CLASS-BASED MODELING FOR PROGRAMMING PLATFORM

This Chapter is intended to describe class based modeling of “**Programming Platform**”.

CLASS BASED MODELING CONCEPT

Class-based modeling represents the objects that the system will manipulate, the operations that will applied to the objects, relationships between the objects and the collaborations that occur between the classes that are defined.

GENERAL CLASSIFICATION

To identify the potential classes, we have first selected the nouns from the solution space of the story. These were then characterized in seven general classification. The seven general characteristics are as follows

1. External entities (e.g., other systems, devices, people) that produce or consume information to be used by a computer-based system.
2. Things(e.g., reports, displays, letters, signals) that are part of the information domain for the problem.
3. Occurrencesorevents(e.g., a property transfer or the completion of a series of robot movements) that occur within the context of system operation.
4. Roles (e.g., manager, engineer, salesperson) played by people who interact with the system.
5. Organizational units (e.g., division, group, team) that are relevant to an application.
6. Places (e.g., manufacturing floor or loading dock) that establish the context of the problem and the overall function of the system.
7. Structures (e.g., sensors, four-wheeled vehicles, or computers) that define a class of objects or related classes of objects.

Following are the specifications of the nouns according to the general classifications:

|  |  |  |
| --- | --- | --- |
| Serial No | Noun | General Classification |
|  | Authentication | 3,5 |
|  | Problem set | 2,3,5,7 |
|  | Contest | 2,3,5,6,7 |
|  | User | 4,5,7 |
|  | Administrator | 4,5,7 |
|  | Problem Solver | 1,4,5,7 |
|  | Username |  |
|  | Email |  |
|  | Password |  |
|  | Recovery pin |  |
|  | Institute |  |
|  | Registration number |  |
|  | Designation |  |
|  | Problem |  |
|  | Description file |  |
|  | Input file |  |
|  | Solution file |  |
|  | Duration |  |
|  | Time limit |  |
|  | Language |  |
|  | Verdict |  |
|  | Wrong answer |  |
|  | Code |  |
|  | Submission | 2,5,7 |
|  | Submission file |  |
|  | Compiler |  |
|  | Timestamp |  |
|  | Starting time |  |
|  | Contest name |  |
|  | Rank | 2,7 |
|  | No of problems solved |  |
|  | Profile | 2,7 |
|  |  |  |

SELECTION CRITERIA

The potential classes were then selected as classes by six Selection Criteria. A potential class becomes a class when it fulfills all six characteristics.

1. Retain information: The potential class will be useful during analysis only if information about it must be remembered so that the system can function.
2. Needed services: The potential class must have a set of identifiable operations that can change the value of its attributes in some way.
3. Multiple attributes: During requirement analysis, the focus should be on “major” information; a class with a single attribute may, in fact, be useful during design, but is probably better represented as an attribute of another class during the analysis activity.
4. Common attributes: A set of attributes can be defined for the potential class and these attributes apply to all instances of the class.
5. Common operations: A set of operations can be defined for the potential class and these operations apply to all instances of the class.
6. Essential requirements: External entities that appear in the problem space and produce or consume information essential to the operation of any solution for the system will almost always be defined as classes in the requirements model.

|  |  |  |
| --- | --- | --- |
| **No** | **Noun** | **Selection criteria** |
|  | Authentication | 3 |
|  | Problem set | 1, 3-5 |
|  | Contest | 1, 3-5 |
|  | User | 1-5 |
|  | Administrator | 1-5 |
|  | Problem Solver | 1-5 |
|  | Submission | 1, 3-5 |
|  | Rank | 3-5 |
|  | Profile | 1, 3-5 |

ASSOCIATE NOUN AND VERB IDENTIFICATION

We will now identify the nouns and verbs associated with the potential classes to better find out the attributes and methods of each class.

|  |  |  |  |
| --- | --- | --- | --- |
| No | Potential class | Noun | Verb |
|  | Authentication | Problem solver, Administrator | Log in, sign up, recover account and log out. |
|  | Problem set | Problem name, time limit, question file, input file, solution file, problem id. | Set new problem, remove problem, display problem set. |
|  | Contest | Contest id, contest name, duration, starting time. | Set new contest remove contest, modify contest, display contests, show rank of a contest. |
|  | User | User name, password, recovery pin, user type, institute. | Create new user |
|  | Administrator | User name, password, recovery pin, user type, institute, designation. | Managing contest, managing problem |
|  | Problem solver | User name, password, recovery pin, user type, institute, registration number. | Participating in contest, submitting code, check submission status. |
|  | Submission | Submission language, verdict, submission code. | Receiving code, compiling and running code, giving verdict. |
|  | Rank | User, number of problems solved, contest id | Calculating number of problems solved by a user in a contest, calculating rank. |
|  | Profile | User | Show profile and modify profile. |

Attribute Identification

|  |  |  |
| --- | --- | --- |
| No | Name | Attribute |
|  | Authentication | username  password  recoveryPin  userType  institute |
|  | User | username  password  recoveryPin  userType  institute |
|  | Administrator | username  password  recoveryPin  userType  institute  designation |
|  | Problem solver | username  password  recoveryPin  userType  institute  registrationNumber |
|  | Contest | contestId  contestName  startingTime  duration |
|  | Problem | problemName  timeLimit  questionFile  inputFile  solutionFile |
|  | Submission | submissionLanguage  verdict  submissionCode |
|  | Rank | user  numberOfProblemsSolved  contest id |
|  | Profile | user |

Method Identification

|  |  |  |
| --- | --- | --- |
| No | Class | Methods |
|  | Authentication | * signUp() * login() * recoverAccount() * logout() |
|  | Problem Set | * addProblem() * removeProblem() * displayProblem() |
|  | Contest | * showContest() * createContest() * removeContest() * modifyContest() |
|  | User | * createUser() |
|  | Administrator | * prepareContest() * removeContest() * modifyContest() * showResult() * displayProblem() |
|  | Problem solver | * checkSubmission() * submitCode() * participateInContest() |
|  | Submission | * matchWithSolutionFile() * compileCode() * runCode() * giveVerdict() * receiveCode() |
|  | Rank | * showRankOfContest() |
|  | Profile | * modifyProfile() * showProfile() |

Class card

|  |  |
| --- | --- |
| Authentication | |
| Attribute | Method |
| * username * password * recoveryPin * userType * institute | * signUp() * login() * recoverAccount() * logout() |
| Responsibilities | Collaboration |
| * Register a user * Login to the system * Recover a user account * Logout from the system | Validation  DBConnector |

|  |  |
| --- | --- |
| UserProfile | |
| Attribute | Method |
| * username * password * recovery pin * userType * institute | * modifyProfile() * showProfile() |
| Responsibilities | Collaboration |
| * Showing user profile * Modifying user profile | Validation  DBConnector |

|  |  |
| --- | --- |
| Problem solver | |
| Attribute | Method |
| * Registration number | * checkSubmission() * submitCode() * participateInContest() |
| Responsibilities | Collaboration |
| * Display the submissions * Submit code * Participate in contest | Contest  Submission  DBConnector |

Note: Problem solver will extend UserProfile class.

|  |  |
| --- | --- |
| Contest | |
| Attribute | Method |
| * contestId * contestName * startingTime * duration | * showContest() * prepareContest() * removeContest() * modifyContest() * showRankOfContest() |
| Responsibilities | Collaboration |
| * Showing all contests * Creating new contest * Removing a contest * Updating a contest * Showing rank of a contest | Problem  Validation  DBConnector |

|  |  |
| --- | --- |
| Problem | |
| Attribute | Method |
| * problemName * timeLimit * questionFile * inputFile * solutionFile | * addProblem() * removeProblem() * displayProblem() |
| Responsibilities | Collaboration |
| * Adding a new problem * Removing a problem * Displaying problem of a contest. | Validation  DBConnector |

|  |  |
| --- | --- |
| Validation | |
| Attribute | Method |
|  | * validateInput() |
| Responsibilities | Collaboration |
| * Validate data | DBConnector |

|  |  |
| --- | --- |
| Submission | |
| Attribute | Method |
| * submissionLanguage * verdict * submissionCode | * matchWithSolutionFile() * compileCode() * runCode() * giveVerdict() * receiveCode() |
| Responsibilities | Collaboration |
| * Compiling the submitted code * Running the compiled code * Matching with solution file * Giving a verdict | DBConnector |

|  |  |
| --- | --- |
| DBConnector | |
| Attribute | Method |
|  | * connectToDatabase() |
| Responsibilities | Collaboration |
| * connecting to database |  |