



**Department of Computer Engineering
Faculty of Technology
Dharmsinh Desai University
Nadiad.**

B.Tech. CE Semester-VI

Subject: System Design Practice

Project Title: Coding Competition System

By

- 1) Savan Chovatiya - CE 19 (ID:16CEUOG057)**
- 2) Darshan Lukhi - CE 56 (ID:16CEUOS101)**
- 3) Ishan Ghelani - CE 32 (ID:16CEUOG051)**

Guided by:

**Prof. Mrudang T. Mehta
Associate Professor,
Dept Of Computer Engg.**

CERTIFICATE

This is to certify that the project work titled

Coding Competition System

is the bonafide work of

Savan Chovatiya - CE19 (ID 16CEUOG057)

Darshan Lukhi - CE56 (ID 16CEUOS101)

Ishan Ghelani - CE32 (ID 16CEUOG051)

carried out in the subject of System Design Practice and Semester-VI
of the degree of Bachelor of Technology in Computer Engineering at
Dharmsinh Desai University in the academic session

December 2018 to April 2019.

Prof. Mrudang T Mehta,
Associate Professor,
Dept. of Computer Engg.

Dr. C.K. Bhensdadia,
Head,
Dept. of Computer Engg.



Department of Computer Engineering
Faculty of Technology
Dharmsinh Desai University
April 2019

Table of Contents

Chapter		Page
1.	Introduction	5
2.	Software Requirements Specification	6
	2.1 Introduction	6
	2.2 Overall Description	7
	2.3 Functional Requirement	8
	2.4 Non-Functional Requirement	11
3.	Design:	12
	3.1 Use-case Diagram	12
	3.2 Class Diagram	13
	3.3 Sequence Diagram	14
	3.4 State Diagram	15
	3.5 E-R Diagram	16
	3.6 DFD (Data Flow Diagram)	17
	3.7 Data Dictionary	19
	3.8 Development Architecture	22
	3.9 System Architecture	22
4.	Implementation Detail	23
5.	Testing	24
6.	Screenshots	27
7.	Conclusion	31
8.	Limitation and Future Extensions	32
9.	Bibliography	33

Abstract

“Coding Competition System” this system is generally used by some of users like Admin and Student, professor and coding lover people. This system should allow user to participate in contest, compile and run code online, solve practice problem, ask doubts worldwide and also answers to others. Also, admin can start contest and add new problems etc. By using this web application user can measure their coding power. For the implementation of this project, we have use Node.js and Express framework as our backend and Angular as frontend. we have used MongoDB as our database. Mongo Compass and Robo3T is intended to handle the administration of MongoDB. We have used VS Code which is the IDE for developers. We have selected VS Code because it provides all the tools that are required for productive MEAN (Mongo, Express, Angular4, NodeJS) development. Development tools used during complete implementation of this project are VS Code, Umlet.

1.1 Brief Overview

Coding Competition System is a web application which is developed with the purpose of learning of programming language and increase logic by hard problems. With this web application any developer or programmer can participate in contest or they can solve practice example. Users can also ask question worldwide if they have any doubt. And also, can give answer to another user and help them.

1.2 Technology Used

Front End: HTML, CSS, Bootstrap, JavaScript, jQuery and Angular

Back End:

Node.js & ExpressJS: We use node.js for manage the request come from client side so basically work as a server.

Database:

MongoDB: We use mongodb for handle data of users and other. MongoDB is better to store large data.

Development Tool:

VS Code: Develop Code

Mongo Compass: Handle Database

Diagram Tool: Umlet

2.1 Introduction

2.1.1 Problem Definition

Coding Competition System is a system used to test correctness and efficiency of programs written in various programming languages. They are used on programming competition held. Most importantly they are used to practice for a programming contest. These systems are used by lots of people, particularly students, to learn issue related to programming languages and to improve programming skills through training and competing in several programming contests. Furthermore, they have been used by recruiters of the well-known companies for job applications.

In this project, we are planning to build an environment that will help users to interact, practice and improve their programming skills via interaction, and organizing online competitions.

2.1.2 Purpose

The purpose of this document is to describe the software requirements for the Coding Competition System. It is important that these requirements are reached in order to answer all the expectations. This document uses written descriptions as well as modeling diagrams to illustrate the high-level structure of the software. Furthermore, this document aims to give a necessary view of requirements for each stakeholder considering their responsibilities.

2.1.3 Scope

Coding Competition System aims to build an environment for the people, specifically students, around the world to interact, practice and improve their programming skills. It is designed to organize programming contests in a fair and secure way. Moreover, it provides an environment for practicing programming languages.

2.1.4 Feature

- Automatic test case checking.
- Ranking feature based on the maximum solved problem.
- The user can also view all last submission with status.
- Compiler for compile any code online.
- View anyone profile.
- Previous Contest Problem for practice.
- Practice problem in various category like Easy, Medium, Hard, Challenge etc.
- Question-answer feature
- Contest Setting feature
- Login and Register

2.2 Overall Description

2.2.1 Product Perspective

Coding Competition System consists of a database server which contains data for users and the problems added by Admins.

2.2.2 User Classes and Characteristics

This system can be used by the User and the admin of the system. The User are mainly student, teachers and people interested in coding. The user can log in, participate in the contest, update profile, view last submission, view ranking list for the contest. Admin can set contest by adding contest problems.

2.2.3 Operating Environment

This System can work good in windows operating system. The system is open source for use.

2.2.4 Assumptions and Dependencies

The user must log in for participating in the contest, update profile, view last submission, view ranking list for contest etc. Admin must log in for set the contests. The user can view practice problems and contests list without login but cannot participate in the contest.

2.3 Functional Requirements

R1: User Management: -

R.1.1 Registration; -

I/P: - User Details

O/P: - If user registration is successful then home page is displayed otherwise registration page with error is displayed.

Process: -Verification, Validation for entered details.

R.1.2 Login: -

I/P: - User Id, Password.

O/P: - If user registration is successful then requested page is displayed otherwise registration page with error is displayed

Process: -Verification, Validation, and authentication.

R.1.2.1: -Forgot Password

I/P: - Email Verifications.

O/P: - Message displayed that password reset successfully if successfully reset
Otherwise, an error is displayed with that page.

Process: -Credential are checked for a password.

R2: Profile Management: -

R.2.1 View Profile: -

I/P: -Request for view profile

O/P: -If a user is logged in then profile page is displayed with personal details, recent activity.

Process: -Checking for login and fetch data from the database and display it.

R.2.2 Update Profile: -

I/P: -Request for update profile

O/P: -If a user is logged in then update profile page is a display where the user can update personal, professional and additional details.

Process: -Entered details is updated the database.

R.2.3 View all submissions: -

I/P: -Request for view my submission

O/P: -If the user is logged in then all previous submission is displayed for practice problems and contest problem.

Process: -All data for user's submission is fetched from the database.

R3: Contest Management: -

R.3.1 Participate in the contest: -

I/P: -Select particular contest from the list.

O/P: - If the user is logged in then contest page is displayed with problem description otherwise, login page will display.

Process: -Problem's file name will be fetched from database and display description from files.

R.3.2 Submit solution: -

I/P: -Solution for a particular problem in any language which supported.

O/P: -Page with solution status will be displayed and also test cases passed or not will display.

Process: -Test cases will pass to the solution and store solution with status in the database.

R.3.3 View Rank list: -

I/P: -Select contest from list

O/P: -Rank list page will display for the particular contest based on the maximum problem Solved with Username and which problem solve.

Process: -Data from the database will be fetched for the contest with username and Solved details for user.

R.3.4 View Contest List: -

I/P: -Solution for a particular problem in any language which supported.

O/P: -The page with past, current, and future events is display.

Process: -The contest details for all contests are fetched from the database.

R4: Discuss Management: -

R.4.1 View Question-Answer List: -

I/P: -Request for this page.

O/P: -The page with the question of users is displayed with the answer given by user.

Process: -This list is fetched from files which locations are stored in the database.

R.4.2 Ask Question: -

I/P: -Question inappropriate way.

O/P: -Some message that question submitted successfully.

Process: -Question is stored into database and display into discuss part.

R.4.3 Give Answer: -

I/P: -Answer for any question.

O/P: -Message displayed that answer is submitted successfully otherwise an error message.

Process: -Answer is stored with that question id in the database.

R.4.4 View your Question: -

I/P: -Select view question option from the profile list.

O/P: -The page with user question and answer for question given by another user is displayed.

Process: -Question are loaded from database based on asked by user.

R.4.5 View your Answer -

I/P: -Select view my answer option from the profile list.

O/P: -The page with user's question for which user posted answer.

Process: -Questions are loaded from database based on answer given by user.

R5: Practice Management: -

R.5.1 Open Problem List: -

I/P: -Select category.

O/P: -Based on the category problems list.

Process: -This list is fetched from files which locations are stored in the database.

R.5.2 Open Problem Description: -

I/P: -Select a problem from the problem list.

O/P: -Problem description with input, output, constraints.

Process: -Problem description file is fetched from the appropriate location using the database.

R.5.3 Submit solution: -

I/P: -Solution for a particular problem in any language which supported.

O/P: -Page with status will be displayed and also test cases pass or not will display.

Process: -Test cases will pass to the solution and store solution with status in the database.

R6: Compiler Management: -

R.6.1 Compile code: -

I/P: -Select language and upload solution.

O/P: -Appropriate output is displayed for input with success or error message.

Process: - Code is compile on server and output is return.

R7: Admin Management: -

R.7.1 Create/Add contest: -

I/P: -Enter contest name, unique ID.

O/P: -Page for adding problem is displayed.

Process: -Validation apply then information is stored into the database.

R.7.2 Add problem: -

I/P: -Enter name, unique problemCode and file description.

O/P: -Problem will add into the contest and page is again displayed.

Process: -Validation apply and the problem is stored to the database with contest ID.

R.7.3 Start contest: -

I/P: -Select contest from the list and select start contest option.

O/P: -Appropriate message is displayed.

Process: -For started contest state is changed into a database for that contest.

R.7.4 End contest: -

I/P: -Select contest from list and select end contest option.

O/P: -Appropriate message is displayed.

Process: -For ended contest state is changed into a database for that contest.

R.7.5 Manage contest: -

I/P: -Select contest from the list and select delete or update contest option and enter new details for update contest.

O/P: -The page is displayed for contest list with updated detail.

Process: -For contest record is deleted or updated into a database after validation.

R.7.6 Manage problem: -

I/P: -Select problem from the list for the contest and select delete or update contest option and enter new problem description for the update problem.

O/P: - The page is displayed for problem list with updated detail.

Process: -For contest record is deleted or updated into database for that contest after validation.

2.4 Non-Functional Requirements

1) Usability

As far as possible, the system should provide a simple, responsive interface. Although Coding Competition System is composed of diverse systems, applications, and processes, the underlying architecture should be transparent to the administrators. The system should be configurable from a single source at a central administrative position and Should provide a central, easy-to-use interface that will allow administrators to configure the user interface and features in a way that reduces page clutter.

2) Reliability

The system must be backed up on a configurable schedule Back-up requirement will need to be determined, based on individual components of the system. The system should be backed up with a frequency that ensures system and all data is protected. Since the updates and changes will be done to the database via web-interface, it should be backed up on a nightly basis, with options for weekly, as well as an off-site backup when necessary. The system must have the ability and capacity to restore back-up data within five hours so that the system is never offline for an inordinate period of time.

3) Performance

The performance is good of this system for user by availability, good database management and good user interface. So, while using this system user looks good for its user-friendly frond-end and back-end of the system.

4) Security

The system must comply with the permission roles Security is the most important attribute of the system. The system should not allow unauthorized accesses. It should not mix the roles of the users. For instance, contestants should not be allowed to reach the solutions of the problems or access to the problem test data. the system should prohibit the actions that are not permitted for a user.

Use Case Diagram

Use Case diagram for Coding Competition System is below with functionalities and following purposes.

- Used to gather the requirements of a system.
- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Show the interaction among the requirements are actors.



Figure 1: use case diagram

Class Diagram

Class diagram for Coding Competition System is below for this purpose.

- Analysis and design of the static view of an application.
- Describe the responsibilities of a system.
- A Base for component and deployment diagrams.

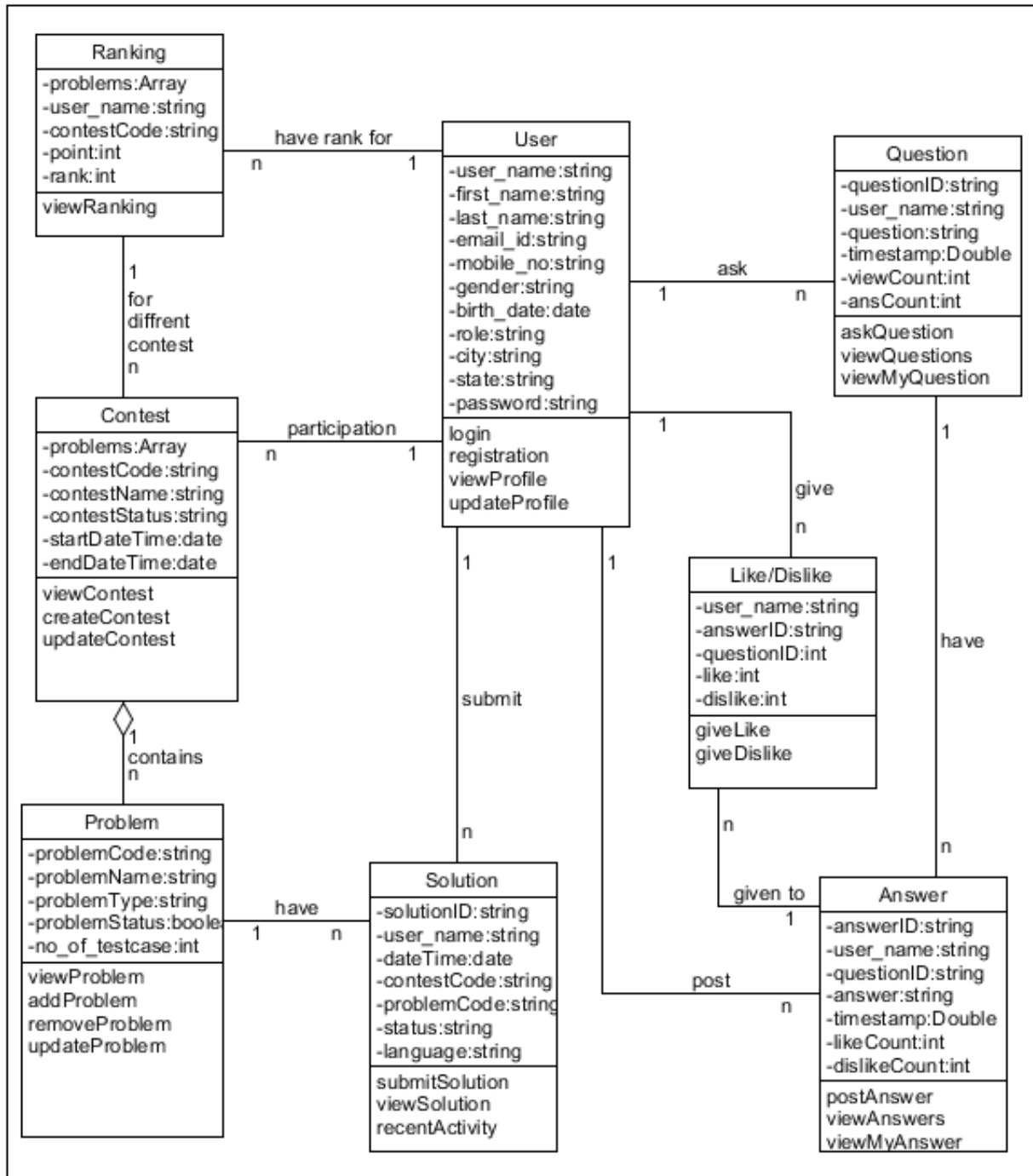


Figure 2: class diagram

State Diagram

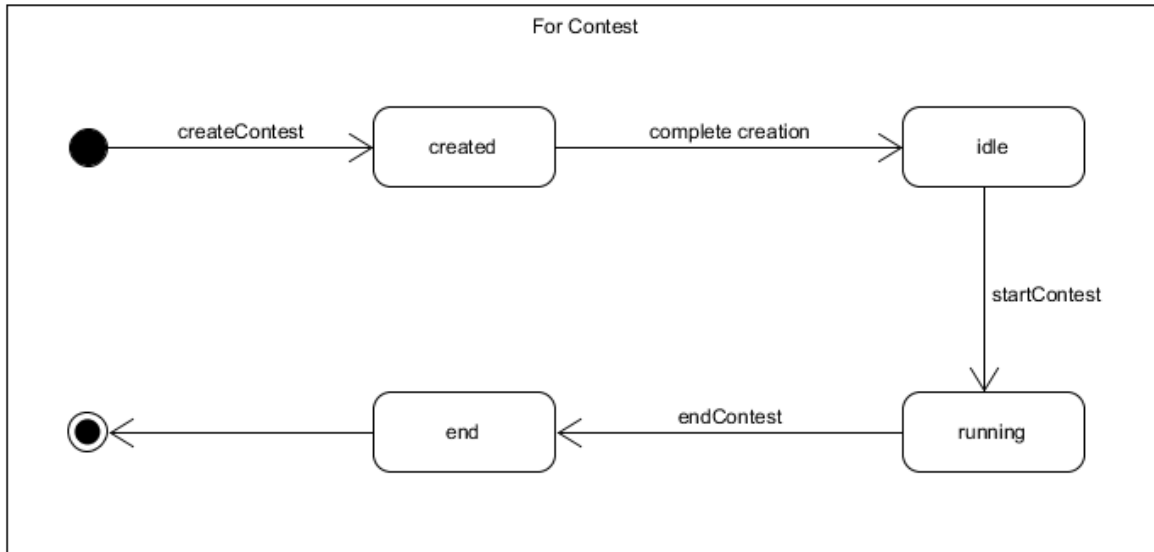


Figure 3: state diagram

Sequence Diagram for User Profile

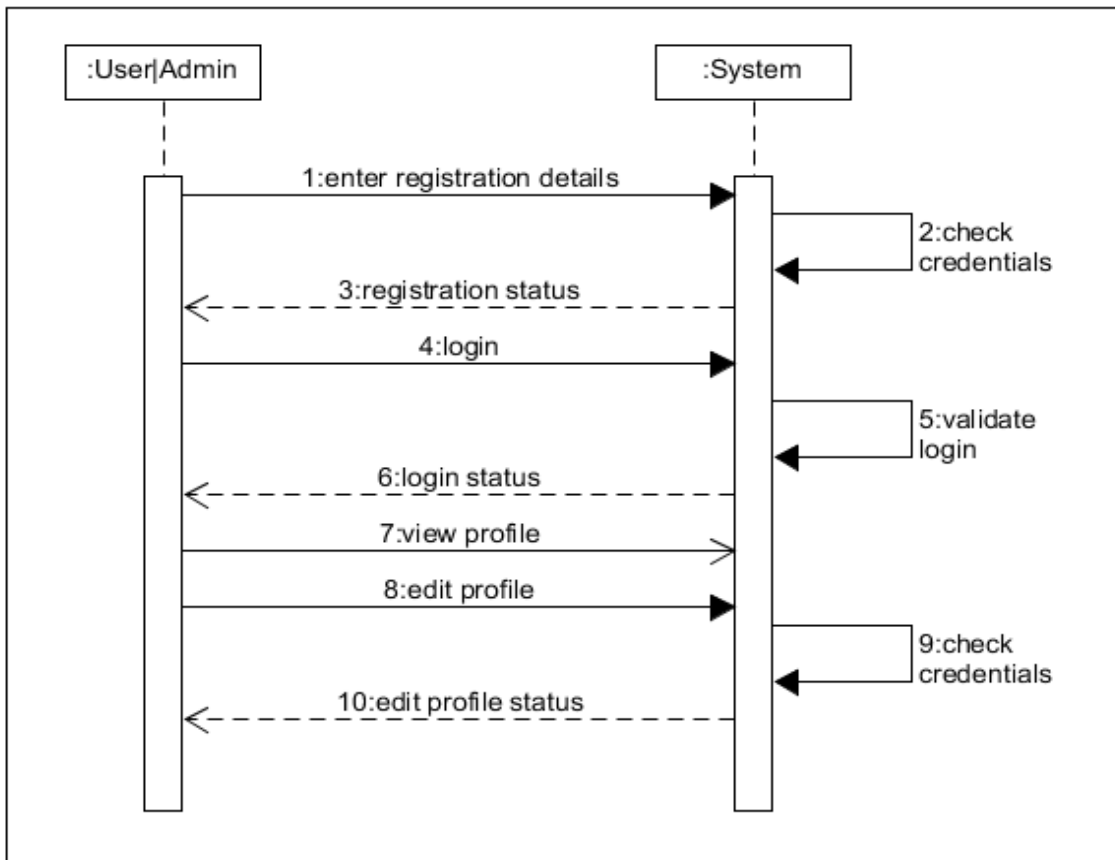


Figure 4: sequence diagram for user profile

Sequence Diagram for Contest

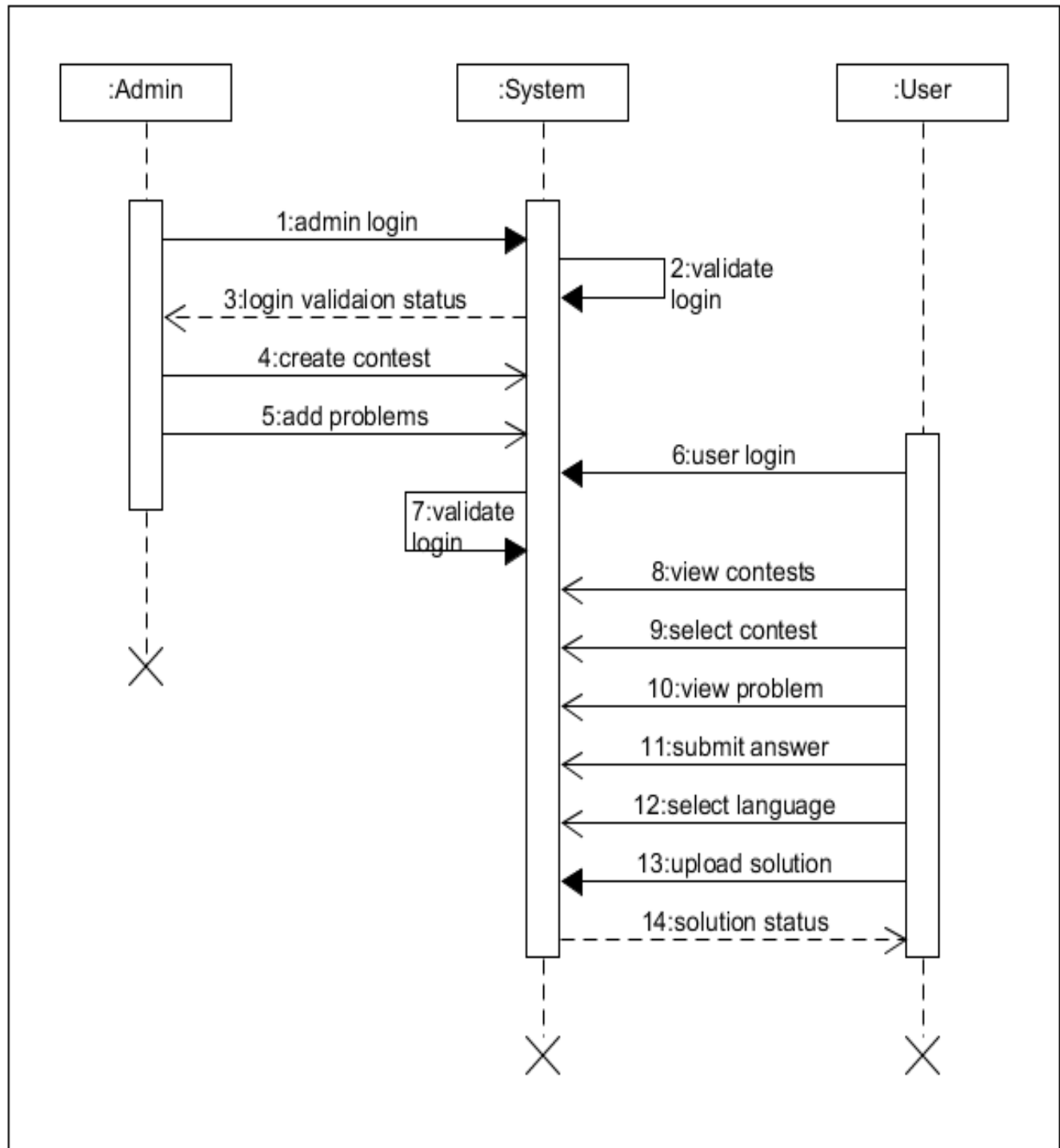


Figure 5: sequence diagram for contest

Entity Relationship Diagram

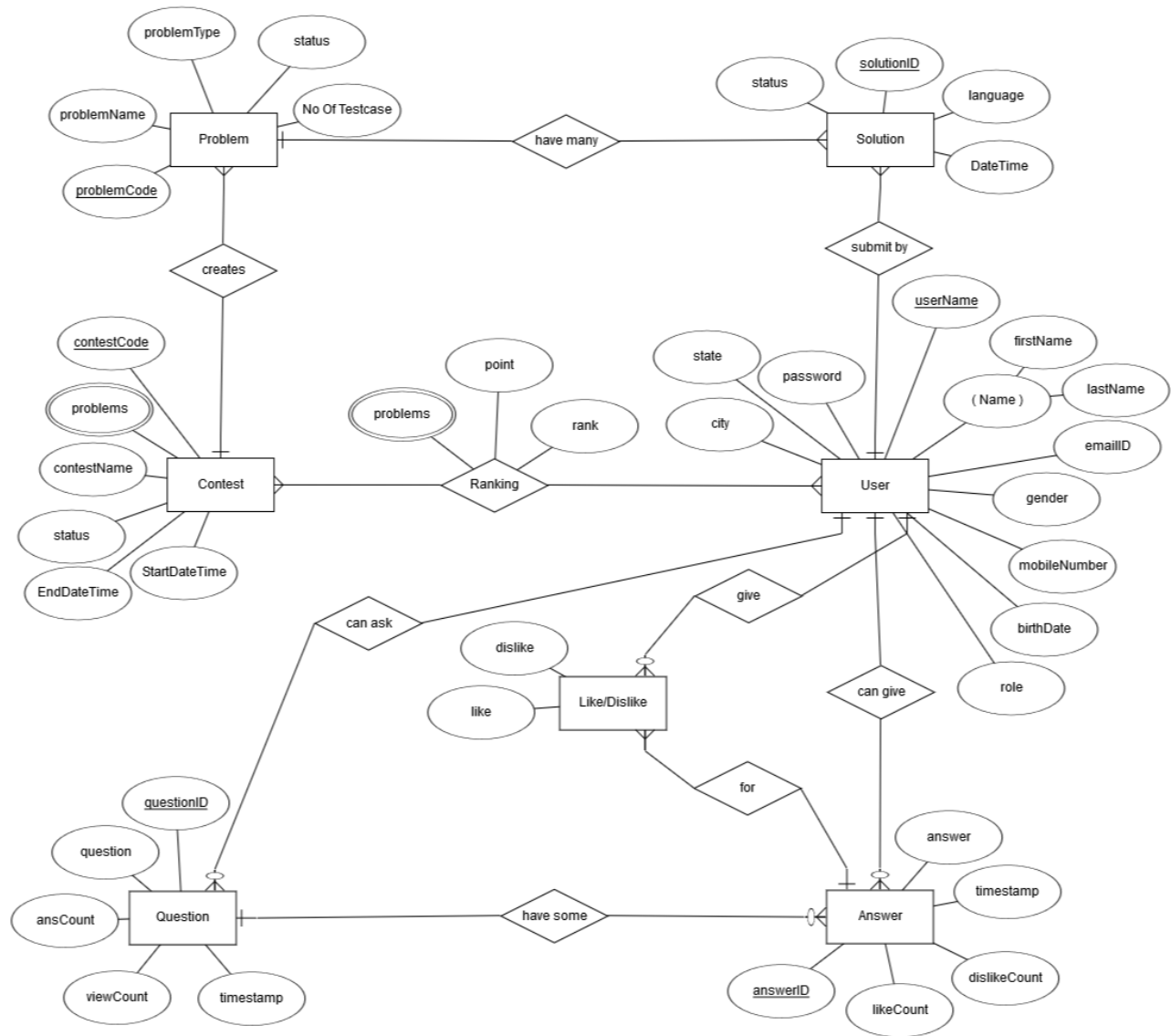


Figure 6: ER diagram

Data Flow Diagram DFD Level-0

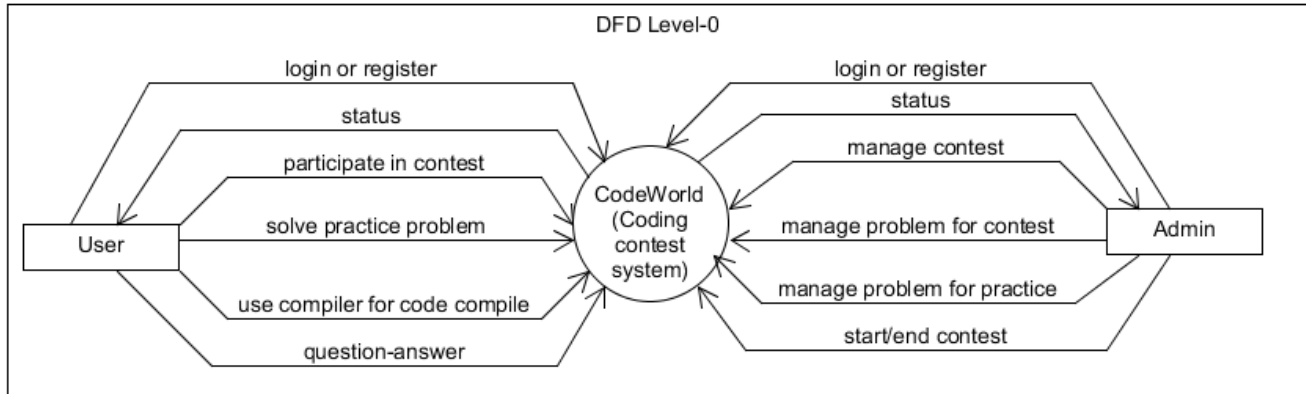


Figure 7: DFD level 0

DFD Level-1

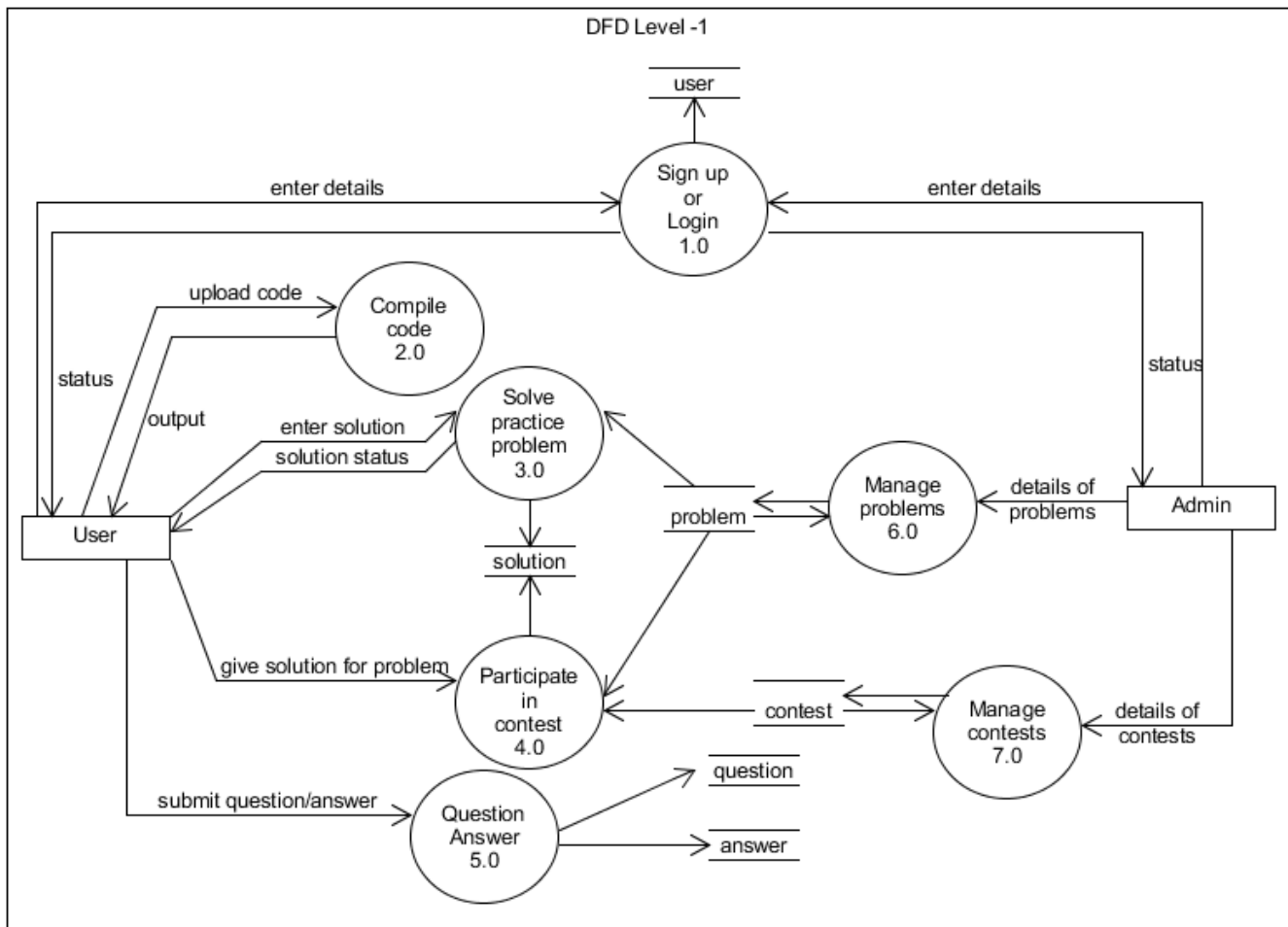


Figure 8: DFD level 1

DFD Level-2

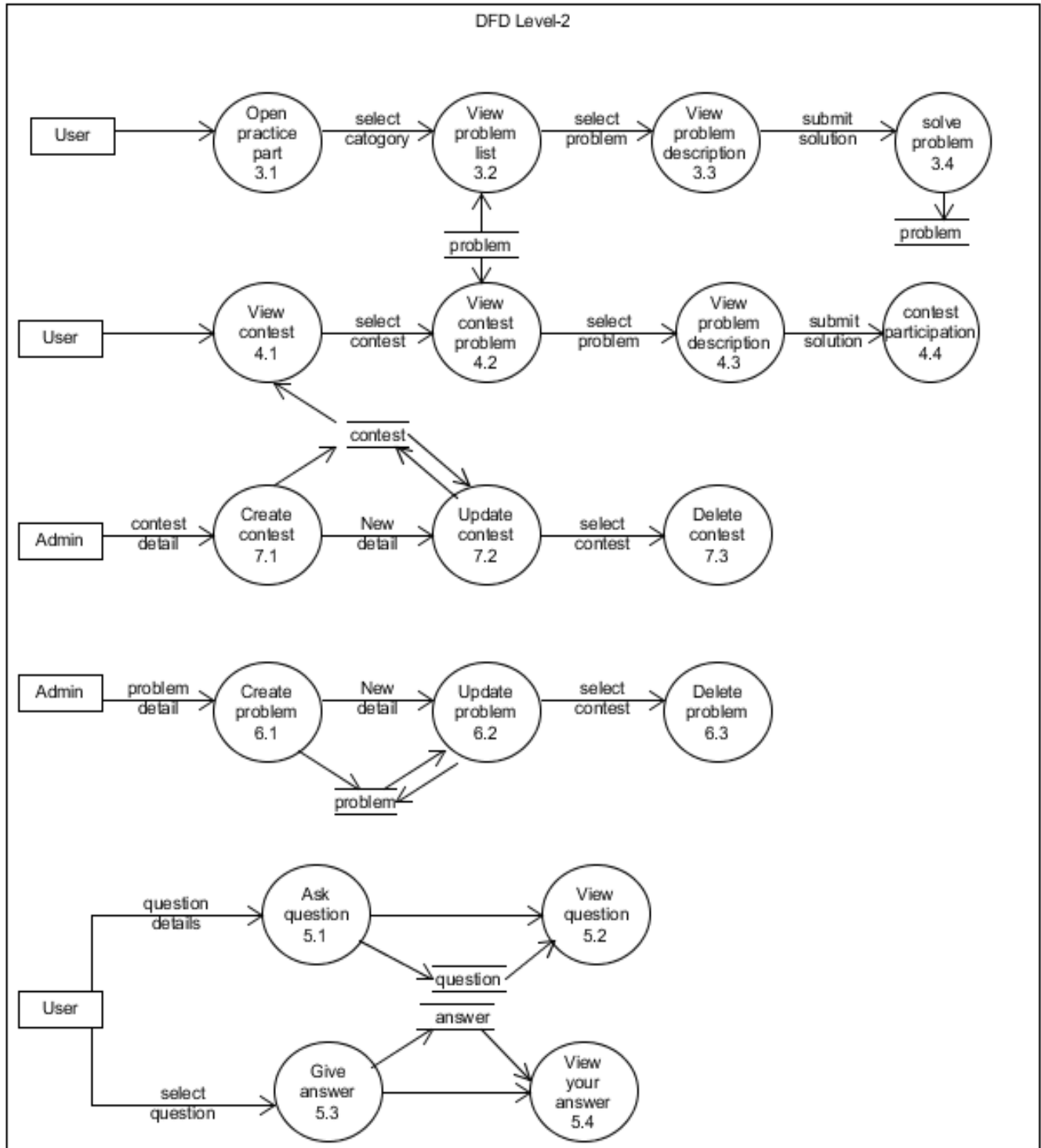


Figure 9: DFD level 2

Data Dictionary

User Table

Field Name	Data Type	Constraints	Description
user_name	String	Primary Key	Id of user
first_name	String	Not Null	First name of user
last_name	String	Not Null	Last name of user
email_id	String	Not Null	Email Address
mobile_no	String	Not Null	Mobile Number
gender	String	Not Null	Gender
birth_date	Date	Optional	Birth Date
role	String	Not Null	User by default
city	String	Optional	City
state	String	Optional	State
password	String	Not Null	password

Table 1: Data Dictionary for User Table

Contests Table

Field Name	Data Type	Constraints	Description
Problems	Array	Not Null	Codes of all problem in contest
contestCode	String	Primary Key	Unique Contest code
contestName	String	Not Null	Name of the Contest
contestStatus	String	Not Null	Status like publish or not publish
startDateTime	Date	Not Null	Contest start time
endDateTime	Date	Not Null	Contest end time

Table 2: Data Dictionary for Contests Table

Problems Table

Field Name	Data Type	Constraints	Description
problemCode	String	Primary Key	Code of the problem
problemName	String	Not Null	Name of the problem
problemType	String	Not Null	Easy, Medium, Hard etc.
problemStatus	Boolean	Not Null	False by default
No_of_testcases	Int32	Not Null	How many test cases created for this problem

Table 3: Data Dictionary for Problem Table

Solution Table

Field Name	Data Type	Constraints	Description
_id	String	Primary Key	Solution id, Auto Generated
user_name	String	Not Null	Id of user who submitted solution
DateTime	Date	Not Null	Time of submit, Auto Generated
contestCode	String	Not Null	Code of contest
problemCode	String	Not Null	Code of Problem
status	String	Not Null	TLE, WA, NZEC, AC etc.
language	String	Not Null	Programming Language used for writing solution code

Table 4: Data Dictionary for Solution Table

Questions Table

Field Name	Data Type	Constraints	Description
_id	String	Primary Key	Question Id, Auto Generated
userId	String	Not Null	Id of user who asked question
question	String	Not Null	Question asked by user
timeStamp	Double	Not Null	Time of record insertion, Auto Generated
viewCount	Int32	Not Null	0 initially, increase when any user opens question to view
ansCount	Int32	Not Null	0 initially, increase when any user gives answer for that question

Table 5: Data Dictionary for Question Table

Likes Table

Field Name	Data Type	Constraints	Description
_id	String	Primary Key	Like or dislike Id, Auto Generated
userId	String	Not Null	Id of the user who given like
questionId	String	Not Null	Question Id for given Like
Like	Int32	Not Null	1 if like, 0 if dislike
Dislike	Int32	Not Null	0 if like, 1 if dislike

Table 6: Data Dictionary for Likes Table

Answer Table

Field Name	Data Type	Constraints	Description
_id	String	Primary Key	Answer Id, Auto Generated
userId	String	Not Null	Id of the user who given answer
questionId	String	Not Null	Question id for given Answer
answer	String	Not Null	Answer provided by user
timestamp	Double	Not Null	Time of record insertion, Auto Generated
likeCount	Int32	Not Null	Value 0 initially, increase when any user likes
dislikeCount	Int32	Not Null	Value 0 initially, increase when any user dislikes

Table 7: Data Dictionary for Answer Table

Rankings Table

Field Name	Data Type	Constraints	Description
_id	String	Primary Key	Auto Generated
problems	Array	Optional	Id of Problems solved by user
User_name	String	Not Null	Participated user in contest
contestCode	String	Not Null	Code of contest
Points	Int32	Not Null	0 initially, increase as per no. of problem solved and type.
rank	Int32	Not Null	Rank of the user in contest

Table 8: Data Dictionary for Ranking Table

Development Architecture

The development architecture for Coding Competition System is like below. It contains below parts.

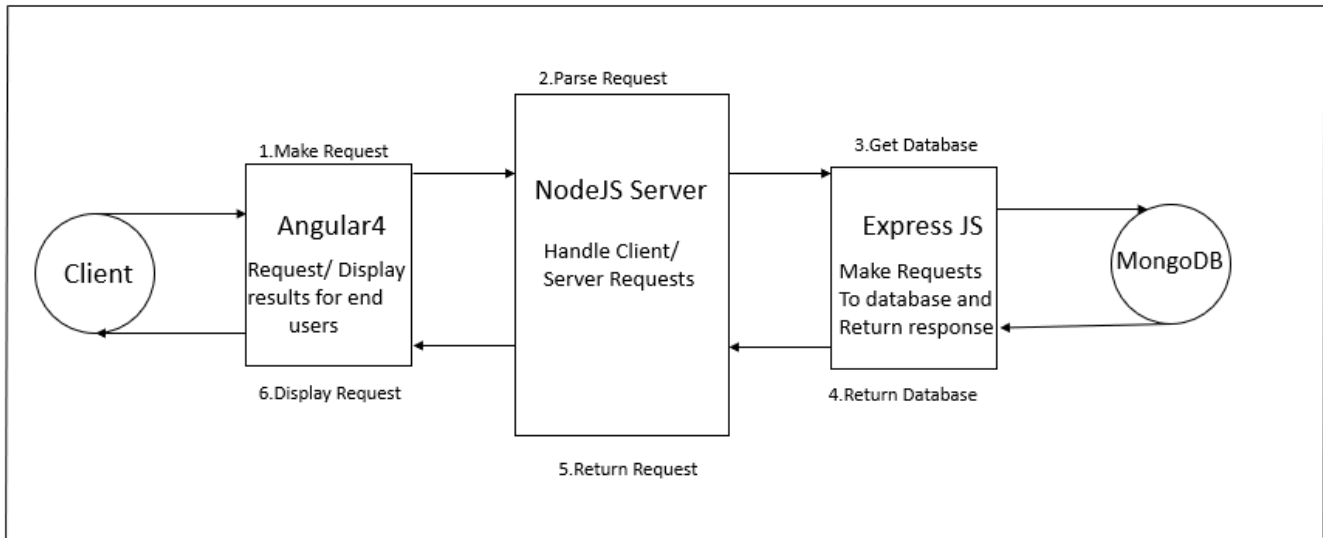


Figure 10: Development Architecture

System Architecture

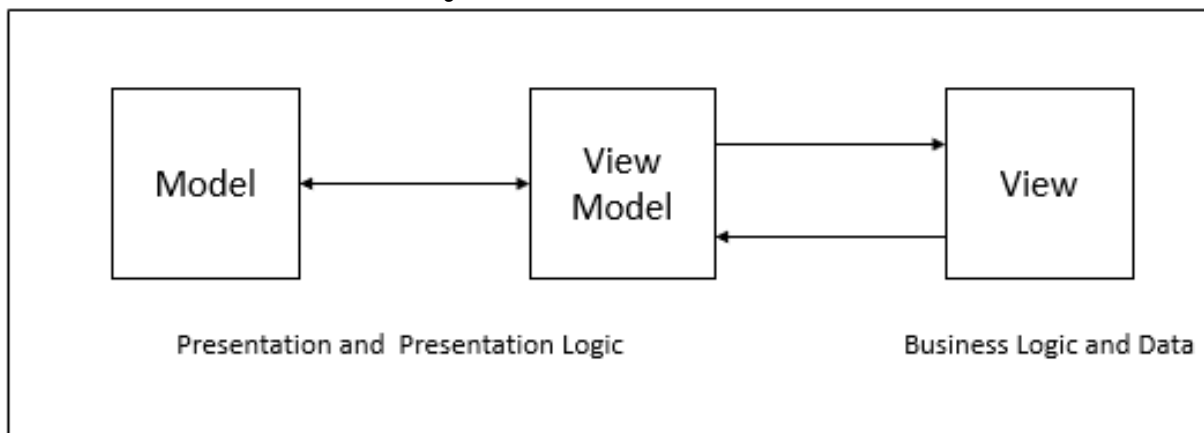


Figure 11: System Architecture

Model: - It simply holds the data and has nothing to do with any of the business logic. In our system we use type script file as a model file which contain class.

View Model: - It acts as the link between Model and View. In our system we use typescript file as a view model which connect model and view parts with two-way binding.

View: - It holds the formatted data and delegates it to model. In our system we use html file for represent view part

1. Authentication Module:

This module contains registration, login, forgot password and reset password components. First time registration component is used for user registration. After that login component used for login purpose. Forgot password and reset password components are for forgot password and set new password after verifying email verification code.

2. Profile Module:

This module contains user profile, other user profile, edit profile, my questions and my answer components. User profile components is for viewing profile of logged user. Other user's profile component is for viewing profile of any user. Edit profile components for edit personal, communication or privacy details. My questions and My answers components are used for view asked questions and posted answer by particular user. This module also used for view solution that are already submitted by user.

3. Contest Module:

This module contains view contest, view contest problem and ranking components. View contest component is for display all past contest, live contest and some future contests. View contest problem component is for display all problems in particular contest. Ranking component is for display ranking table for particular contest. It also uses compile and run component for compile code and submit solution.

4. Practice Module:

This module contains practice component. Which is for display all practice problem with different category. This module also use compile and run component for compile code and submit solution.

5. Discuss Module:

This module contains discuss and answer components. Discuss component is for display recently asked question on system with answer given by any user. Answer component is for post answer for particular question. This answer components also maintain like and dislike parts for answer.

6. Admin Module:

This module contains mainly three sub modules manage problems, manage contest and manage user. Manage problems module contains create, view and edit contests components. Similarly Manage contest module contains create, view and edit components for contests. Manage contest module is for start and end contest with publish contest functionality. Manage user module contains one component which is for delete any misbehavior user.

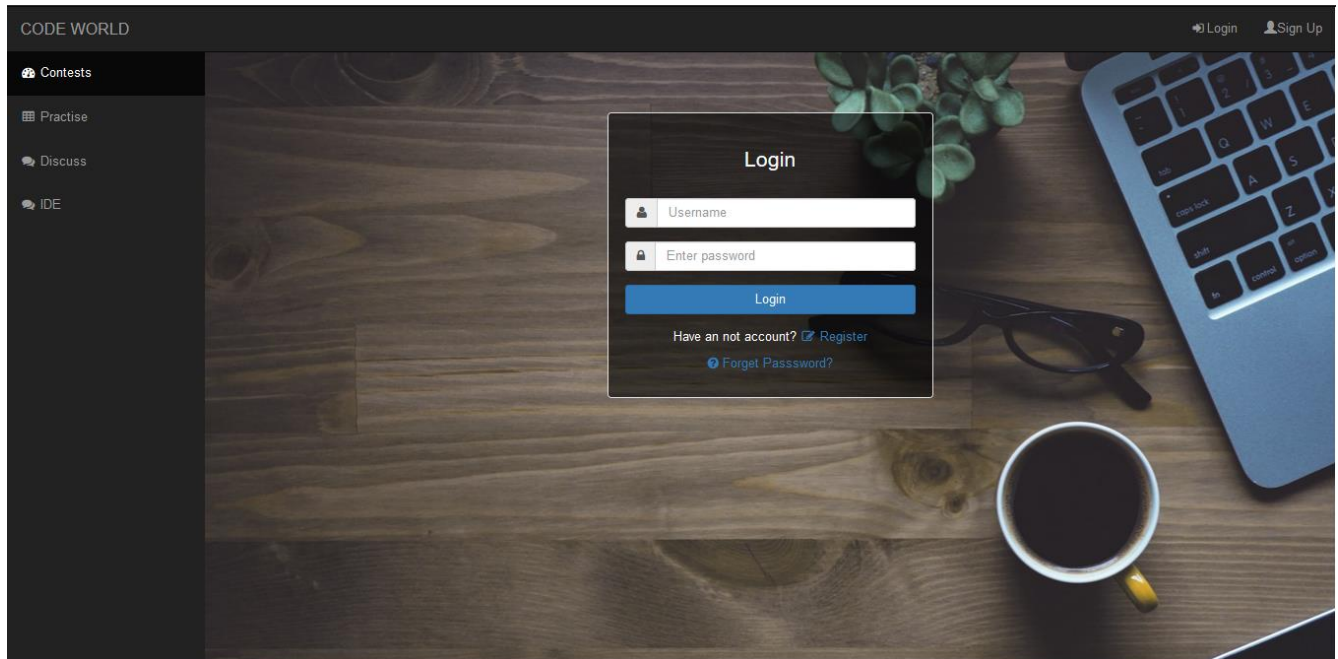
Test, case ID	Test Scenario	Test Steps	Test Data	Expected Results	Pass Or Fail
T01	Register with Valid Data.	Provides Correct information in the form.	User Information	User registers successfully	Pass
T02	Register with no details.	Provide no information in Register form.	No information	Error Message	Pass
T03	Register with invalid details.	Provide incorrect information.	User Information	Error message	Pass
T04	Login with valid details.	Provide correct details.	Login Credentials.	User Logged in.	Pass
T05	Login with no details.	Provide no information.	No Data.	Error message.	Pass
T06	Login with invalid details.	Provide incorrect details.	Login Credentials.	Error message.	Pass
T07	Valid Username for Forgot password.	Provide Registered Username	User Name	Send OTP in mail to registered email id	Pass
T08	Invalid user name for Forgot Password.	Provide invalid user name	User Name	Error message.	Pass
T09	Forgot password: Valid Username but wrong OTP	Provide incorrect details.	User Name	Validate OTP and show Error message	Pass

T10	Upload Correct Solution	Provide Correct Solution	Solution of problem in contest	Check test cases and show output (Correct Solution)	Pass
T11	Upload wrong solution (logical)	Provide wrong solution	Solution of problem in contest	Check test cases and show error message (wrong answer)	Pass
T12	Upload wrong solution (syntax)	Provide wrong solution	Solution of problem in contest	Check test cases and show error message (Time limit exceed)	Pass
T13	Upload Correct Solution but not optimal (taking more time)	Provide Correct Solution	Solution of problem in contest	Project allocated successfully	Pass
T14	Invalid URL Or user Tries to access admin page	Provide dummy URL or restricted URL	Wrong URL	Redirect to designated page (Home)	Pass
T15	Asking question in discuss section	Provide Question	Question	Question should be inserted in database	Pass
T16	Asking Question without login	Provide Question	Question	Show Error message	Pass
T17	Post Answer of question in discuss	Provide Answer	Answer	Answer should be inserted in database	Pass
T18	Post answer without login	Provides Answer	Answer	Error message	Pass
T19	Give like or dislike on answers	Hit like or dislike	Like or dislike	Appropriate data should be inserted in likes table	Pass
T20	Admin deletes any Question	Provide delete request	Delete request	Questions as well as Answer and like tables should also updated	Pass

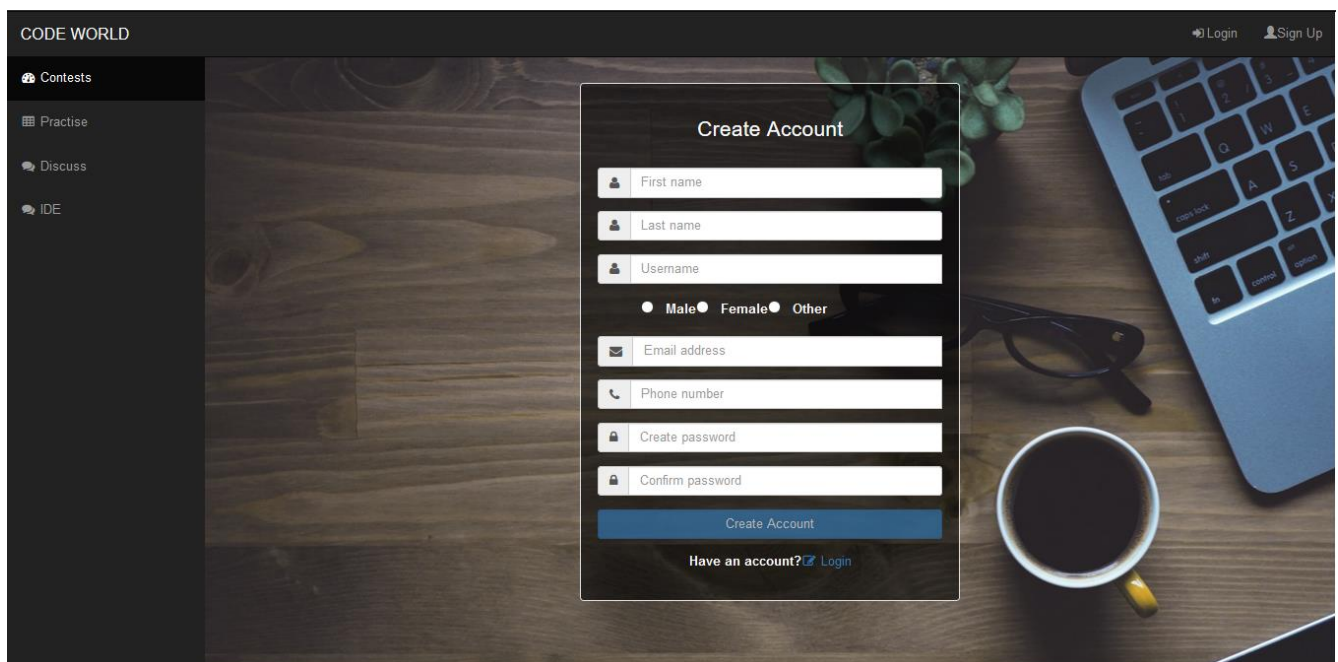
T21	Admin Creates new contest with same contest id	Provide credentials for creating contest	StartTime, endTime, id	Show error message for duplicate contest id	Pass
T22	Admin Adds new problem with same problem id	Provide credentials for adding new problem	Problem, test cases, id	Show error message for duplicate problem id	Pass

Chapter 6

Screen Shot



Login page



Registration page

CODE WORLD

Login

Sign Up

Contests

Practise

Discuss

IDE

View Contests

Contests

Name	Code	Successful Submission	Accuracy
Two VS Ten	TWOVSTEN	10	33.33 %
Magician versus Chef	MAGICHF	8	40.00 %
Byte To Bit	BITOBYT	5	18.52 %
Chef and Serves	CHSERVE	12	60.00 %
Judging delay	JDELAY	1	20.00 %

View Ranking List

Contest problems page

CODE WORLD

Savan Chovatiya

Manage Problem

Manage Contest

Manage User

Manage Discuss

Manage Problems

View Problems

Create Problem

Problem Code	Problem Name	Problem Type	Operations
TWOVSTEN	Two VS Ten	Easy	View
MAGICHF	Magician versus Chef	Medium	View
BITOBYT	Byte To Bit	Medium	View
CHSERVE	Chef and Serves	Hard	View
JDELAY	Judging delay	Medium	View

Manage problem

CODE WORLD

Savan Chovatiya

Manage Problem
Manage Contest
Manage User
Manage Discuss

Profile

Home / Savan Chovatiya

About

Communication

Username	SavanChovatiya
Full name	Savan Chovatiya
Gender	Male
BirthDate	1999-07-31

Recent Activity

Date/Time	Problem	Result	Language
Mar 22, 2019 01:58	JDELAY	AC [100 pts]	C++
Mar 22, 2019 01:57	CHSERVE	AC [100 pts]	C++
Mar 22, 2019 01:57	MAGICHF	AC [100 pts]	Python
Mar 22, 2019 01:56	BITOBYT	AC [100 pts]	C++
Mar 22, 2019 01:56	BITOBYT	TLE	C++
Mar 22, 2019 01:44	MAGICHF	AC [100 pts]	Python
Mar 22, 2019 01:43	MAGICHF	TLE	C

Profile page

CODE WORLD

Login Sign Up

Contests
Practise
Discuss
IDE

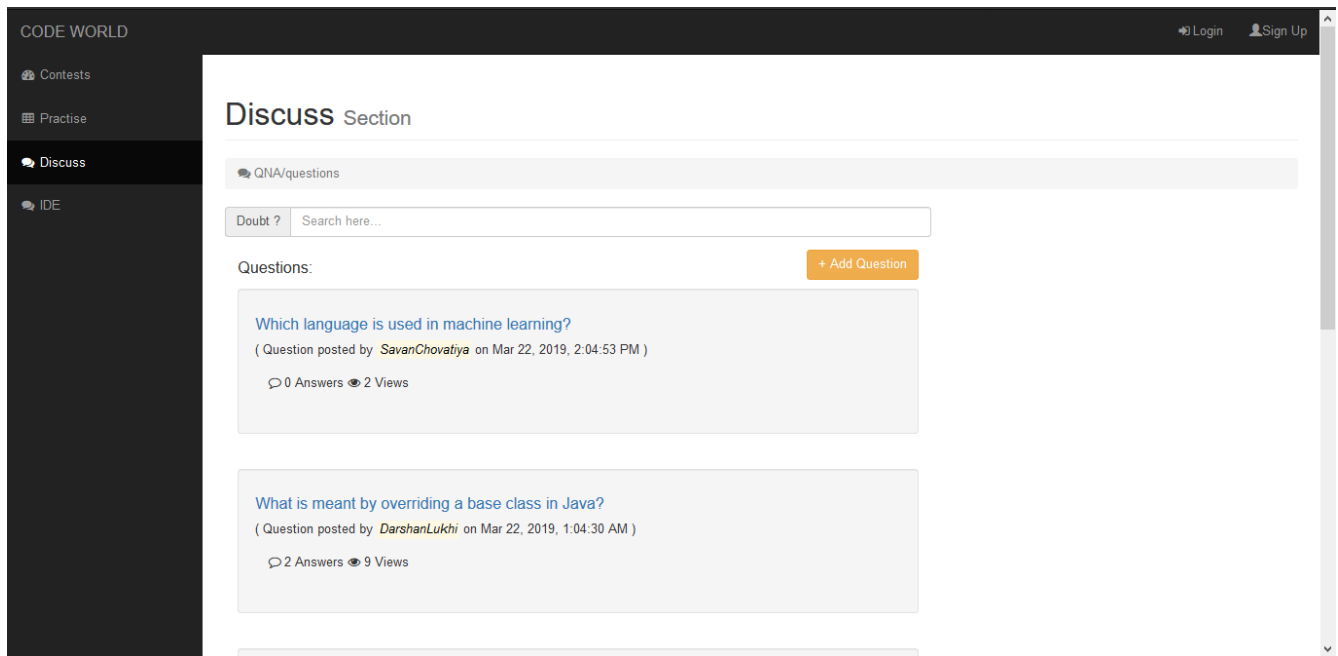
RANKS - MARCH19

Ranking

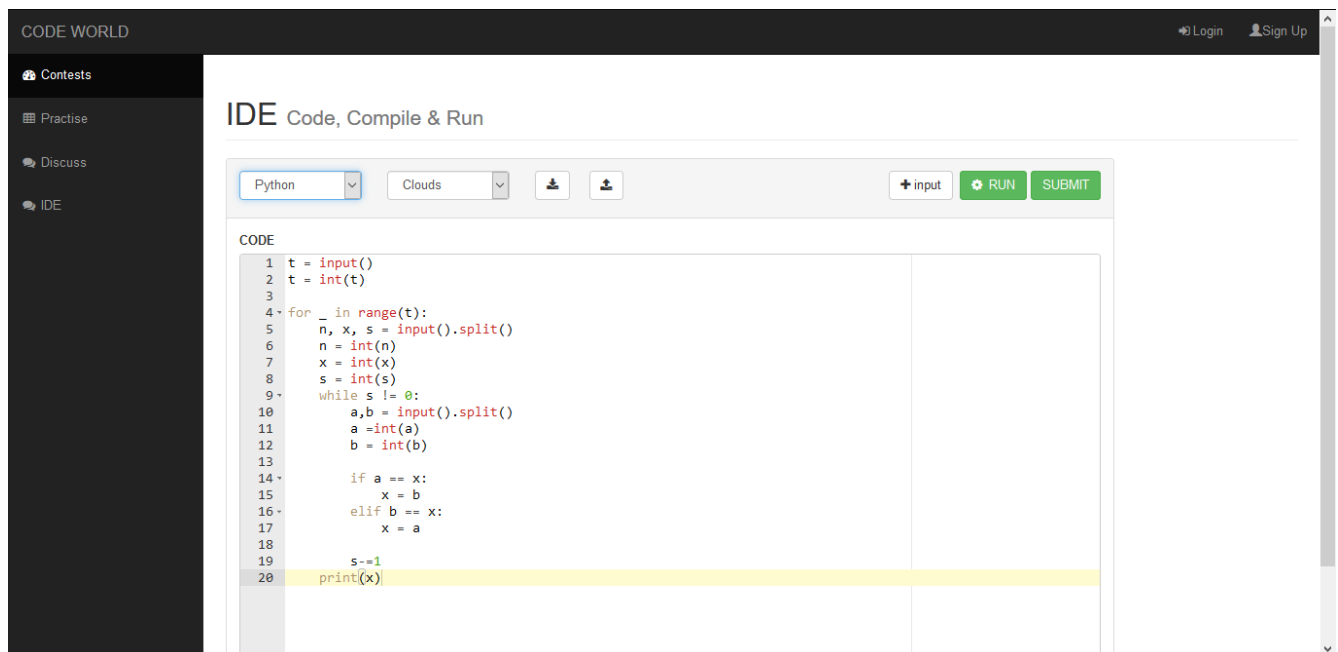
RANK	USER_NAME	POINTS	TWOVSTEN	MAGICHF	BITOBYT	CHSERVE	JDELAY
1	gotulukhi	500	100	100	100	100	100
2	SavanChovatiya	400	---	100	100	100	100
3	IshanGhelani	200	100	100	---	---	---
4	DarshanLukhi	100	100	---	---	---	---

« Previous 1 Next »

Ranking



Discuss Question page



Submit solution

In this Coding Competition System project, we implement most of functionality precisely. We complete most important functionality automatic testcase checking. We complete discuss part which contains question-answer feature, contest creation and participation, ranking for every contest, IDE to run code, practice problem solving for get knowledge about different language, solution submission for problems, view solution which are already submitted by user, profile with my question and my answer part. So, this are some major functionality which are implemented successfully.

Chapter 8

Limitations and Future Extensions

This Coding Competition System is not hosted on any online platform so this system is not available online now. But this Coding Competition System can be used at university level coding competitions by hosting on intranet.

In future this Coding Competition System can be online after hosting on any platform so which can be useful for all students and coding lovers for increase coding level. This system can also extend by adding some functionality like hosting contest which can be useful for small coding competition and university level competition. This system be updated by feature like ranking based on institute or country which will be useful for university to find top students from university.

1. For Angular: -
<https://angular.io/docs>
2. For Node.JS: -
<https://nodejs.org/en/docs>
3. For MongoDB: -
<http://www.tutorialspoint.com/mongodb>
4. For Error and exception handling: -
<https://stackoverflow.com>
5. For Package resources: -
<https://www.npmjs.com>
6. For Front End: -
<https://www.w3schools.com/html>
<https://www.w3schools.com/Css>
<https://www.w3schools.com/bootstrap>