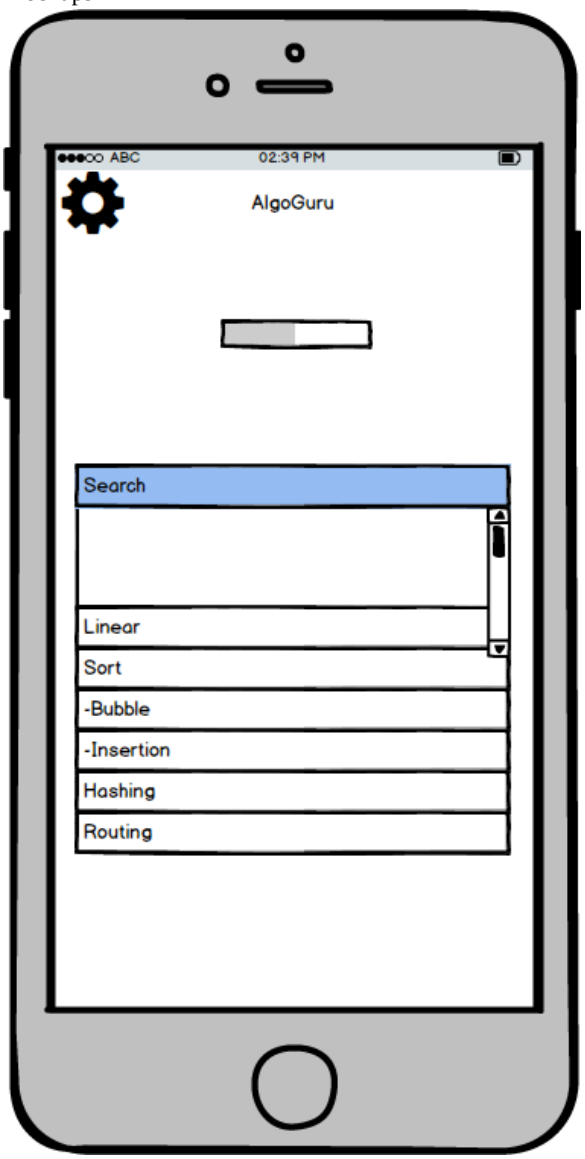
References

[1] Definition of algorithm source dictionary.com
[2] Veramod.com android best practices and security testing
[3] D Zone mobile architecture

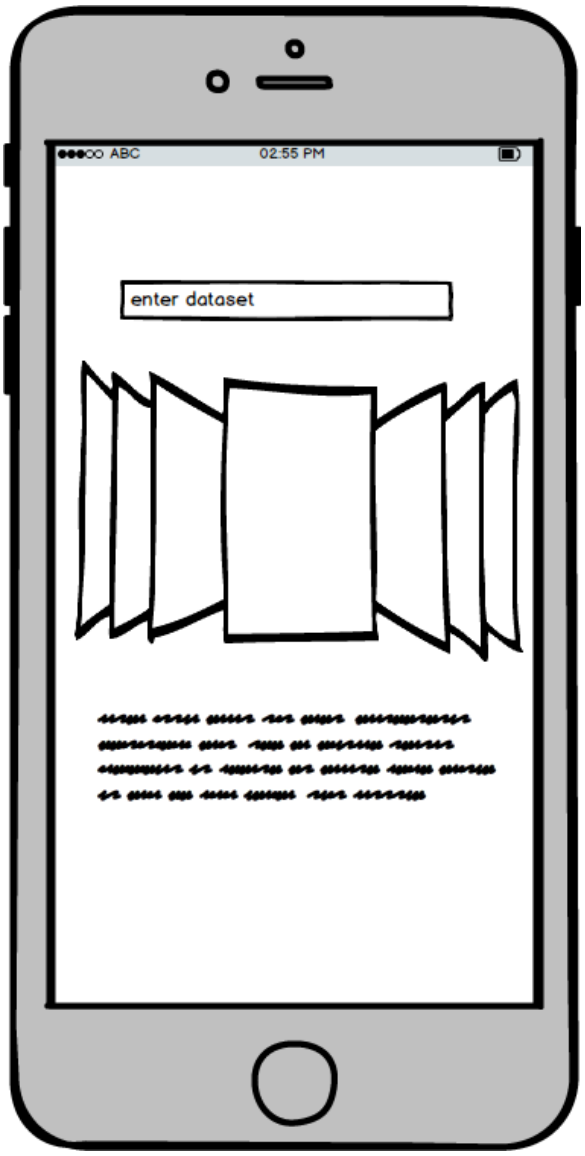
Appendix:

Commented [SP1]:

Mockups



1. Basic menu view with progress bar (main menu)



2. Mockup with data input, algorithm breakdown animation and explanation.