Software Requirement Specification

Functional Requirements

CS 345-346

Competitive Programming Helper Forum App

Group 23

Name	Roll Number
Aditya Trivedi	190101005
Akshat Sinha	190101008
Atharva Vijay Varde	190101018

Contents

1 Introduction

1.1 Brief Overview

The purpose of this document is to build an application to aid engineering college students to improve their competitive programming skills. The document will exhaustively list the features of the software and will be presented to a customer to seek their approval. It will also serve as a reference to the developers building the first version of the software.
This SRS Document is divided into two parts to provide the user with functional and nonfunctional requirements of the competitive programming helper app. The first section deals with gathering of various functional requirements and their hierarchy as concluded by an iterative process of speculating and contextual enquiries.
Furthermore, the second section deals with Contextual Inquiry and representation of our collective understanding of the user through Affinity Diagram.

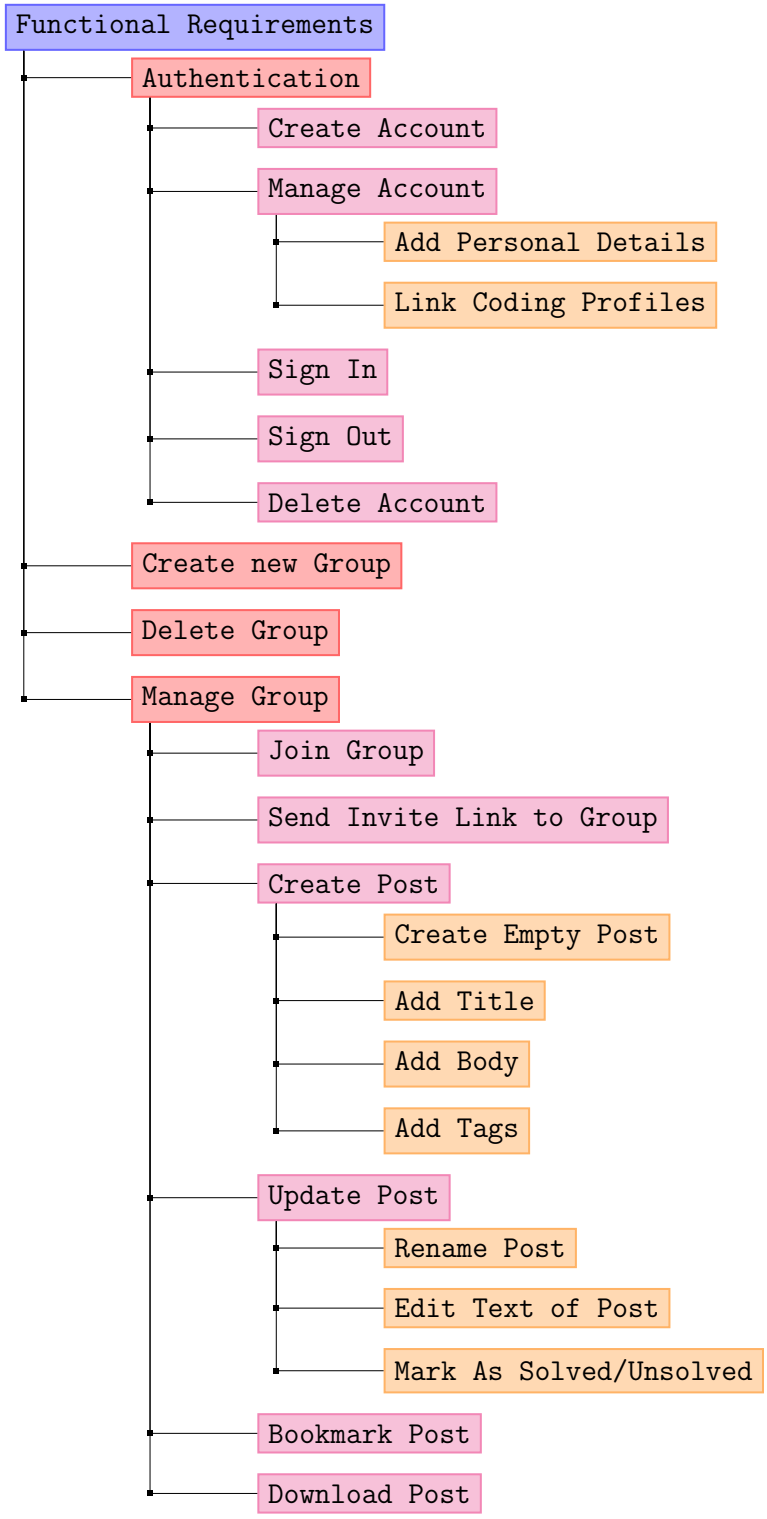
1.2 Problem Statement

Across engineering colleges of India, thousands of students take part in competitive coding contests and practice problems on data structures and algorithms. There are many resources available for students to get started and grow in this domain. However one important part - a centralised forum for doubt discussion - is missing. Contests occur 3-4 times a week across the various websites, and each contest has 4-6 problems. Very few solve all questions correctly, so there exist many who missed out on solving some question or grasping the concept behind some problem. Simply reading the correct intended solution is not enough and this is where one's seniors/friends come to help. A friend properly explains the solution along with prerequisites, alternate theories and shows where one might go wrong. But not everyone has such good friends. Hence we introduce such a forum which also has additional helping features which will lead to a programmer's growth.

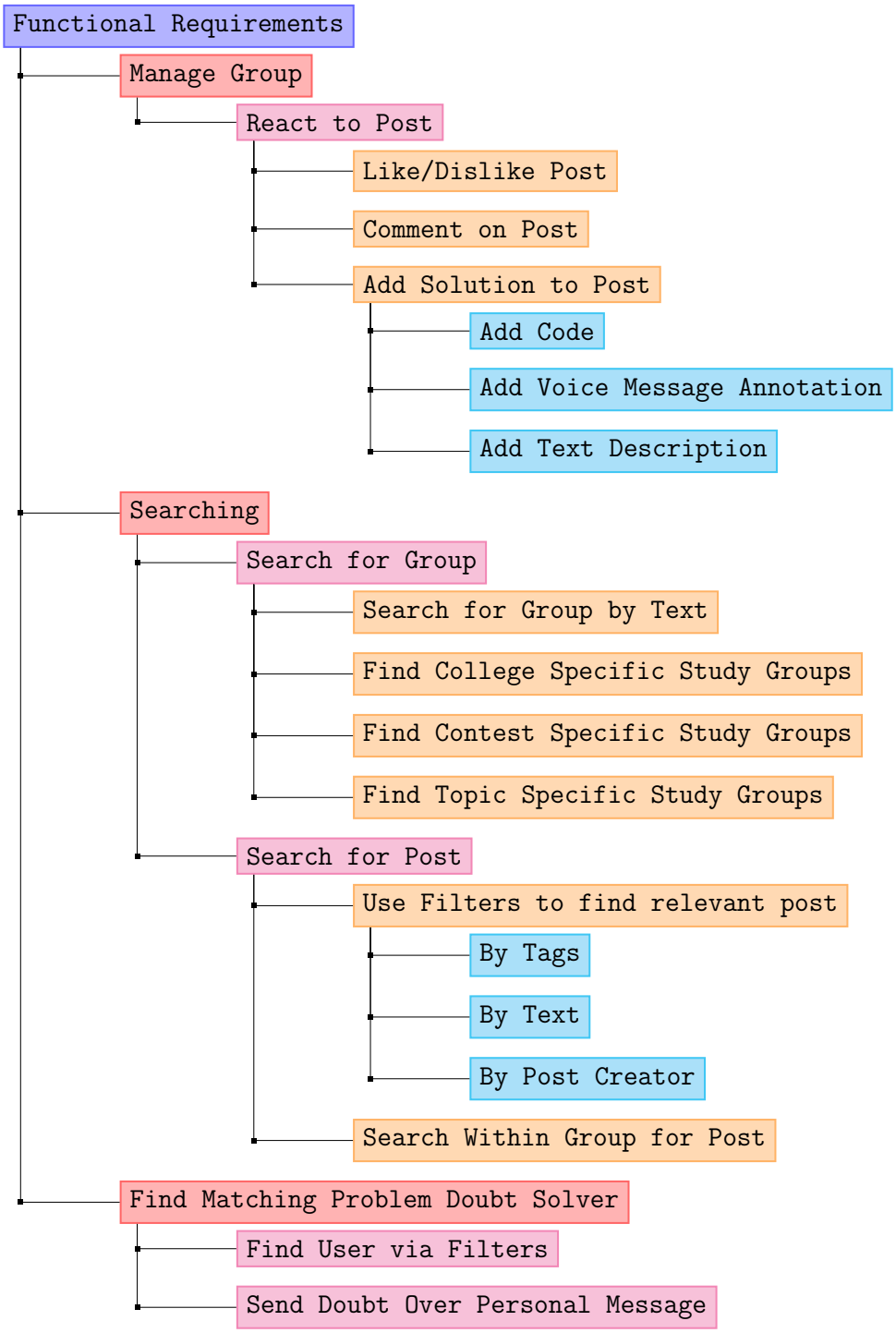
1.3 Scope of the System and Target Audience

The application is intended to be used by engineering college students, particularly those who are active on competitive programming platforms such as codeforces.com, atcoder.jp, codechef.com, etc. This app will serve as a centralised forum for discussions about coding problems and act as a resource repository.
The primary functionality is implemented via groups, posts, comments and solutions. The users can create groups based on their colleges, aptitudes, and areas of interest. Any doubts or problems that the users want discussed will be uploaded as posts, to which other users can add comments and solutions.

2 Functional Requirements Hierarchy



continued...


Functional Requirements
Manage Group
React to Post
Like/Dislike Post
Comment on Post
Add Solution to Post
Add Code
Add Voice Message Annotation
Add Text Description
Searching
Search for Group
Search for Group by Text
Find College Specific Study Groups
Find Contest Specific Study Groups
Find Topic Specific Study Groups
Search for Post
Use Filters to find relevant post
By Tags
By Text
By Post Creator
Search Within Group for Post
Find Matching Problem Doubt Solver
Find User via Filters
Send Doubt Over Personal Message

3 Functional Requirements

3.1 Authentication

3.1.1 Create Account

- **Input :** User Information
- **Output :** Created Account ID
- **Description :** Creates a new account using details provided and returns the unique ID of the account.

3.1.2 Manage Account

3.1.2.1 Add Personal Details

- **Input :** User Details
- **Output :** Account Content
- **Description :** Adds personal details such as date of birth, institute attended to the user's profile.

3.1.2.2 Link Coding Profiles

- **Input :** Hyperlinks of Coding Profiles
- **Output :** Account Content
- **Description :** Adds the links of the user's coding profiles to his account

3.1.3 Sign In

- **Input :** Account Credentials
- **Output :** Login Status
- **Description :** Signs the user into the account after verifying credentials

3.1.4 Sign Out

- **Input :** Account ID
- **Output :** Logout Message
- **Description :** Signs the User out of the website and displays Logout Page

3.1.5 Delete Account

- **Input :** Account Credentials
- **Output :** Success Message
- **Description :** Deletes the account and displays the Logout Page

3.2 Create New Group

- **Input :** Group Name, Account ID
- **Output :** New Group
- **Description :** Creates a new group with the given name and with the current account as its creator.

3.3 Delete Group

- **Input :** Group Name, Account ID
- **Output :** Success/Error Message
- **Description :** Deletes the group with given name if the current account is the same as its creator.

3.4 Manage Group

3.4.1 Join Group

- **Input :** Group Name, Account ID
- **Output :** Group Members List
- **Description :** Adds the account ID to the list of Group Members and allows access to group.

3.4.2 Send Invite Link to Group

- **Input :** Group Name
- **Output :** Hyperlink
- **Description :** Generates a hyperlink to be used by other users for joining the group.

3.4.3 Create Post

3.4.3.1 Create Empty Post

- **Input :** Group Name, Account ID
- **Output :** New Post, Post ID
- **Description :** Creates an empty post in the required group, and registers its creator.

3.4.3.2 Add Title

- **Input :** Post ID, Text String for Title
- **Output :** Post Title
- **Description :** Gives the post the required title.

3.4.3.3 Add Body

- **Input :** Post ID, Text Body

- **Output :** Post Content
- **Description :** Adds text passed as the post content of the empty post.

3.4.3.4 Add Tags

- **Input :** Post ID, List of Text Tags
- **Output :** Post Tags
- **Description :** Adds tags to the required post.

3.4.4 Update Post

3.4.4.1 Rename Post

- **Input :** Post ID, Account ID, Text string
- **Output :** Post Title
- **Description :** Changes the title of the post if the user matches with the creator.

3.4.4.2 Edit Text of Post

- **Input :** Post ID, Account ID, Text Body, Location
- **Output :** Post Content
- **Description :** Modifies/Adds the required text at the specified location in the post body.

3.4.4.3 Mark As Solved/Unsolved

- **Input :** Post ID, Account ID, Boolean Flag
- **Output :** Post Status
- **Description :** Marks the post as solved or unsolved depending upon the value of the bool flag. Note that like all functions above, it will only run successfully if the current user ID matches with the creator ID.

3.4.5 Bookmark Post

- **Input :** Post ID, Account ID, Boolean Flag
- **Output :** Post Bookmark Status for current User
- **Description :** Marks the post as bookmarked for the current user only. Note that for bookmarking, the user does not need to be the creator since the change is only reflected locally.

3.4.6 Download Post

- **Input :** Post ID, Account ID, Boolean Flag
- **Output :** Post Downloaded Status for current User
- **Description :** Saves the entire discussion post on local device and marks the post as downloaded for the current user only.

3.4.7 React to Post

3.4.7.1 Like/Dislike Post

- **Input :** Post ID, Boolean Flag
- **Output :** Post Like Status
- **Description :** Likes/dislikes the post depending upon the flag value, and increments/decrements the Post Like Status globally.

3.4.7.2 Comment on Post

- **Input :** Post ID, Comment Text, Account ID
- **Output :** New Comment and List of Comments
- **Description :** Creates a new comment and adds it to the list of comments linked the post. Account ID of the commenter is registered.

3.4.7.3 Add Solution to Post

3.4.7.3.1 Add Code

- **Input :** Post ID, Account ID, Code text
- **Output :** New Code Solution and List of Solutions
- **Description :** Takes the input code and creates a new code solution which is added to the list of solutions. Account ID of the solver is registered against the solution.

3.4.7.3.2 Add Voice Message Annotation

- **Input :** Post ID, Account ID, Audio file
- **Output :** New Audio Solution and List of Solutions
- **Description :** Creates an audio solution using the audio file and adds it to the list of solutions.

3.4.7.3.3 Add Text Description

- **Input :** Post ID, Account ID, Text Body
- **Output :** New Text Solution, List of Solutions
- **Description :** Creates a text solution using the text body and adds it to the list of solutions.

3.5 Searching

3.5.1 Search For Group

3.5.1.1 Search For Group by Text

- **Input :** Text Phrase
- **Output :** List of Groups
- **Description :** Returns all groups which have the given text phrase in their titles.

3.5.1.2 Find College Specific Study Groups

- **Input :** College Name(Text Phrase)
- **Output :** List of Groups
- **Description :** Returns all groups which are specific to the required college.

3.5.1.3 Find Contest Specific Study Groups

- **Input :** Contest Name(Text Phrase)
- **Output :** List of Groups
- **Description :** Returns all groups which are created for the required contest.

3.5.1.4 Find Topic Specific Study Groups

- **Input :** Topic Name(Text Phrase)
- **Output :** List of Groups
- **Description :** Returns all groups which are specific to the required topic.

3.5.2 Search for Post

3.5.2.1 Use Filters to find relevant post

3.5.2.1.1 By Tags

- **Input :** List of Tags
- **Output :** List of Posts
- **Description :** Returns a list of posts which have the required tags.

3.5.2.1.2 By Text

- **Input :** Text Phrase
- **Output :** List of Posts
- **Description :** Returns a list of posts whose titles contain the required phrase.

3.5.2.1.3 By Post Creator

- **Input :** Creator ID
- **Output :** List of Posts
- **Description :** Returns a list of posts which were created by the user with the given creator ID.

3.5.2.2 Search Within Group for Post

- **Input :** Group ID, Text Phrase
- **Output :** List of Posts
- **Description :** Returns a list of posts posted in the given group, and having the required text phrase in their title.

3.6 Finding Matching Problem Doubt Solver

3.6.1 Find User via Filters

- **Input :** Filters such as College, Rating, etc.
- **Output :** List of Matching Users
- **Description :** Returns a list of potential doubt solvers based on criteria defined by user.

3.6.2 Send Doubt Over Personal Message

- **Input :** User and Message
- **Output :** Message sent to selected user
- **Description :** Sends a personal message containing specific doubt to selected users.

4 Contextual Inquiry

4.1 Method Of Conducting

All three members have participated in many competitive programming contests over the period of two years. Furthermore, there is a diversity of expertise in all three members. One only used it for few months for the purpose of securing internship, one gave contests infrequently and one regularly participated in contests. Thus this diversity is represented in the thoughts expressed in affinity diagram.
Group members secured good summer internships by using such platforms for practicing problems effectively. Broad ideas were gathered and then sorted. Usability requirements were extracted and converted to functional requirements.

4.2 Affinity Diagram

4.2.1 Initial Ideas Gathered

4.2.2 Ideas Sorted

4.3 Usability Requirements gathered from Affinity Diagram

1. **Add Voice Message Annotation** - F.3.7.3.2
   Users explained that just the code and text content is not enough for understanding. Additional cues such as audio descriptions and images aid in understanding concepts.
2. **Create Group** - F.3.2
   Finding an already existing community is sometimes difficult so the option to create a group provides users with a chance to establish one.
3. **Send doubt over personal message** - F.3.6.2
   Users want to connect with other users and then use this feature to request solutions and answers in a one-on-one manner.
4. **Bookmarking** - F.3.4.5
   Users wanted to create a comprehensive list of important questions and discussions for future reference.
5. **Use filters to find relevant post** - F.5.2.1
   For ambiguous queries, filter out the discussions by topic and then look for a better result. These also help users to find multiple problems fitting some single criteria.
6. **Download post** - F.3.4.6
   Users may eliminate the use of internet by downloading some discussions they might want frequently.