

# Software Requirements Specification

---

*For Ecodena  
(An Online Judge System)*

This document is an attempt to provide a detailed description of Ecodena, our Online Judge System. It begins by describing the project purpose, scope and the tools used. It then gives an overview of the system by describing the perspective, functions, user characteristics, architecture and database diagrams. It then entails some more specific requirements with Use Case and Activity Diagrams

# Table of Contents

|   |    |
|---|----|
| 1. Introduction: .....                              | 3  |
| 1.1 Purpose: .....                                  | 3  |
| 1.2 Scope .....                                     | 4  |
| 1.3 Definitions, Acronyms, and Abbreviations: ..... | 8  |
| 1.4 References:.....                                | 8  |
| 1.5 Technologies to be used: .....                  | 9  |
| 1.6 Overview: .....                                 | 9  |
| 2. Overall Description: .....                       | 10 |
| 2.1 Product Perspective: .....                      | 10 |
| 2.2 Software Interface: .....                       | 10 |
| 2.3 Hardware Interface: .....                       | 10 |
| 2.4 Product Functions:.....                         | 11 |
| 2.5 User Characteristics: .....                     | 13 |
| 2.6 Constraints: .....                              | 13 |
| 2.7 Architecture Design: .....                      | 14 |
| 2.8 Use Case Description : .....                    | 15 |
| 2.9 ER Diagram: .....                               | 20 |
| 2.10 Assumptions and Dependencies: .....            | 21 |
| 3. Specific Requirements: .....                     | 22 |
| 3.1 Use Case Model Survey: .....                    | 22 |
| 3.2 Supplementary Requirements: .....               | 23 |
| Index .....   | 27 |

# 1. Introduction:

## 1.1 Purpose:

The purpose of this document is to describe the software requirements for the Ecodena. 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 documents aims to give necessary view of requirements for each stakeholder considering their responsibilities.

This document is intended for:

- ❖ **Developers:** In order to be sure they are developing the right software that fulfils requirements provided in this document.
- ❖ **Programmers:** To get familiar with the idea of the software and suggest other features that would make it even more functional.
- ❖ **Problem Setters:** To get familiar with set of problems provided in different levels and suggest more problems.
- ❖ **Contest holders:** To get familiar with procedures involved in assigning contest problems and grading them and suggest more practical features.
- ❖ **Administrators:** In order to know exactly what they have to expect from the system, right inputs and outputs and response in error situations.
- ❖ **Documentation writers:** To know the features and in what way they have to explain. What security technologies are required, how the system will response in each user's action.

## 1.2 Scope

**Information regarding users of the system and their description is given below:**

- Online Judge System will have four different types of Users
  - Programmers
  - Problem Setters
  - Contest Holders
  - Site Administrators
- A Programmer is the most basic unit instance of a user on the system. He has registered himself and can solve problems in the tutorial as well as battle section.
- A Problem Setter has the privileges to upload and submit a problem which can be added to the above two sections or used solely in a contest.
- A Contest Holder is a user with privileges to hold a contest on the site. He can use his own problems or problems existing on the site.
- Site Administrators has all the above privileges plus he can add/remove any user/problem/solution/contest/comments/posts in forums
- All other users of the system are granted their respective privileges from site admins.

- Different users can give their feedback or post Queries related to website. Admin does the job of reviewing and resolving them.
- Admin has the ability to backup/recover site data, generate various reports, etc

### **The project will be divided into two major sections:**

- ❖ Tutorial Section
- ❖ Battle Section

### **Features of the Battle Section:**

- At-least four different languages support to be provided. (C, C++, java, Python)
- Maintenance of the database of questions solved and each and every solution submitted by a registered user.
- Pointing System -
  - Each user will be initially provided with about 1000 points.
  - If the user submits a correct question, points will be scored based on the pointing scheme as defined for that question.
  - If the user submits an incorrect question, points will be deducted from his score.
  - Hints shall be provided if the user is unable to submit the correct solution more than once.
- Badging System - If a user solves more than “x” questions of a particular type, then he/she will be provided with a badge of mastery for that type.
- Problem Rating -
  - Assignment of type and the level of all problems - tagged internally

- Test Cases:
  - Check the algo/program with a pretty small set of solutions.
  - If the above test succeeds then test the algo/code for large test cases.
    - Crash
    - Time limit exceeded
    - Memory allocation
    - Corner Cases
  - Note: The problem setter should give the suitable classifications for the above test cases along with the possible ideas for the errors.
- Error Reporting.
  - Specific Errors like exceptions (if any), memory error, compile time error, time limit exceeded will be provided along with a description (attempt to provide both a technical and non-technical) of the error. Also, depending on in which of the above three test zones a failure occurs, appropriate recommendation would be suggested.
- User statistics to be provided -
  - Overall rank
  - Total points scored.
  - Badges gained through solving of problems.
  - Table like SPOJ for all submissions by the user.
  - Graph of type proficiency (if time permits).
- Algorithm for Problem rating:
 
$$(\text{Number of Users}) * (x \text{ points}) * (\text{accuracy or multiplier})$$
- Algorithm for User rating:
 
$$\text{Base Points} + (\text{Points per problem}) - \{(\text{number of submissions}) * (\text{some points})\}$$
- Class-room chat session can be initiated by any super-user who has the TUTOR badge for a particular type of problem (if time permits).
  - Users won't be able to chat with each-other directly.

- The users can directly PM to the super-user, his/her doubts and question regarding a particular problem.
  - If the super-user finds the doubt/question worth mentioning, then the super-user would then be giving his feedback in real time dynamically.
- Provision for Login or register with Google/Facebook account. (If time permits)

### Features of the Tutorial section:

- Not all the tutorial problems will be available initially.
- In order to unlock more than X tutorial problems in each topic user needs to register.
- Links will be available for video tutorials.
- Forum and Wiki section will be available for assistance in various topics.
- As and when the user progresses in the tutorial section, more questions will be unlocked.
- If any anonymous user who is unregistered, if tries to submit a solution to any tutorial problem, then he/she will have to sign in/ sign up in order to continue.

## 1.3 Definitions, Acronyms, and Abbreviations:

- HTML (Hyper Text Markup Language): It is used to create static web pages.
- DIA (Diagram): Dia is a gtk+ based diagram creation program released under the GPL license.
- PGAdmin3 (PostgreSQL): PostgreSQL is a powerful, open source relational database system.
- Django ( ): Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.
- CSS (Cascading Style Sheets): Cascading Style Sheets is a style sheet language used to describe the presentation semantics (the look and formatting) of a document written in a Markup language.
- HTTP (Hyper Text Transfer Protocol): It is a transaction oriented client/server protocol between a web browser and a web server.
- Dreamweaver: Dreamweaver (formerly Macromedia Dreamweaver) is a proprietary web development application originally created by Macromedia.
- Photoshop: Photoshop is the professional standard digital picture & photo editing software.
- Gedit (GNOME Editor): Gedit is the official text editor of the GNOME desktop environment.
- Git: Git is an extremely fast, efficient, distributed version control system ideal for the collaborative development of software.
- GitHub: GitHub is the best way to collaborate with others. Fork, send pull requests and manages all your **public** and **private** git repositories.
- BitBucket: *Bitbucket* is a free code DVCS hosting site for Git and Mercurial. Manage your development with a hosted wiki, issue tracker and source code.
- Python : *Python* is a general-purpose, high-level programming language whose design philosophy emphasizes code readability. Python claims to combine "remarkable power with very clear syntax".

## 1.4 References:

[1] Roger S. Pressman, "Software Engineering-a practitioner's approach", 5th ed., McGraw-Hill.



- [2] Prof. Asim Banerjee slides on “Software Engineering”.
- [3] IEEE SRS Format
- [4] IBM TGMCM Sample Synopsis
- [5] Wikipedia
- [6] [Bruade] The principal source of textbook material is “Software Engineering: An Object-Oriented Perspective” by Eric J. Bruade (Wiley 2001).
- [7] Github ( <http://github.com> )
- [8] Bitbucket ( <http://bitbucket.org> )

## 1.5 Technologies to be used:

- Django ( ): Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.
- Smart draw – A visual modelling tool for modelling various types of diagrams
- Dreamweaver: Dreamweaver (formerly Macromedia Dreamweaver) is a proprietary web development application originally created by Macromedia.
- Photoshop: Photoshop is the professional standard digital picture & photo editing software.
- Gedit (GNOME Editor): Gedit is the official text editor of the GNOME desktop environment.
- Git: Git is an extremely fast, efficient, distributed version control system ideal for the collaborative development of software.
- DIA (Diagram): Dia is a gtk+ based diagram creation program released under the GPL license.
- PGAdmin3 (PostgreSQL): PostgreSQL is a powerful, open source relational database system.

## 1.6 Overview:

**Overall Description:** This section will describe major components of the system, interconnections, and external interfaces.

**Specific Requirements:** This section will describe the functions of actors, their roles in the system and the constraints faced by the system.

## 2. Overall Description:

### 2.1 Product Perspective:

Ecodena is a platform for programmers to meet, compete and enhance their programming skills. Basically it's an online judge system, which serves for the automatic assessment of user-submitted programs. As it primarily focuses on tutoring to actually learn new programming paradigms and implement them in code, before moving on to actual more difficult problems. With Ranking system for users based on solving problems provides the competitive edge for normal users to perform better and enhance their programming skills through set of problems.

### 2.2 Software Interface:

**Front End Client:** Web Browser, Operating System (any)

**Web Server:** Apache

**Data Base Server:** PostGre SQL

**Back End:** Django, Python Libraries, OS (Linux)

### 2.3 Hardware Interface:

#### Client Side

|                   | Client Side                        |        |            |
|-------------------|------------------------------------|--------|------------|
|                   | Processor                          | RAM    | Disk Space |
| Web browser (any) | Intel Pentium III or AMD – 800 MHz | 128 MB | 100 MB     |

#### Server Side

|                  | Server Side                        |        |            |
|------------------|------------------------------------|--------|------------|
|                  | Processor                          | RAM    | Disk Space |
| Concurrent Model |                                    | 1 GB   | 8 GB       |
| PGAdmin 3        | Intel Pentium III or AMD – 800 MHz | 256 MB | 500 MB     |

## 2.4 Product Functions:

Ecodena provides the users with the following functions:

- **Register**  
A login screen is required for the system. There will be Username and e-mail address of the user and their password will be taken and a user profile will be created if there isn't one with that mail address and username already.
- **Login**  
Registered users can login to the system after providing their usernames and passwords.
- **Update profile**  
A profile editing page is required to be more social in this environment. A profile picture and a general profile info creation must be included. A profile should only be edited by its owner.
- **Search Users**  
Users can search a particular user through his Username/e-mail.  
View of problems  
Users that are logged into system can view set of problems within specified categories.
- **Submit a solution**  
Users that are logged into the system can submit a solution of the problem written in one of the programming languages (C, C++, JAVA, and Python).
- **View solutions submitted**  
Users can see his/her submitted solutions any time.

- **View results**

Users can view the results of their submitted solution to any problem, with specified errors if any occur.

- **Register to a contest**

Users can register to a contest which is organized by contest holders.

- **Badge**

Users will get a badge which will define achievements achieved by him/her.

- **Contest Holding**

Users that are logged into the system can organize contests.

- **Tutorials**

Users that are logged into the systems and are newbie to programming can use tutorials where a set of problems are given defined within certain levels. Users have to solve first set of problems than only he/she can go to next level. Users can see an optimal solution for the problem after submitting his/her solution.

- **Set Problems**

Users that are experts in programming and have solved lots of problems will gain a badge. This will give them rights of setting problems for other users primarily for newbie's.

- **Comment**

Registered users can comment over a problem.

- **Forum**

Users can write or read articles related to programming but registered users can comment over it.

- **Wiki**

Users can get more information about any program related topic.

## 2.5 User Characteristics:

### ❖ **Programmer:**

Normal users are primary users of the Ecodena. We expect most of the users to be university students. Most of the users will have medium level knowledge on programming.

### ❖ **Problem setters:**

Problem setters will be essentially contestants with high ranking and a badge received from Admin. Most of the judges will be experts on the field of programming.

### ❖ **Contest holders:**

Contest holders will be the normal users who can hold a contest with Admin's approval. They can hold a contest with set of problems defined by them. Most of them, around %25 will have a programming competition experience.

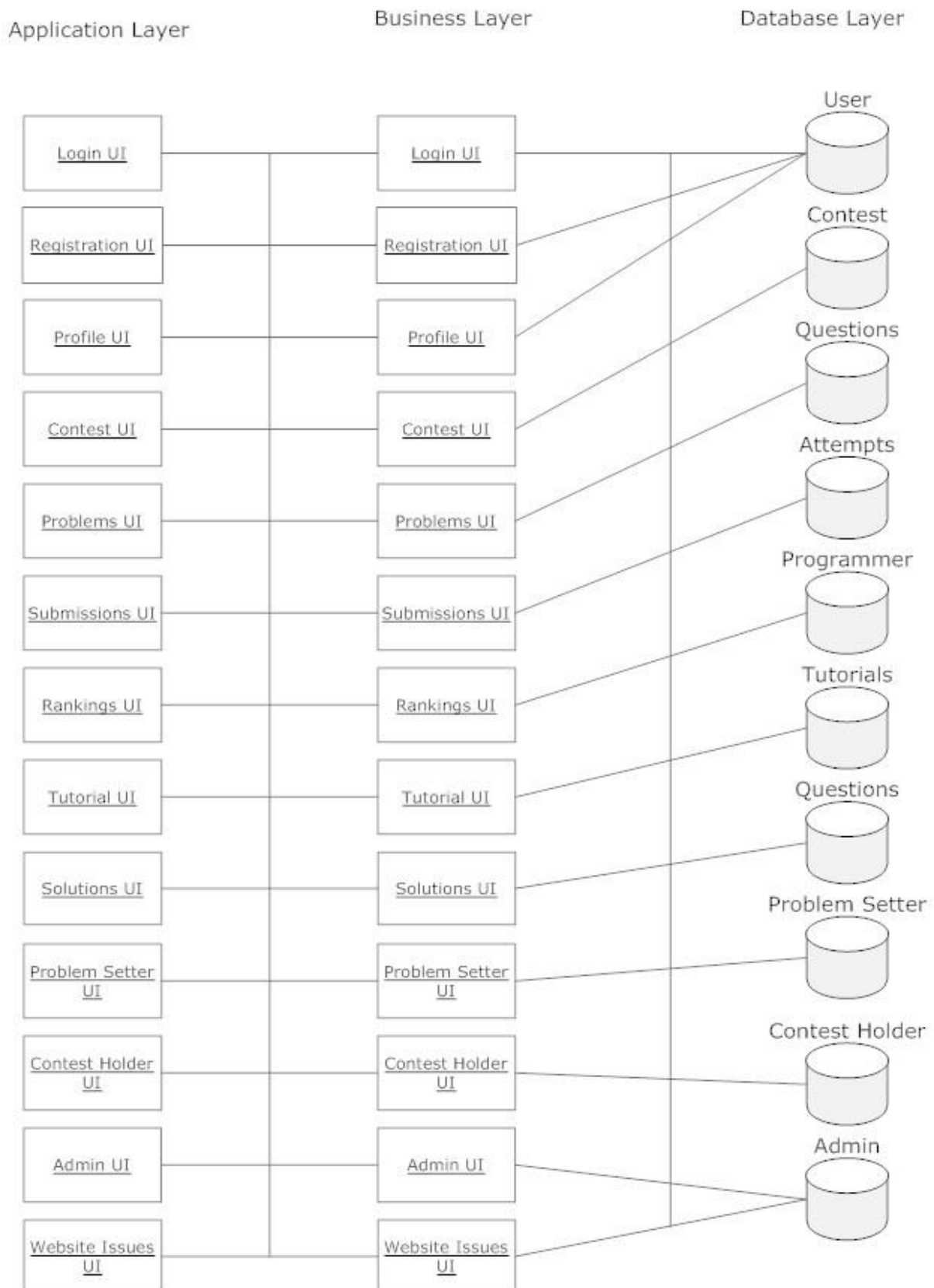
### ❖ **Site administrators:**

Administrators are primarily responsible for maintenance of the system. They contribute minimally to the problems and contests themselves, but spend more time modifying the system configuration and making updates.

## 2.6 Constraints:

- GUI is only in English.
- Username/e-mail and password is used for the identification of users.
- Only registered user will be authorized to submit the solutions and post in a forum.
- Limited to HTTP.
- This system is working for single server.

## 2.7 Architecture Design:



## 2.8 Use Case Description :

### Register Account

- User enters all required information (username, password, email, etc.)
- System verifies the information (whether username already exists, etc.)
- If valid, system creates an entry in the database for the user and presents the user with respective homepage
- If invalid, system presents the user with appropriate error.

### Login

- User enters username and password
- System verifies the details
- If valid, user is presented with the homepage
- Else system presents the user with login page again indicating bad username/password

### View a problem

- User is presented with a list of problems.
- User selected the problem he wants to view
- User is presented with the selected problem

### Submit a solution

- User selects the language and compiler he wants to submit the solution in
- User types the code in the editor OR User uploads the source file
- User clicks on submit button
- System makes entries in database and passes data to judge backend
- Judge backend evaluates the code against the problem, inputs and outputs and passes the results to the system
- System presents the user with appropriate error report.

### View his solutions

- User is presented with the list of available solutions according to the problem, language and timestamp.
- User selects the solution he wants to see by clicking on the link associated with it
- User is presented with the selected solution

### View profile

- User clicks on his profile link
- User is presented with the profile page containing various information and performance statistics including ranks badges, etc.

### Edit profile

- User views his profile
- User clicks on edit button
- User makes changes to various information such as password, full name, college, etc.
- If User clicks on save button, system makes appropriate changes to the database and presents the user with the modified profile
- If User clicks on cancel button, system presents the user with unmodified profile.

### Search

- User enters the text in the search box and clicks on search button.
- System presents the user with search results.

### View Help

- User clicks on the help link
- System presents the user with the help through which he can browse



### **Logout**

- User clicks on logout link
- System makes appropriate changes to database and presents the user with log-out page.

### **Comment on a problem**

- User enter the body of the comment and clicks on submit button
- System presents user with the list of modified comments

### **Post Website Issues to Admin**

- User enters the topic of the issue and the content explaining the issue
- User sends it to admin
- System generates an issue token and sends it to the user for future reference

### **View the contests**

- User is presented with the list of contests
- User clicks on a contest link
- User is presented with the details of the contest

### **Participate in a contest**

- If contest is open, user can submit solutions to problems in a contest
- If contest is closed, user requests to join the contest.
- If Contest holder approves the user, he is sent with a password for the contest and he can then participate
- Else he cannot participate in the contest.

### **View forums**

- User will click on the forum & discussion tab.
- He will be shown the list of active discussions in the forum from which he will chose one.
- He can view the comments posted in the forum.

### Post in forums

- After viewing a discussion the user can post in the discussion.

### View wiki

- User can click on the wiki tab to view on the wiki.
- He will then type the name of the topic in the search box to view the content he needed and he can browse through the wiki.

### Upload a problem

- Problem Setter will get into the Upload problem section by clicking My Problems tab.
- He will either upload a file or will paste the problem in the editor after which he will click the upload button.

### View his problem

- Problem setter can view the list of problems by clicking My Problems tab.
- He can click on any problem listed to view the whole problem.

### Delete his problem

- Problem setter after viewing his problem can click on the DELETE button.
- He will be asked of a confirmation to his decision and if he clicks on OK button his problem will be deleted.

### Hold a contest

- Contest holders can click on contest tab where he will click on Hold a Contest button.
- After which he will be asked to fill in the information on the contest and click on OK button after which the information will be send to site administrators to approve.

### View his contests

- Contest holders will click on contest tab where he will get a button View Your Contests.
- He will get the list of the problems hosted by him and he can view the details by clicking on the contest name.

### Modify his contest

- After viewing the contest he will get a button Modify after which he can modify any field in the contest.
- He will get a confirmation dialog box on which if he clicks OK the changes will be made.

### Delete his contest

- Contest holder after viewing his contest will get the button to DELETE the contest.
- He will be asked for a confirmation after which the contest will be deleted.

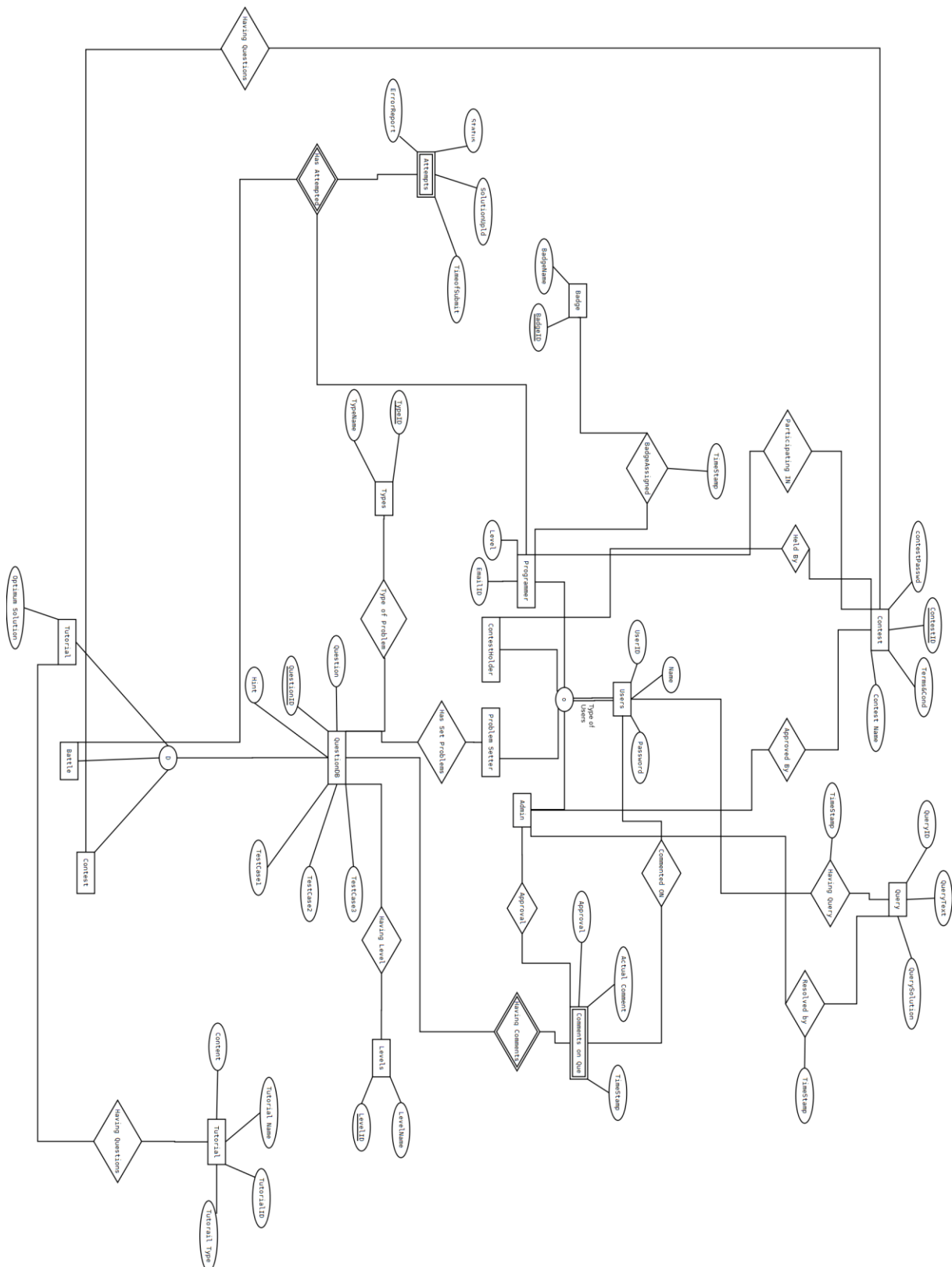
### Authenticate/ give privileges to problem setters and contest holders

- Admin will approve the request of problem setters and contest holders after they will be given the privilege to upload problems and contests.

### Add/remove any user/problem/solution/contest/comments/posts in forums

- Administrators can perform any of the listed functions mentioned in forums for which he will be given options.

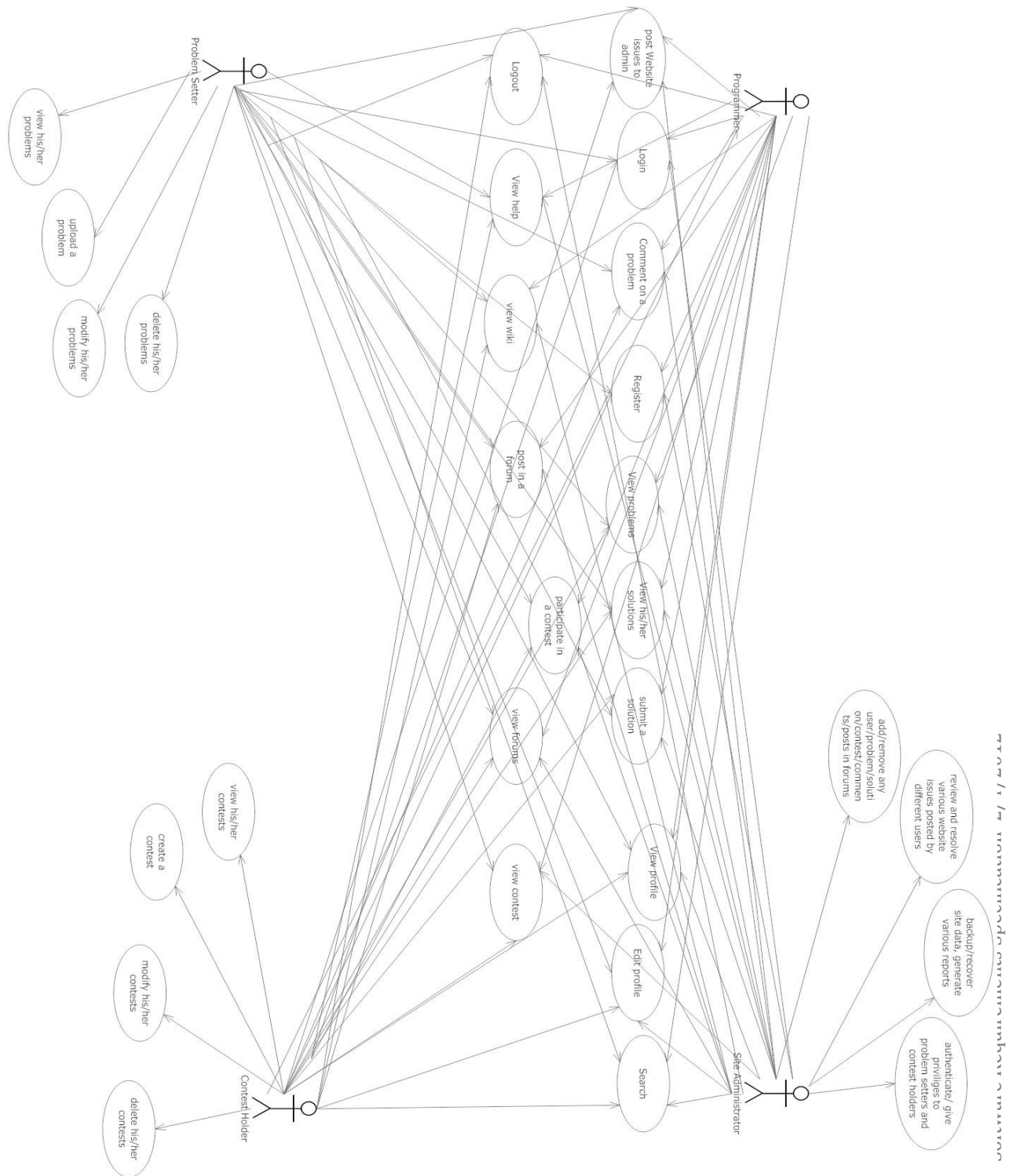
## 2.9 ER Diagram:



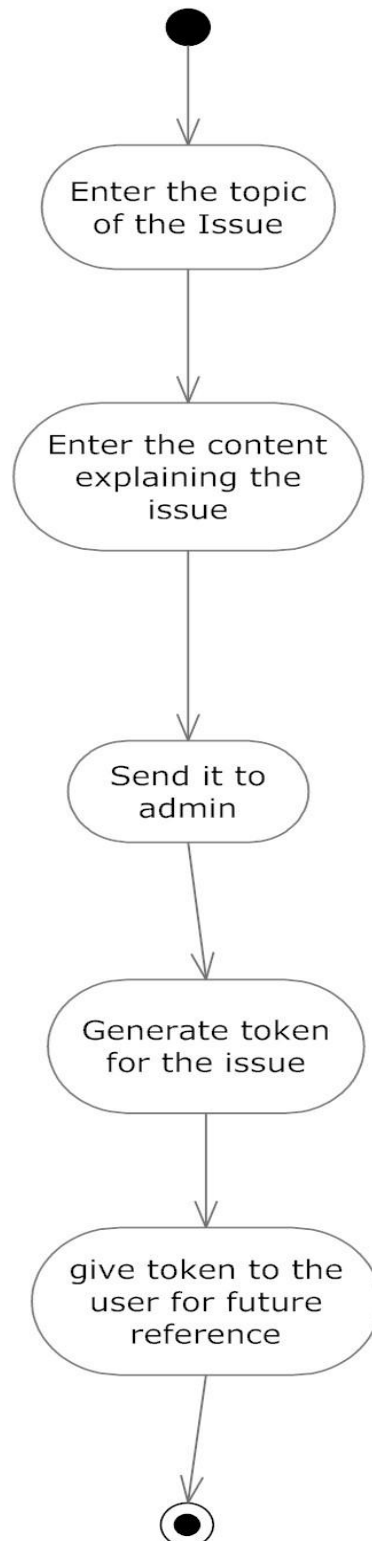
## 2.10 Assumptions and Dependencies:

- Users should know programming.
- Users should have some basic knowledge about web browser.
- System is restricted to work for a few languages only.
- We would not be responsible for the plagiarism on the part of users.

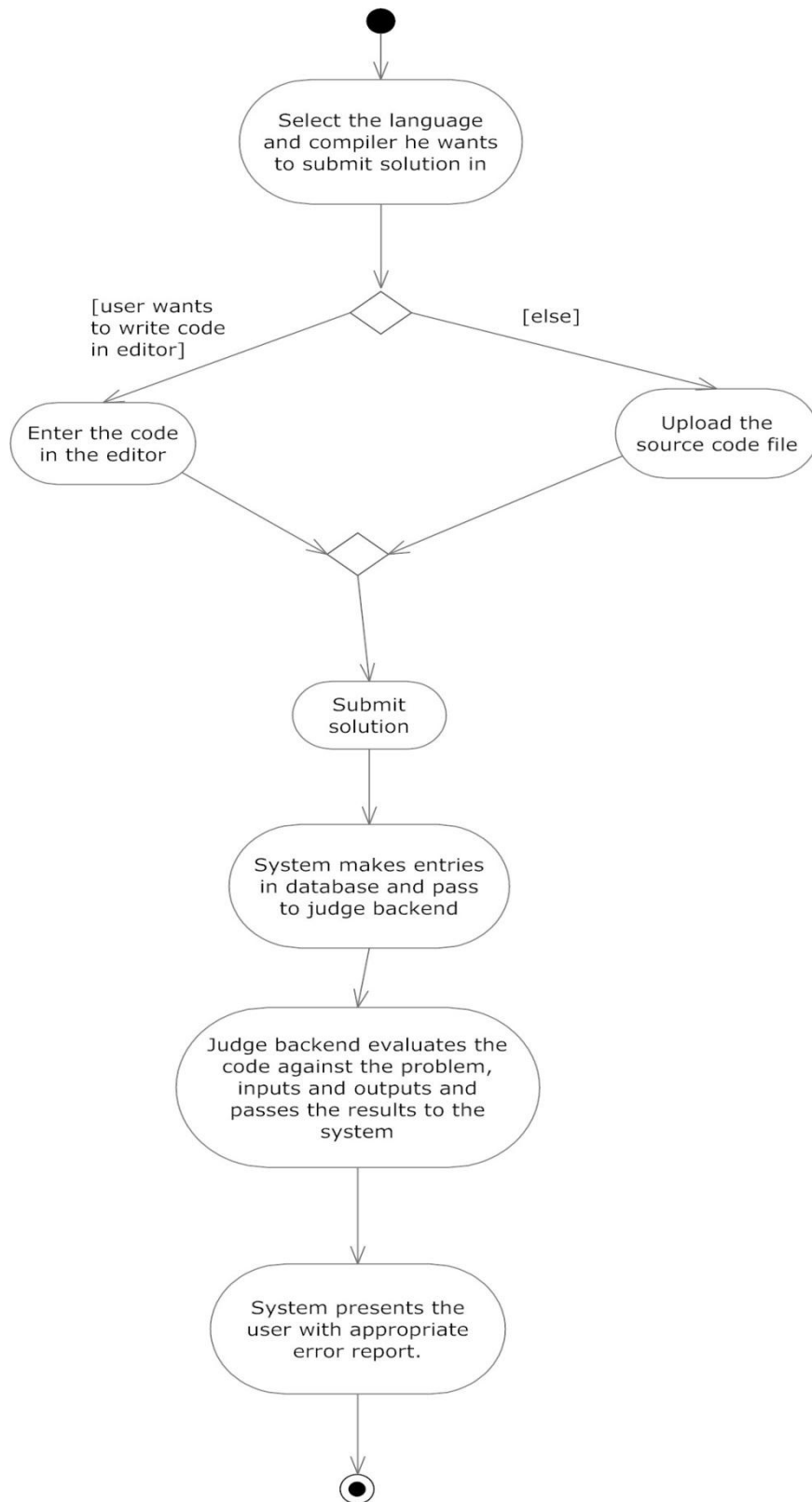
### 3.1 Use Case Model Survey:



## 3.2 Supplementary Requirements:

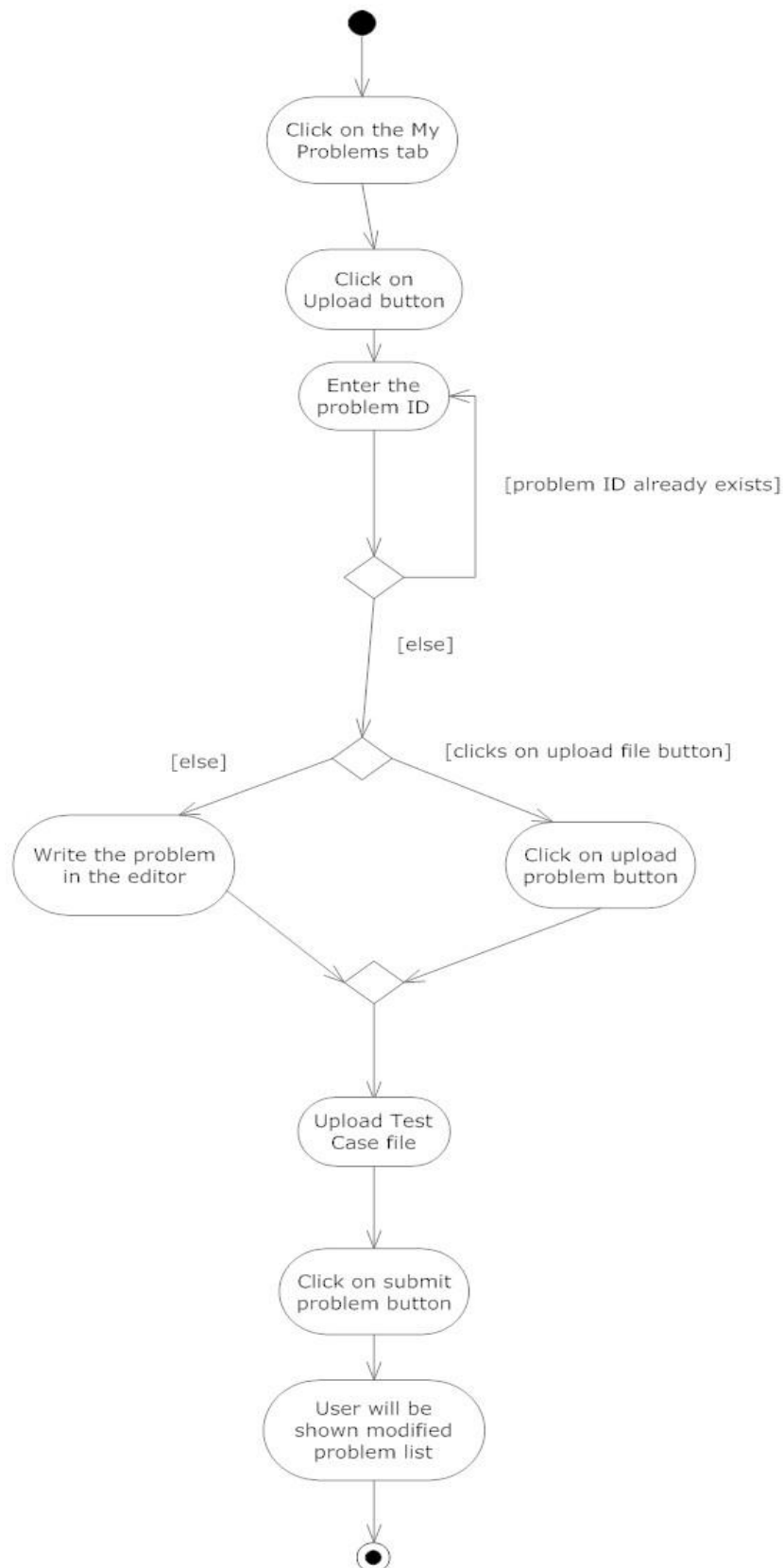


**Activity Diagram for “Post Issue on Website” Use Case**

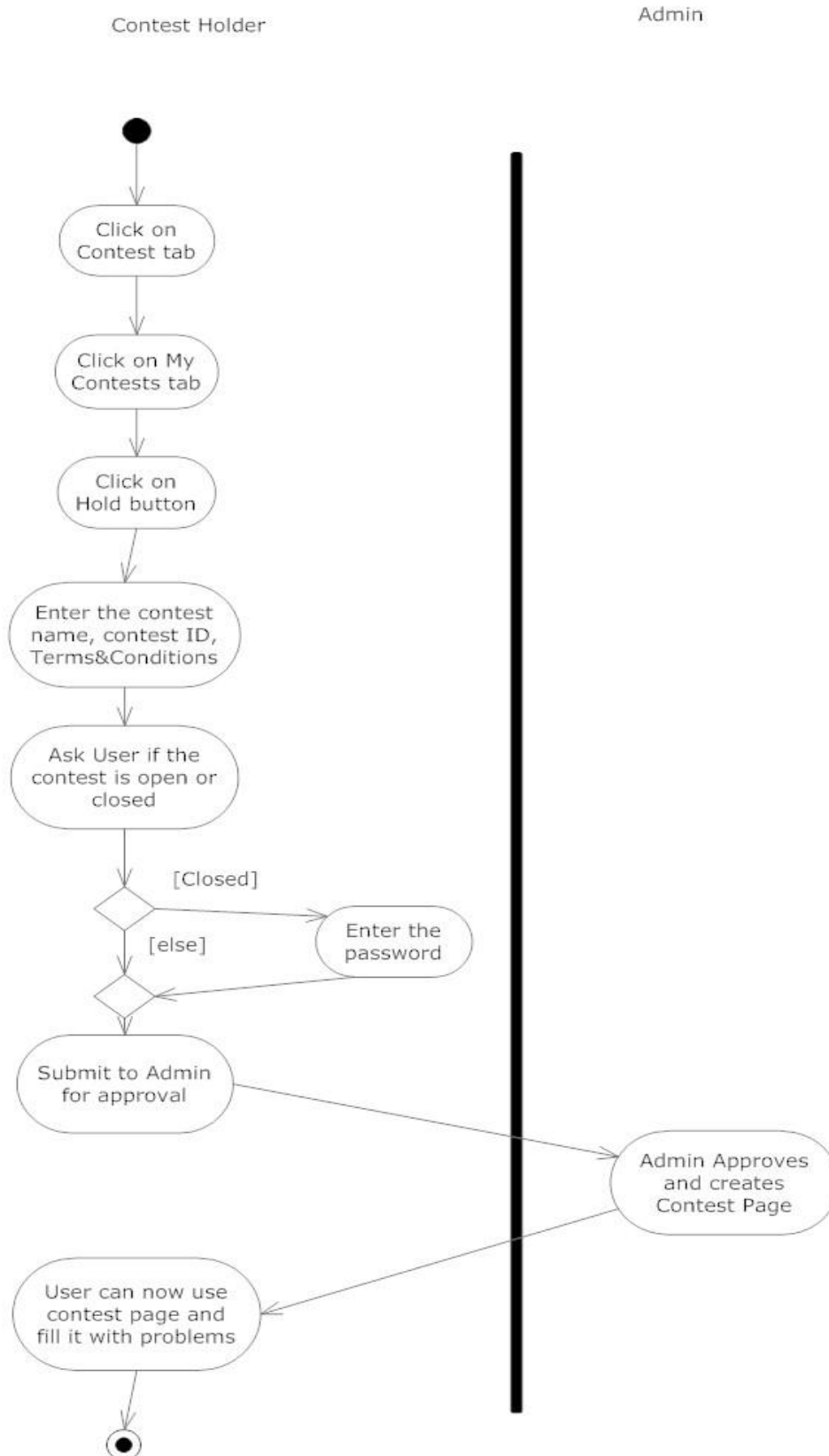


## Activity Diagram for “Submit a Solution” Use Case





## Activity Diagram for “Upload a Problem” Use Case



## Activity Diagram for “Hold a Contest” Use Case

# Index

- Add/remove any user/problem/solution /contest/comments/posts in forums, 21
- Administrators, 4, 5, 14, 21
- Architecture Design, 2, 15
- Assumptions and Dependencies, 23
- Authenticate/ give privileges to problem setters and contest holders, 21
- Back End, 11
- Badge, 13
- Badging System, 6
- Battle Section, 5
- BitBucket, 9
- Client Side, 12
- Comment, 14, 19
- Constraints, 2, 15
- Contest holders, 3, 14, 20, 21
- Contest Holding, 13
- CSS, 9
- Data Base Server, 11
- Definitions, Acronyms, and Abbreviations, 2, 8
- Delete his contest, 21
- Delete his problem, 20
- Developers, 3
- DIA, 8, 10
- Django, 9, 10, 11
- Documentation writers, 4
- Dreamweaver, 9, 10
- Ecodena, 3, 11, 12, 14
- Edit profile, 18
- ER Diagram, 2, 22
- Error Reporting, 6
- Forum, 8, 14
- Front End Client, 11
- Gedit, 9, 10
- Git, 9, 10
- GitHub, 9
- Hardware Interface, 2, 11
- Hold a contest, 20
- HTML, 8
- HTTP, 9, 15
- Introduction, 2, 3
- Login, 7, 12, 17
- Logout, 19
- Macromedia, 9, 10
- Modify his contest, 21
- Online Judge System, 4
- Overall Description, 2, 10, 11
- Overview, 2, 10
- Participate in a contest, 19
- PGAdmin3, 8, 10
- Photoshop, 9, 10
- Pointing System, 6
- Post in forums, 20
- Post Website Issues to Admin, 19
- Problem Rating, 6
- Problem setters, 14
- Problem Setters, 3, 4
- Product Functions, 2, 12
- Product Perspective, 2, 11
- Programmer, 4, 14
- Programmers, 3, 4
- Purpose, 2, 3
- Python, 6, 9, 10, 11, 13
- References, 2, 9
- Register, 12, 13, 17
- Scope, 2, 4
- Search, 12, 18
- Search Users, 12
- Server Side, 12
- Set Problems, 14
- Site administrators, 14
- Software Interface, 2, 11
- Specific Requirements, 2, 10, 25
- Submit a solution, 13, 17
- Supplementary Requirements, 3, 26
- Table of Contents, 2
- Technologies to be used, 2, 10
- Test Cases, 6
- Tutorial Section, 5
- Tutorials, 13
- Update profile, 12
- Upload a problem, 20
- Use Case Description, 2, 17
- Use Case Model Survey, 3, 25
- User Characteristics, 2, 14
- View a problem, 17
- View forums, 19
- View Help, 18
- View his contests, 21
- View his problem, 20
- View his solutions, 18
- View profile, 18
- View results, 13
- View solutions submitted, 13
- View the contests, 19
- View wiki, 20
- Web Server, 11
- Wiki, 8, 14