CSN-254
SOFTWARE ENGINEERING

Project: Recommendation System for CodeForces

Group No: 15

S.NO	NAME	ENROLLMENT NO.	EMAIL
1	BALNE NITEESHA	18114017	bniteesha@cs.iitr.ac.in
2	DIKSHA	18114019	diksha@cs.iitr.ac.in 9929155847
3	KARAN SINGH	18114035	ksingh1@cs.iitr.ac.in
4	KHUSHI	18114040	khushi@cs.iitr.ac.in
5	MANASVI	18114046	mpatidar@cs.iitr.ac.in

- ➢ System Design and Implementation.
- ➢ Testing.

System Design and Implementation

Followed from SRS Document

OVERVIEW OF PROBLEM STATEMENT

Competitive Coding can be a bit infuriating at times, imagine yourself in the shoes of a beginner you would either get bored by those easy problems if you sort them by descending number of solves or get discouraged by jumping straight into the difficult sections. Finding the right problem-set was always a hassle for me. Adding to these problems, CodeForces the most popular platform provides a very minimalistic GUI and a difficulty tag that makes little to no sense. This is a problem faced by all students preparing for a Computer Science oriented Job interview, irrespective of their college, branch or year.

This means that every year, thousands of students start Competitive Coding but only a select few excel at it, this is what we are aiming to cure. The lack of a proper recommendation system can demotivate a lot of these young and fragile minds, but with a proper guide, the students would really be empowered to explore this field to its true depth.

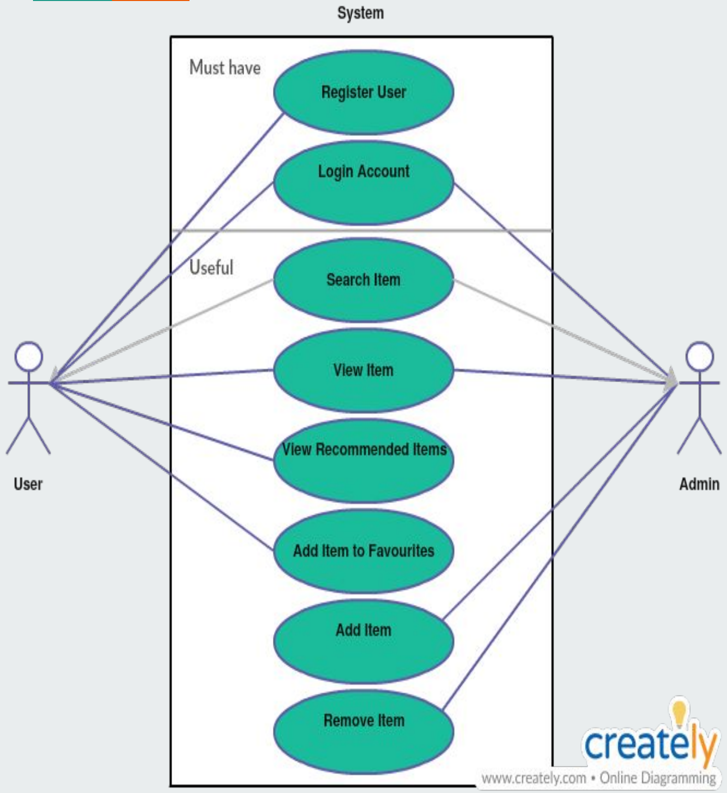
SOLUTION PROPOSED TO CUSTOMER

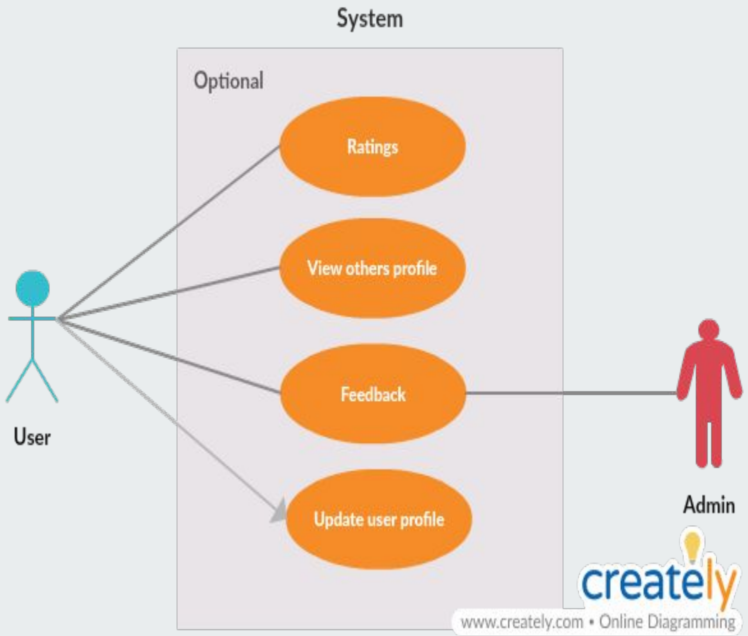
1. BUILD A MOBILE APPLICATION.
2. Requests information from users to **optimize their suggestions**. This information includes:-
   - User’s CodeForces username.
   - User’s Friend’s CodeForces usernames.
3. Accesses problems solved and other details of the user’s friends and the seniors using CodeForces’s API.
4. Display the accessed information in a raw yet well-formatted form.
5. Process the accessed information to generate the aforementioned lists.
6. Display these lists in an exhaustive manner.
7. Requesting user response for the problems they solved recently, on random occasions.

IMPACT OF PROPOSED SOLUTION

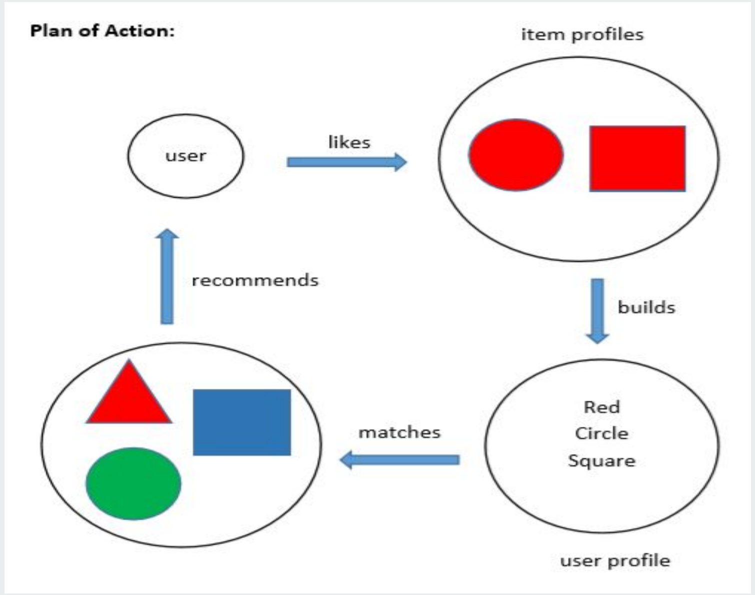
This application will not only act as a guide to CodeForces it will also compensate for the weak GUI they provide by enabling users to read the problems in-app. The user will use this application alongside their regular coding environment, the problems suggested by the application can be accessed swiftly on the browser by typing in the 2-5 character problem code and hitting enter. The user can also search for other problems while they are sitting with fingers crossed on the submissions page waiting for the “Accepted” to pop-up.

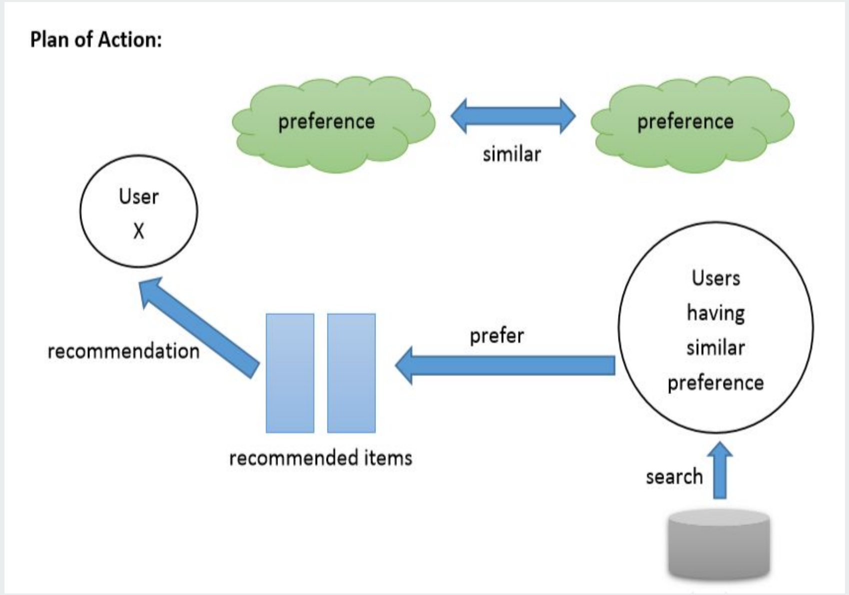
USE-CASES FOR PROPOSED SYSTEM



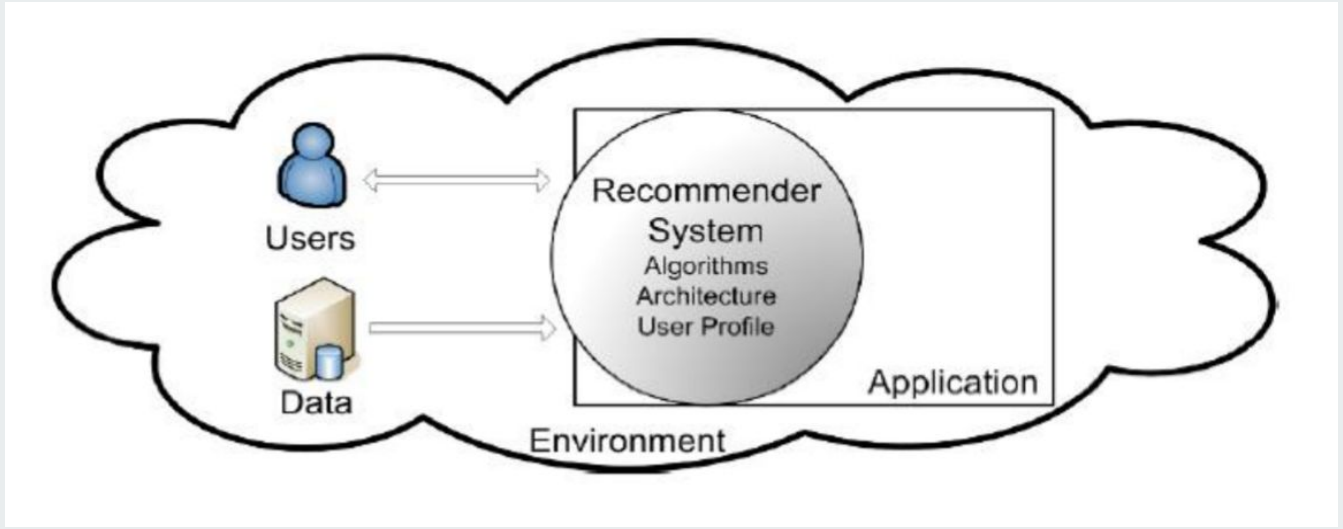


COLLABORATIVE FILTERING APPROACH FOR RECOMMENDATIONS





DESIGN OF APPLICATION



After going through the SRS document , the team finalised following tab designs:

1. **Sign in/Sign up:**

   This tab is a must have tab , user can sign in or sign up , if he/she doesn't already have an account.Once signed in, this account will be linked to user's codeforces account using the codeforces username .

2. **Solved problems:**
   This tab will show the list of problems already solved by the user , this will also ensure that already solved problems and recommended problems do not overlap and will have separate lists.

3.Recommended problems:

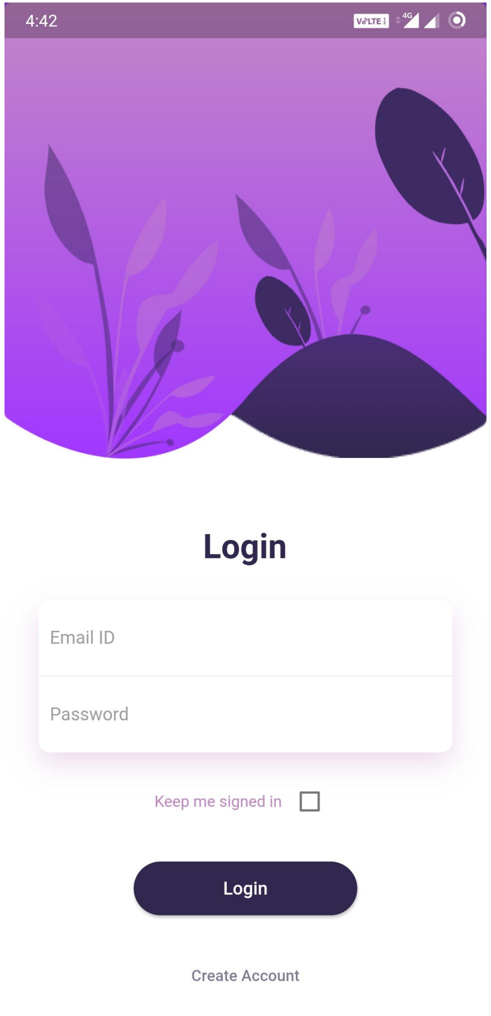
This is the main tab for the project , it will recommend the problems on the basis of user based, item based or mixed categories.

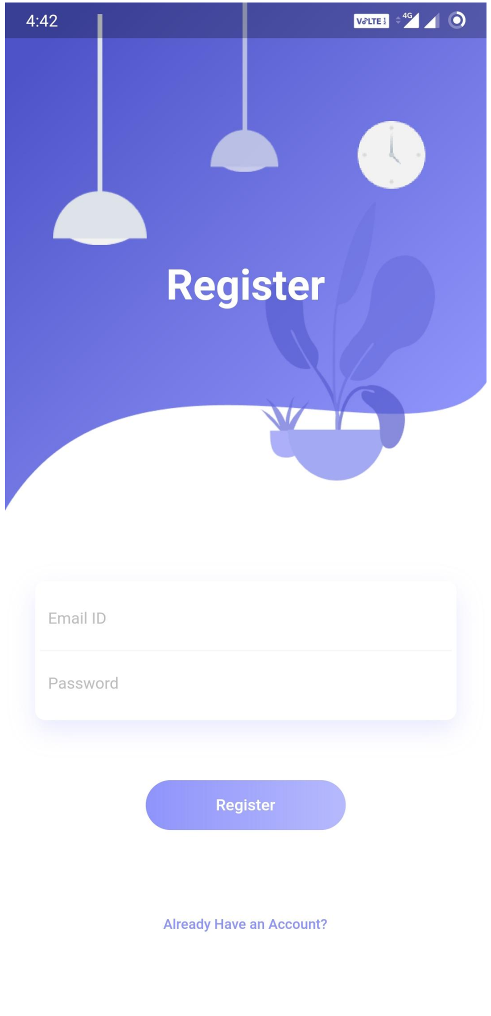
The user based category recommends the problems solved by majority of peers of user at the same level of rating or ranking.

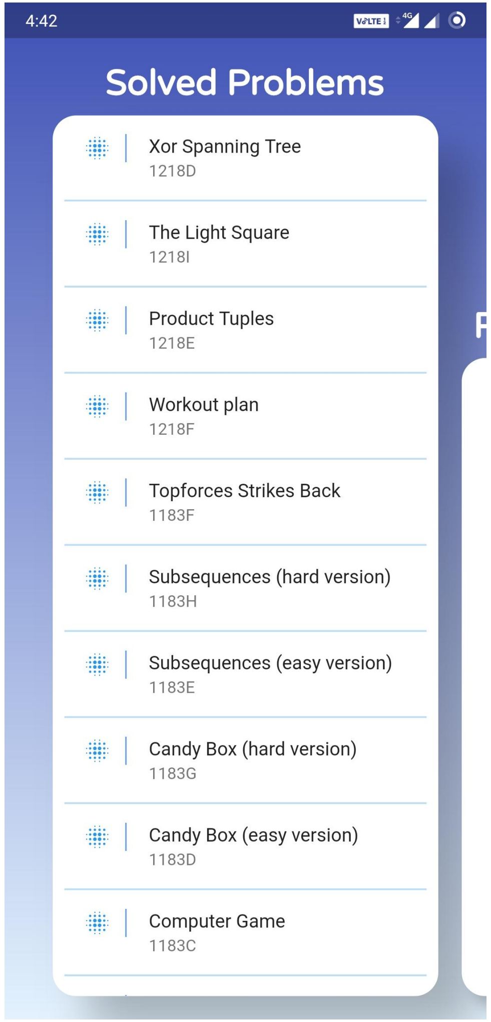
The item based category will recommend questions on the basis of target user’s previously liked questions only.

The mixed category will provide the benefit of both mentioned above , as it recommends according according to global users , so that the user can follow what other global users , at the same level as his/her , are doing , and also , it follows user’s choices , so that he/she can solve int one’s comfort zone too.

Screenshots of the app.


4:42
Login
Email ID
Password
Keep me signed in
Login
Create Account


4:42
Register
Email ID
Password
Register
Already Have an Account?


4:42

Solved Problems

Xor Spanning Tree
1218D

The Light Square
1218I

Product Tuples
1218E

Workout plan
1218F

Topforces Strikes Back
1183F

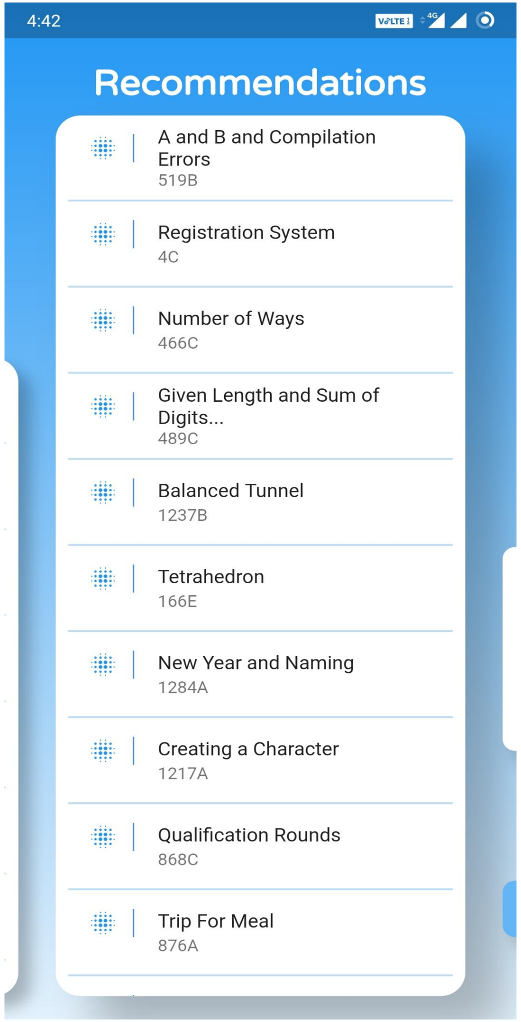
Subsequences (hard version)
1183H

Subsequences (easy version)
1183E

Candy Box (hard version)
1183G

Candy Box (easy version)
1183D

Computer Game
1183C


4:42

Recommendations

A and B and Compilation Errors
519B

Registration System
4C

Number of Ways
466C

Given Length and Sum of Digits...
489C

Balanced Tunnel
1237B

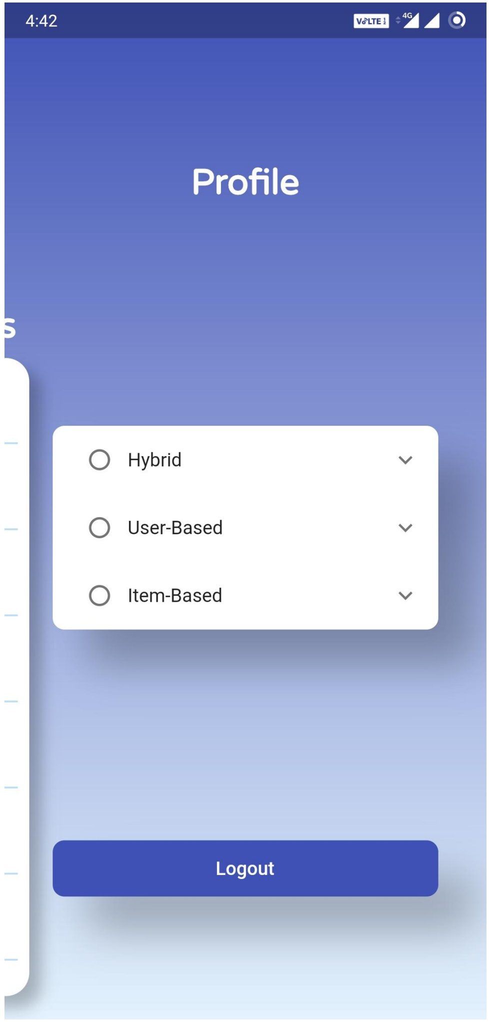
Tetrahedron
166E

New Year and Naming
1284A

Creating a Character
1217A

Qualification Rounds
868C

Trip For Meal
876A


4:42

Profile

Hybrid

User-Based

Item-Based

Logout

Testing

Test Policy

Testing was done to make sure the user experience is smooth and easy.

- Profiling of the flutter app.
- Benchmark test of the api calls.
- Penetration testing of the flutter app for security purposes.
- Firebase cloud firestore security rules were tested to make sure no data is being leaked.

Test strategy

The testing was 2 sided and the job distributed among the team equally:-

1. Test the Flutter Application.
2. Test the Firebase server which includes the Database and the Cloud functions.

Test Data

- The data received from CodeForces API.
- The data uploaded by the user.
- Data of auto generated recommendations.
- The user authentication database.

Test Conditions

The following worst case conditions were tested:-

- Slow internet.
- API calls while migrating the database.
- API race conditions.
- Mobile phones with low memory and processing power.

Test Report

- Screen stutters while switching pages in low end devices.
- The request wait time for recommendations is about 10 sec which shall be reduced.
- Render overflows in case tablets using the app made for mobile phones.
- API requires a key for enforcing authentication to access the public data.
- Animations may lead to app crash in old devices.

Coding And Implementation

Github: https://github.com/KaranKS/Project_CSN_254