

# Requirement Document

---

## Project Name - Coding in Circles

---

## Project Phase 1

---

### Team 1 Data-Analyst

- Aakash Dantre - 2018101039
- Varun Changani - 2019121011
- Shivaan Sehgal - 2018111026

## Introduction

---

### Description of Mini-world

Online Judges generally involves the host presenting a set of logical or mathematical problems to the contestants and they are required to write computer programs capable of solving each problem. Judging is based mostly upon number of problems solved and time spent for writing successful solutions, but may also include other factors (quality of output produced, execution time, program size, etc.)

Problems are added by users only and they are graded by administrative users.

A user can use the database for

- start learning a new programming language.
- keep track of problems he have solved.
- share his knowledge by creating problems and contest.
- improve coding skills.

### Purpose

Purpose is to motivate students to take interest in programming, being able to write solution of given problems and to help other by creating them.

### Intended audience /user

There are multiple type of user, each having their own functionality .For example, a guest can view questions, contest and performance of other user, a user have access to all contest created by himself and the admin have access to all the contest and so on.

## Database Requirements

---

### Entities

#### All Entities

- User
  1. user id - Primary Key

- 2. Name - composite key
  - First Name
  - Middle Name
  - Last Name
- 3. Date of Birth
- 4. Primary Email ID
- 5. Secondary Email IDs - multivalued attribute
- 6. Institute
- 7. Join Date(datetime)
- 8. Age - Derived Attribute
- Question
  - 1. Question ID - Primary Key
  - 2. Title
  - 3. Question text
  - 4. Score
  - 5. Allowed Languages
  - 6. Tags
- Contest
  - 1. Contest ID - Primary Key
  - 2. Contest Name
  - 3. Creation Date
  - 4. Starts at Time
  - 5. Ends at Time
  - 6. Minimum Score allowed
  - 7. Maximum Score allowed
- Test Case
  - 1. Test case ID - Primary Key
  - 2. Input
  - 3. Expected Output
  - 4. Subtask ID
  - 5. Score
- Submission
  - 1. Submission ID - Primary Key
  - 2. Submission Time
  - 3. Code
  - 4. Language
- Result
  - 1. result value
  - 2. evaluation date
- Administrator User
  - 1. Category

## Weak Entities

- Submission - Depends on User and Question. Cannot exist until both exist
- Testcase - Depends on Question. Cannot exist until a question exists corresponding to it.
- Results In - Depends on Solution and its testcase. Cannot exist until both exist

## Relationships

- User - Contest (Creates)
- User - Contest (Participates in)
- User - Question (Forms)
- User - Question - Submission (Solves)
- Contest - Question (Has)
- Question - Testcase (Has)
- Solution - Testcase - Result (Results in)

## n>3 Relationship

- **Solves** Relationship is a Tertiary Relation
- **Results In** Relationship is a Tertiary Relation

## Subclass

- Entity Administrator User is a Subclass of entity User.

## Functional Requirements

---

### Functionalities provided to a guest

- can view other users profile.
- see questions title, text, allowed language and leaderboard.

### Additional functionalities to a user

- Signup/Login for User
- User can add a problem
- User can delete a problem created by himself
- User can add a contest
- User can delete a contest created by himself
- User review
- Rating may increase or decrease based upon performance
- Contest will have different divisions
- Problems will have several testcases for corresponding subtasks for partial marking
- User can update his user information like name, dob, institute
- user can submit code for evaluation
- User can see their progress over time

### Additional functionalities for admin

- Admin user can delete a problem or a contest
- Login for admin
- Admin rates the problem/contest added by the user