ONLINE CODING PLATFORM

A MINI PROJECT REPORT

Submitted by

Arun V. S

20BIT0233

Chavan Mukul Manish

20BIT0238

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING

VIT UNIVERSITY: VELLORE 632014

NOVEMBER, 2022

APPENDIX 2

VIT UNIVERSITY: VELLORE 632 014

BONAFIDE CERTIFICATE

Certified that this project report **" ONLINE CODING PLATFORM"** Is the bonafide work of **" Arun V.S , Chavan Mukul Manish"** who carried out the project work under my supervision.

Signature of the Guide

Dr.GUNASEKARAN G

GUIDE

Senior Professor

SITE

Vellore Institute of Technology, Vellore

TABLE OF CONTENTS:

Abstract

Online programming simulators have significant potential in organizing an effective distance learning system in Ukrainian universities. It is important to use online simulators in the learning process as an additional tool for the formation of professional competencies, which provides more intensive involvement of students in the process of writing code and practical (situational) application of existing knowledge. Gamification of the process of training future IT specialists helps to increase cognitive activity, and hence – the quality of the educational process and distance learning in particular. The authors recommend the use of online programming simulators as an additional tool for teaching computer science disciplines, taking into account their functionality, as well as the level of preparation of students and the expected learning outcomes.

The preliminary component of the interview process is testing the technical skill of a candidate, most of the candidates are in remote locations, so the only access they will have is the internet, so a online coding platform is the best option to test the candidate's technical skill. In this project, the component of the online coding platform is broken down and analyzed.

ii. LIST OF TABLES

iii. LIST OF DIAGRAMS

1. INTRODUCTION

1.1 Motivation

Today's market is replete with a myriad of online tech hiring platforms that enable swift and seamless technical hiring, offering numerous advantages. But, with many online coding assessment players in the market, the selection for the right tech hiring platform must be more precise.

Hence, your shortlisting process should depend on a player's ability to offer 'three pillars' that simplify and scale your assessment and hiring process to make informed decisions.

It includes:

Expansive Coverage of the Topic:

Technical skills are extensive and ever-changing. Hence, when shortlisting a vendor providing a technical hiring platform, ascertain the range of programming languages and frameworks they provide for all significant coding job roles. The tech recruiting software should be diverse to cover every existing and new-age language, framework, technology, topics, and skills requirements. Also, assess their ability to provide accurate technical assessments for micro-skills and concepts within coding system and development. In many ways, this provides an accurate picture of a candidate's all-around fit for a required position.

Combination of Knowledge and Application-Based Questions:

An eclectic mix of expertise and application-based questions is the future of candidate assessments and recruitment. They cumulatively offer a holistic overview of a programmer's aptitude to take on a variety of existing and future coding tasks. Knowledge-based questions are your first filtration process that shortlists based on essential knowledge, concept, application, and analysis. Afterward, the selected few are assessed practically via application-based questions that provide a real sense of candidates' ability by asking them to mirror real-world challenges. As candidates create codes from scratch, a stimulating attribute of the best recruitment software is swift auto-grading to eliminate bias and ensure evidence-based hiring decisions.

Efficient Online Coding Platform:

An efficient online coding platform is the third pillar that enables quality hiring at scale. Choose a vendor based on a dynamic platform that is comprehensive, user-friendly, and offers an easy-to-use interface for the candidate and the assessor alike. As a recruiter, you must be able to schedule tests, screen candidates, and roll out online interviews, seamlessly, with a clear oversight on progress and performance. Pick the top online coding platform based on provisions, such as customization, hands-on skill support, swift auto-evaluation, scalability, a suite of anti-cheating solutions, result validation and proficient tracking of the tech hiring funnel. The best coding platform should smoothly integrate and be location-agnostic for all. Consistency and support promptness throughout the platform's usage is another critical way of assessing the best coding platform.

A substantial technical hiring platform fine-tunes your assessment and recruitment requirements by cumulating all the key features mentioned above. Hence, the absence of these features leads to an incomplete evaluation and hiring, consequently an irrelevant service provider.

1.2 What is OOAD and UML?

Object oriented methodology defines a method to develop an application, software or a system. The process starts with an analysis where the user requirement is understood by the developer. After analysis, in the designing stage, an application model is created where more details are added to the model.

The last stage is an implementation where the programmer translates the model into a system using programming language, database, or hardware. This process is seamless which means that no information is lost when the developers switch from one stage to another.

Unified Modeling Language (UML) is a general purpose modelling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.

UML is not a programming language, it is rather a visual language. We use UML diagrams to portray the behavior and structure of a system. UML helps software engineers, businessmen and system architects with modelling, design and analysis.

1.3 SYSTEM DEVELOPMENT MEDTHODOLOGY

Object Oriented Software Development Life Cycle Model

It involves the following:

1. Object Oriented Analysis Phase

2. Object Oriented Design Phase

3. Object Oriented Construction Phase

4. Object Oriented Testing Phase

5. Object Oriented Maintenance Phase

1.4 Literature Survey

Title	Author	Content
A Framework for Personalized Competitive Programming Training	Tania Di Mascio, Luigi Laura, Marco Temperini	Different Architecture platforms, various platforms used, types of contests
HackerRank Architecture	Sunil Tej	HackerRank's architecture overview
Interactive Coding Platform for Students	T.K. Chandru, M. Dinesh Kumar, S. Karthikeyan, K. Saranya	Existing systems in use and architecture of those platforms and their features,
A coding platform: For all programming labs and practical examinations.	Bhanuse, Roshan Bawankar, Uttkarsh Sharma, Durgesh M. Patle, Mayuresh Narsapurkar, Vedashree Sawate, Shubham	In this system, all the evaluation processes are automated. Thus, creating a trouble-free platform for teachers and a judicious assessment for students
A Pedagogical Analysis of Online Coding Tutorials	Ada S. Kim, Andrew J. Ko	Strengths and weaknesses of online coding tutorials,

		opportunities for improvement..

Table 1.1 Literature Survey

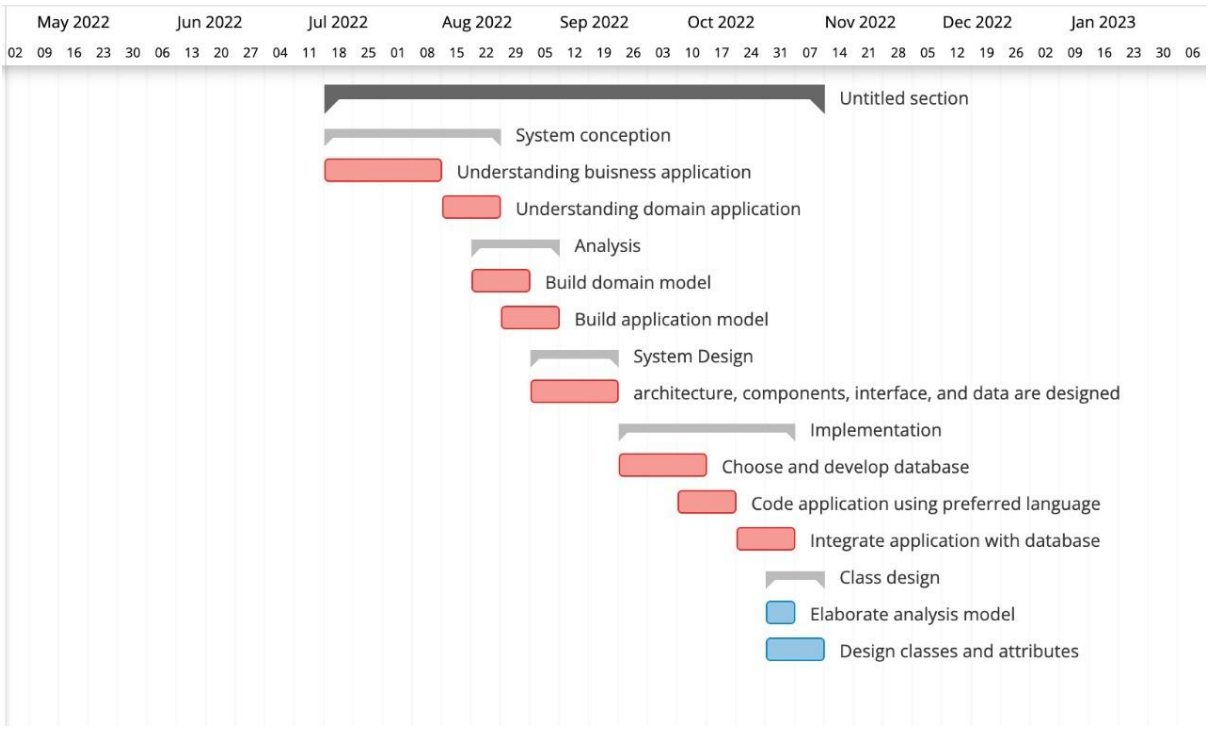
2. OBJECT ORIENTED REQUIREMENT WORKFLOW

2.1 Problem Statement

The primary recruiting practice for tech companies nowadays is a coding round that potential employees must go through; the user must think out of the box for a solution and pass all test cases provided and the platform must be made in such a way that all possible answers are accepted as the logic/ the method used to arrive at the solution can be different in each case. The problem statement asked should represent some real-world problem where the user should be able to apply and test it in code.

The main motive behind the coding platform is to help the user simulate that test-like experience and prepare themselves for the coding interview, expose them to a community that helps them clear doubts, get tips/ pointers from those who have already gone through the process and share their experiences. The platform can also provide tutorials on specific topics to help the user learn and understand better. The user should be able to track/ view his/her progress and plan their journey accordingly, this model follows a freeware software model.

2.2 COMPLETE PROJECT SCHEDULE



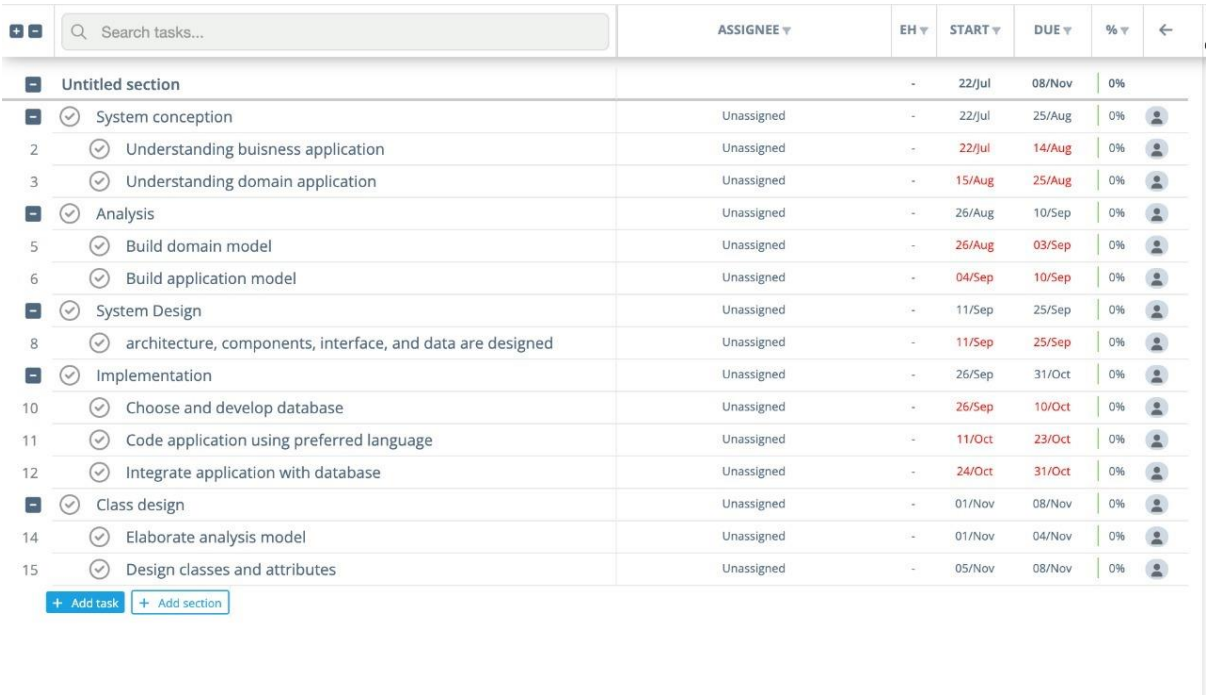


Fig 2.1 PROJECT SCHEDULE GANTT CHART

2.3 Glossary

S.no	Term	Glossary
1.	Coder	User of the platform
2.	Test case evaluator	Software which contains all the testcases to be evaluated
3.	Problem	The coding question given
4.	Courses	Names of various courses available, i.e different languages
5.	Certificate	Final certificate given to user after completion of a course
6.	Compete	Competing in a global coding challenge where all have the same question but the user to solve under the lowest time wins
7.	Prep	Question bank which contains all interview questions
8.	Leaderboard	Leaderboards which contain user who have completed the greatest number of competitions
9.	Bookmark	Bookmark course or challenges to complete later, i.e do some other course for a time and come back to it later.

10.	Submissions	Submissions made for each challenging program

Table 2.1 Glossary

2.4 Flow chart for Object Oriented Requirement Workflow

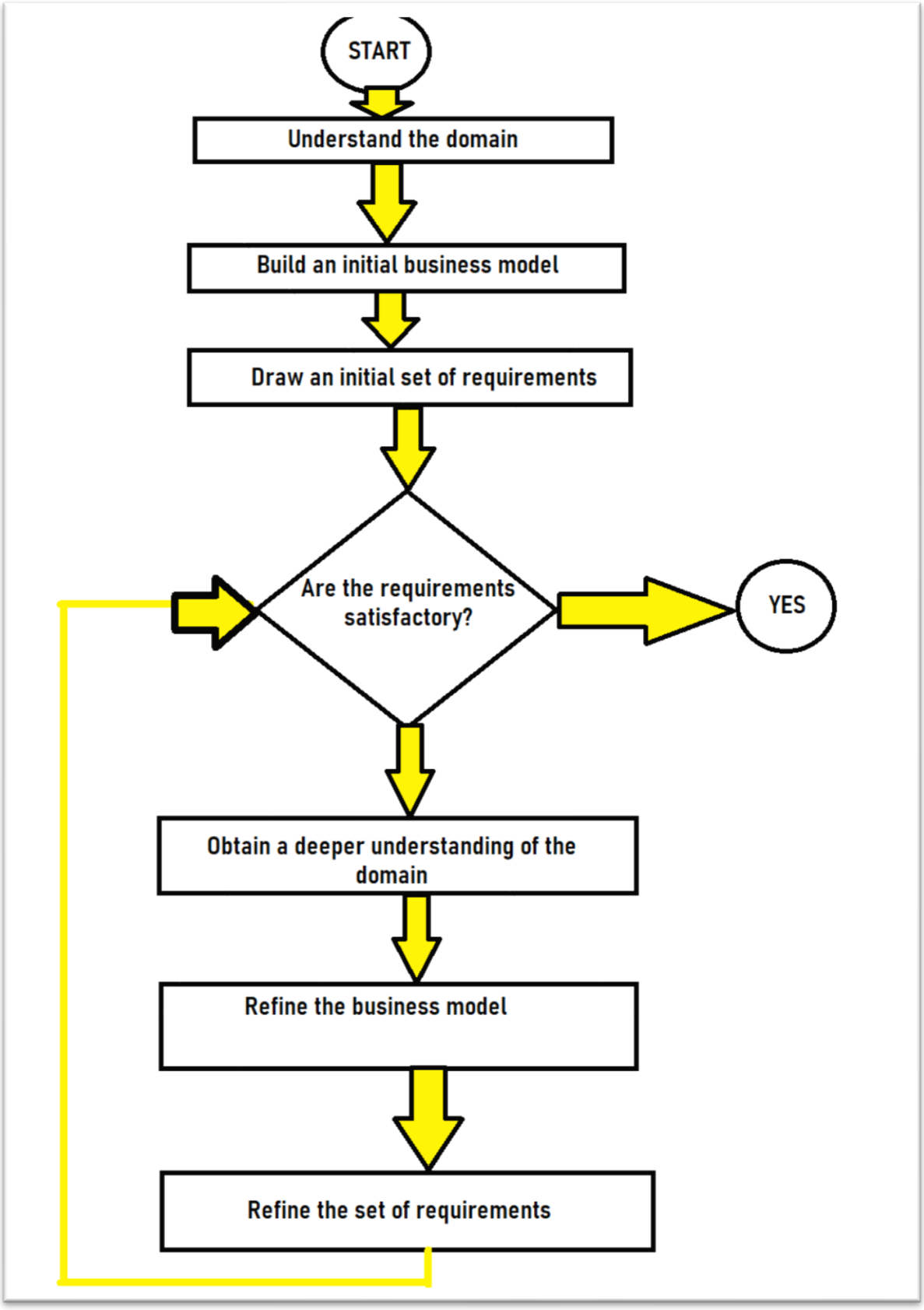


Fig 2.2 Flow chart for Object Oriented Requirement Workflow

2.5 Business Model

2.5.1 Use Case Model

a. List the actors

1. Student
2. Job-seeker
3. Educational Institution
4. Company
5. Server_side
6. Platform Team
7. Output_checker
8. Compiler
9. Database

Use cases:

1. Register
2. Login
3. Solve problem
4. Search
5. View profile
6. View leaderboard
7. Manage interview preparation schedule
8. Hire users
9. Manage problem questions
10. View Timer
11. Complaint and feedback
12. Logout

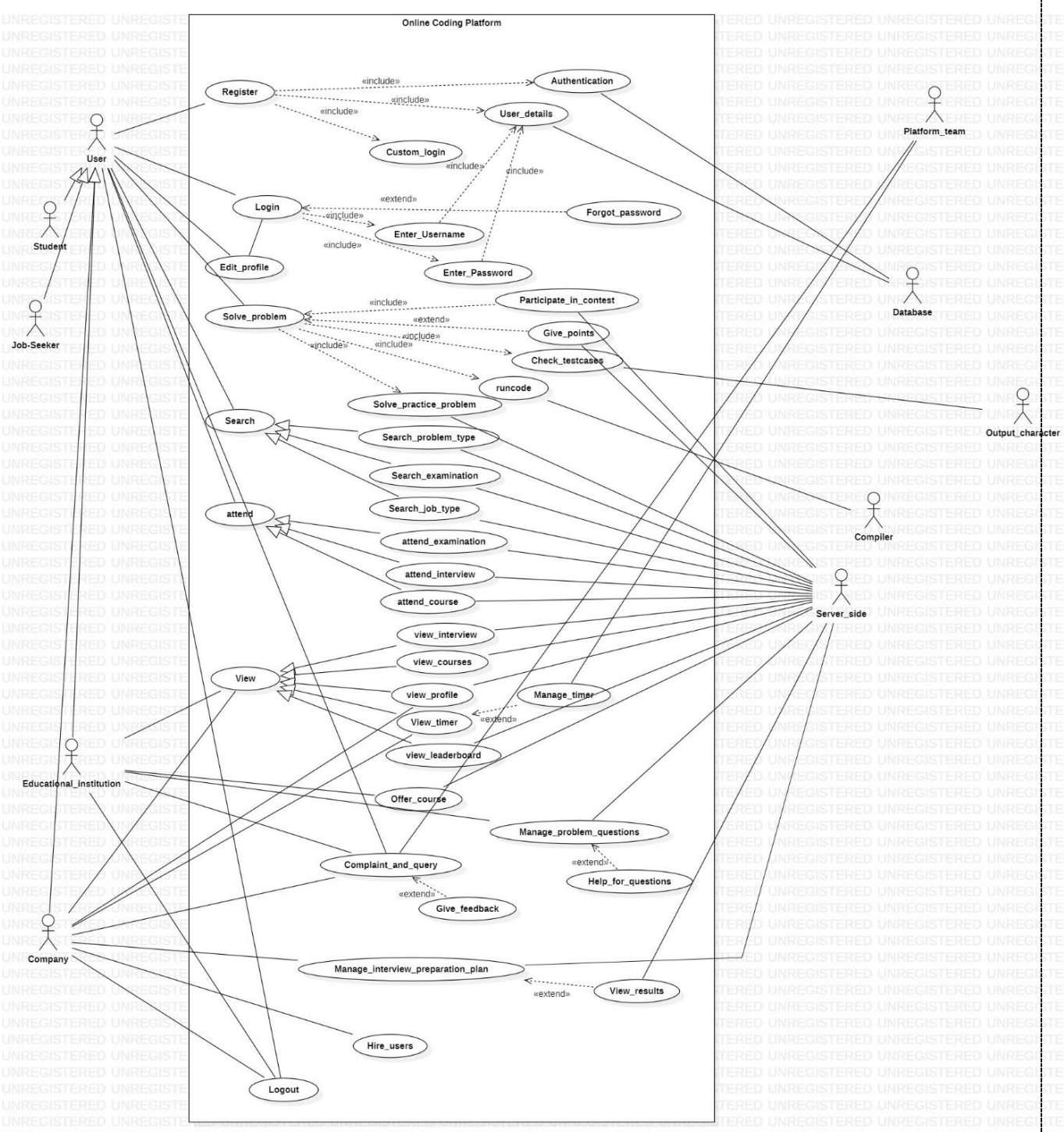


Fig 2.3 Use case model

2.3.2 Use Case Description

1. Login

2. Edit profile

3. Complaint_and_query

4. Manage_problem_question

5. View (includes view interview, view profile, view course and view timer)

Note: End-user: user (student, job-seeker), educational institution and

company.

1.USE CASE SPECIFICATION: LOGIN

Use Case name:
The name of the use case is Login.
Brief Description
This use case will be initiated by the users comprising the students as well as job seekers, educational institutions and Company. Same use case is used for both the developers and company because the register use case is decomposed to hold unique domains using custom_login. This use case will be used by everyone who enters the portal to login into the system to proceed with their requirement.

Table 2.1 USE CASE SPECIFICATION: LOGIN

2. USE CASE SPECIFICATION: EDIT-PROFILE.

Use Case name:
The name of the use case is Edit profile
Brief description
This use case will be used by the end users to edit their profile.

Table 2.2 USE CASE SPECIFICATION: EDIT-PROFILE.

3. USE CASE SPECIFICATION: MANAGE_PROBLEM_QUESTIONS

Use Case name:
Manage_problem_questions
Brief Description:
This use case is initiated by the Educational institution, where he/she can Add, Edit, Delete problem questions of his/her wish.

Table 2.3 USE CASE SPECIFICATION: MANAGE_PROBLEM_QUESTIONS

4. USE CASE SPECIFICATION: COMPLAINT_AND_QUERY

Use case name:
Complaint_and_query
Brief Description:
This use case is initiated by the Users, where he/she can raise a complaint, queries, suggestions about our platform.

Table 2.4 USE CASE SPECIFICATION: COMPLAINT_AND_QUERY

5. USE CASE SPECIFICATION: VIEW

Use case name:
The name of the use case is view (credential)
Brief Description:
This use case will be used by all the end -users to view Interview schedules, courses offered, profile of other end-users and timer functionality.

Table 2.5 USE CASE SPECIFICATION: VIEW

2.5.3 Initial UML diagrams

2.5.3.1. Class diagram

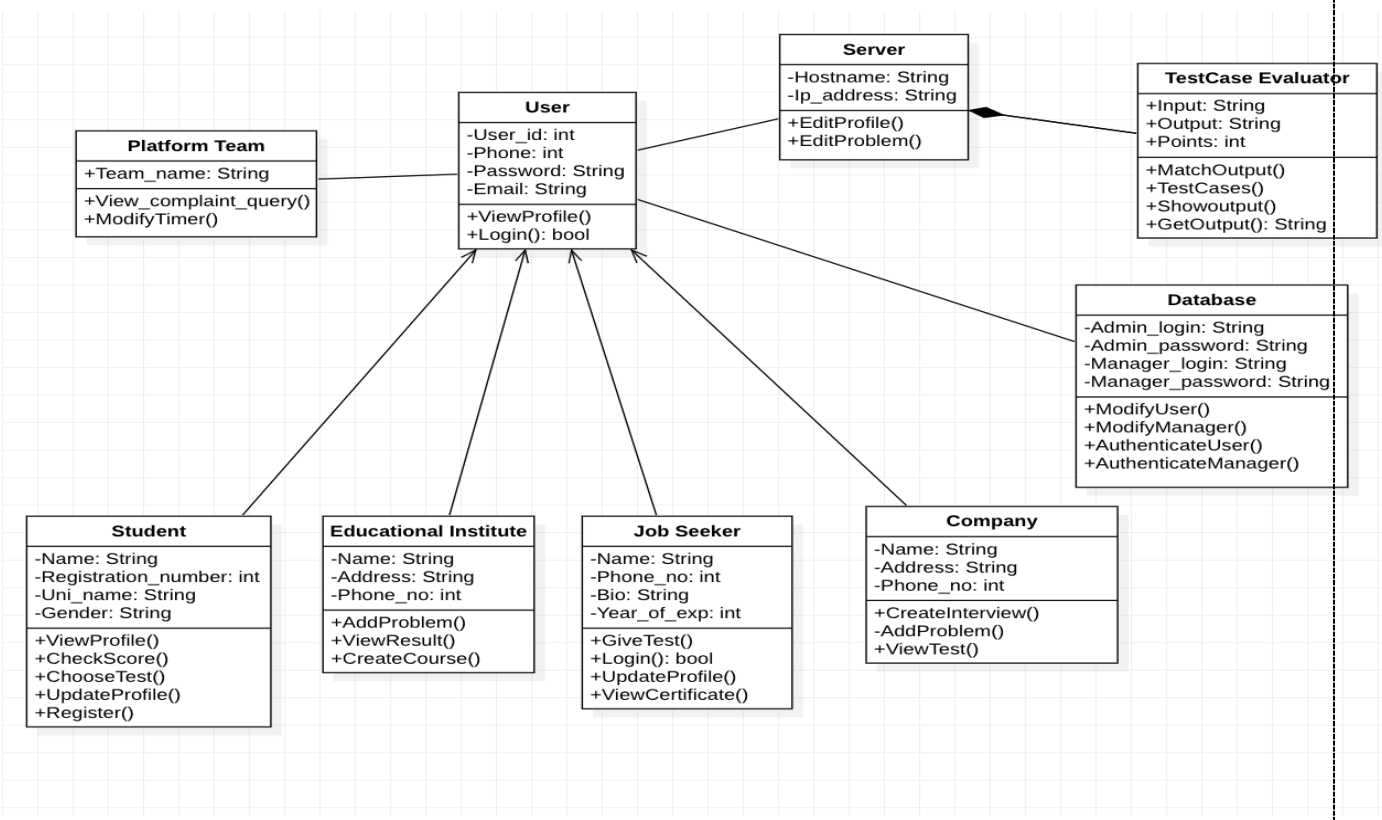


Fig 2.4 Class diagram

2.5.3.2. Object diagram

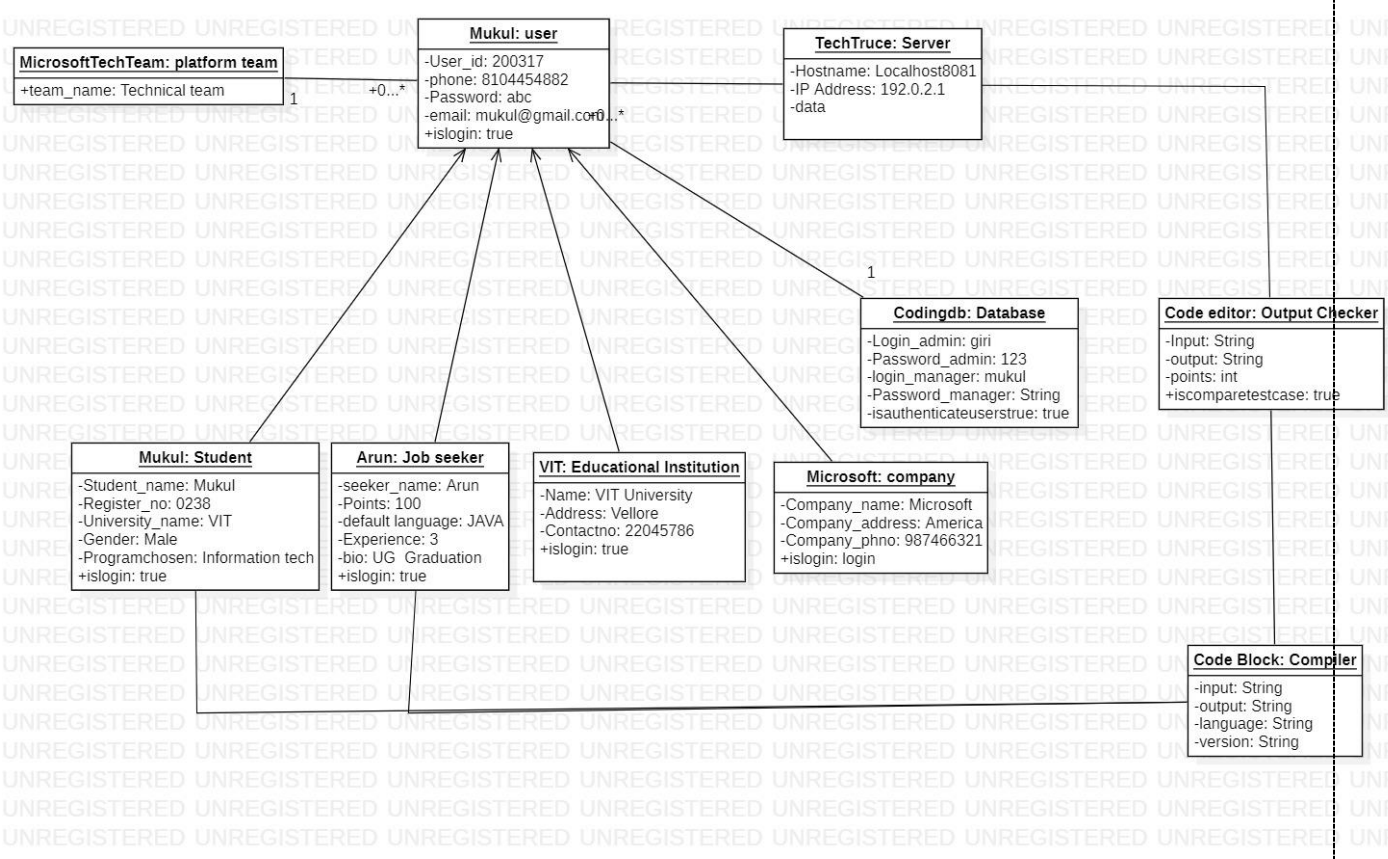


Fig 2.5 Object diagram

2.5.3.3. Package diagram

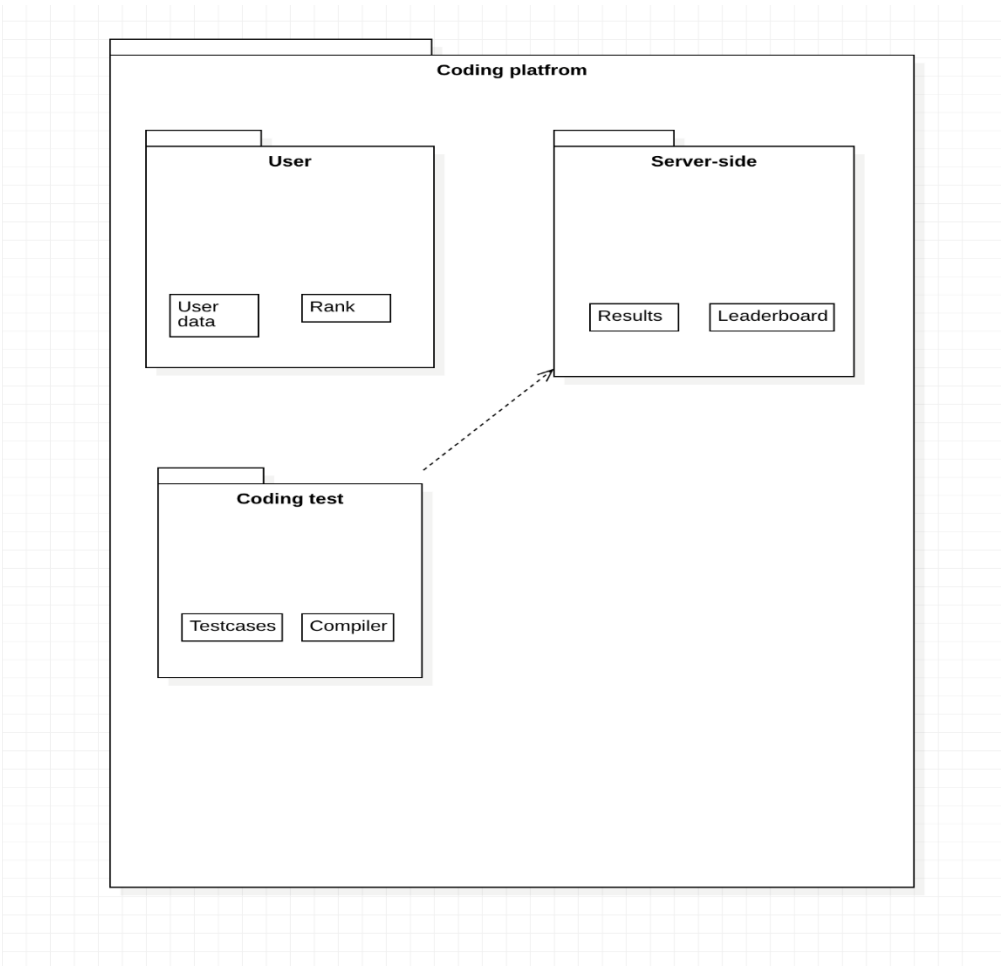


Fig 2.6 Package diagram

2.5.3.4 Sequence diagram

1. Login

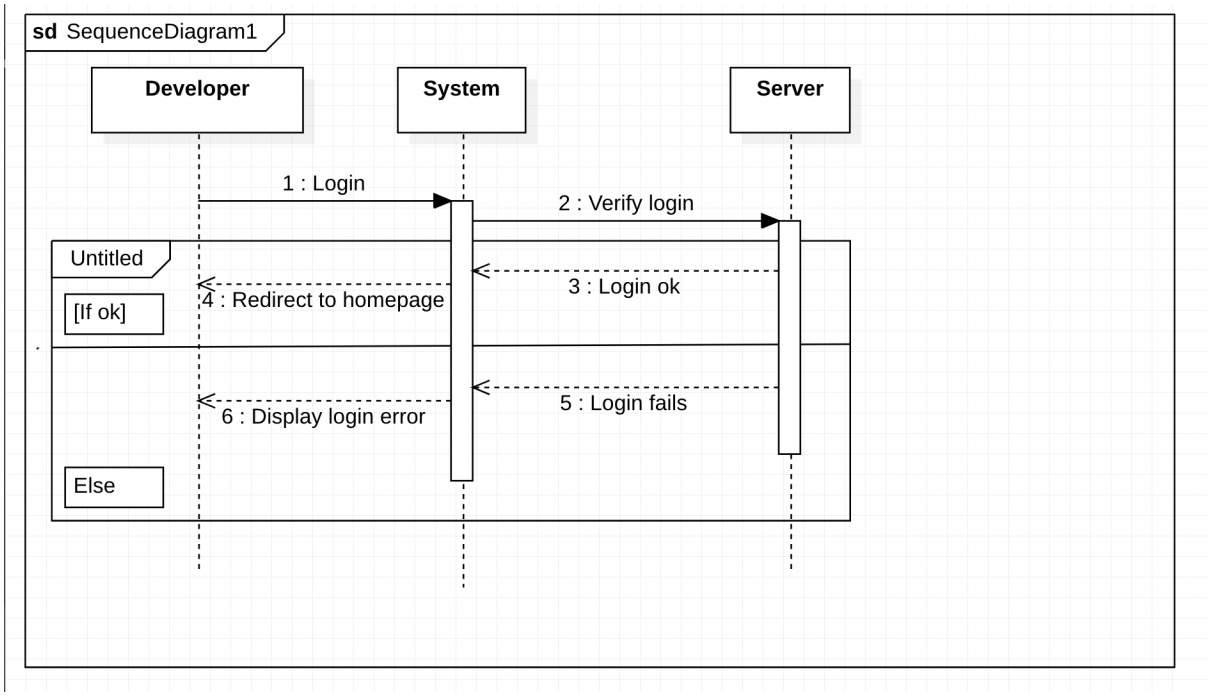


Fig 2.7 Login

2. Participate contest:

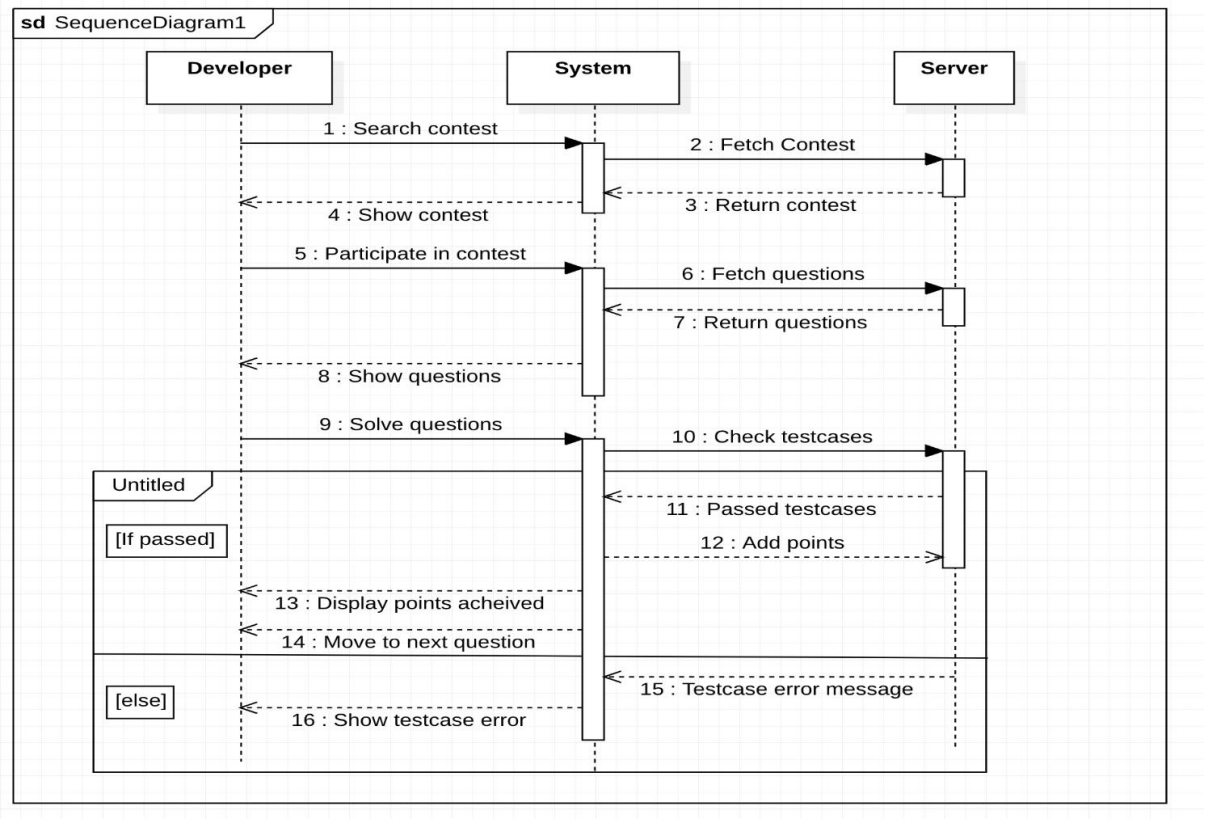


Fig 2.8 Participate contest:

2.3.3.5 Collaboration diagram

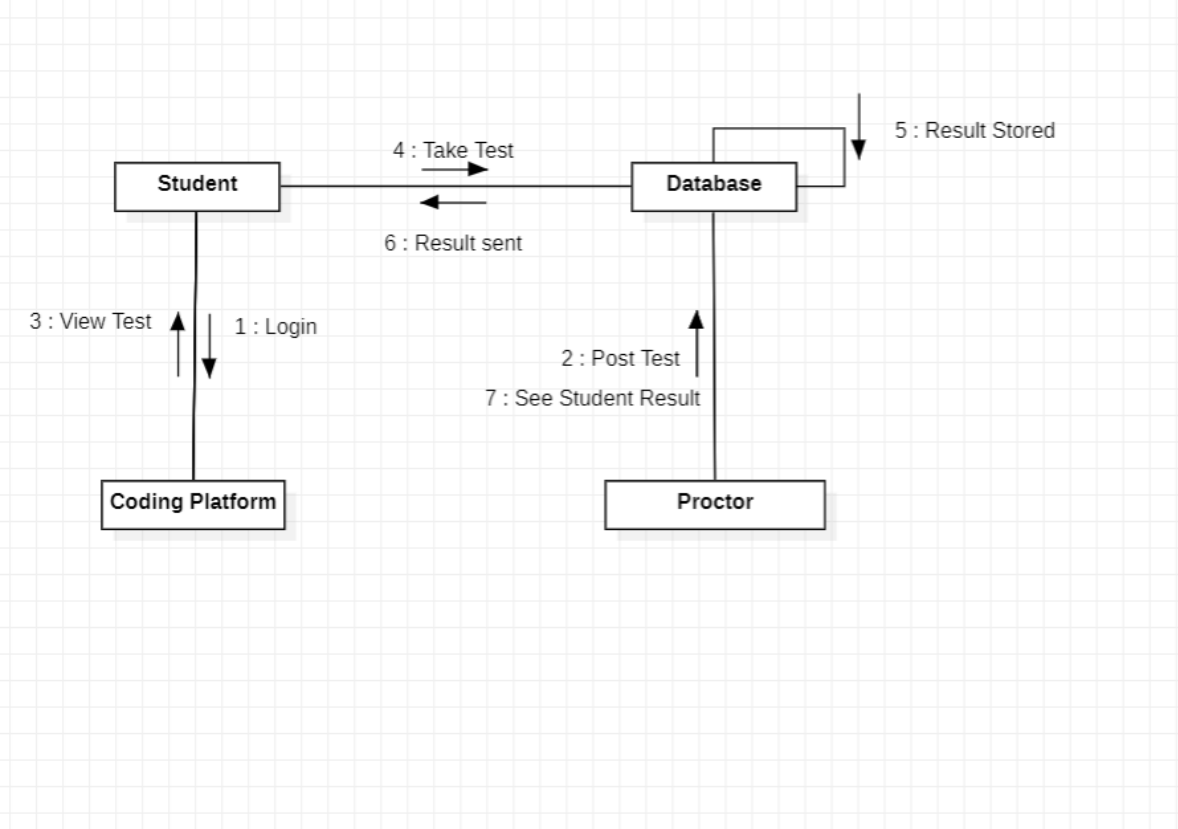


Fig 2.9 Collaboration diagram

2.5.3.6 State chart diagram

1. Edit profile

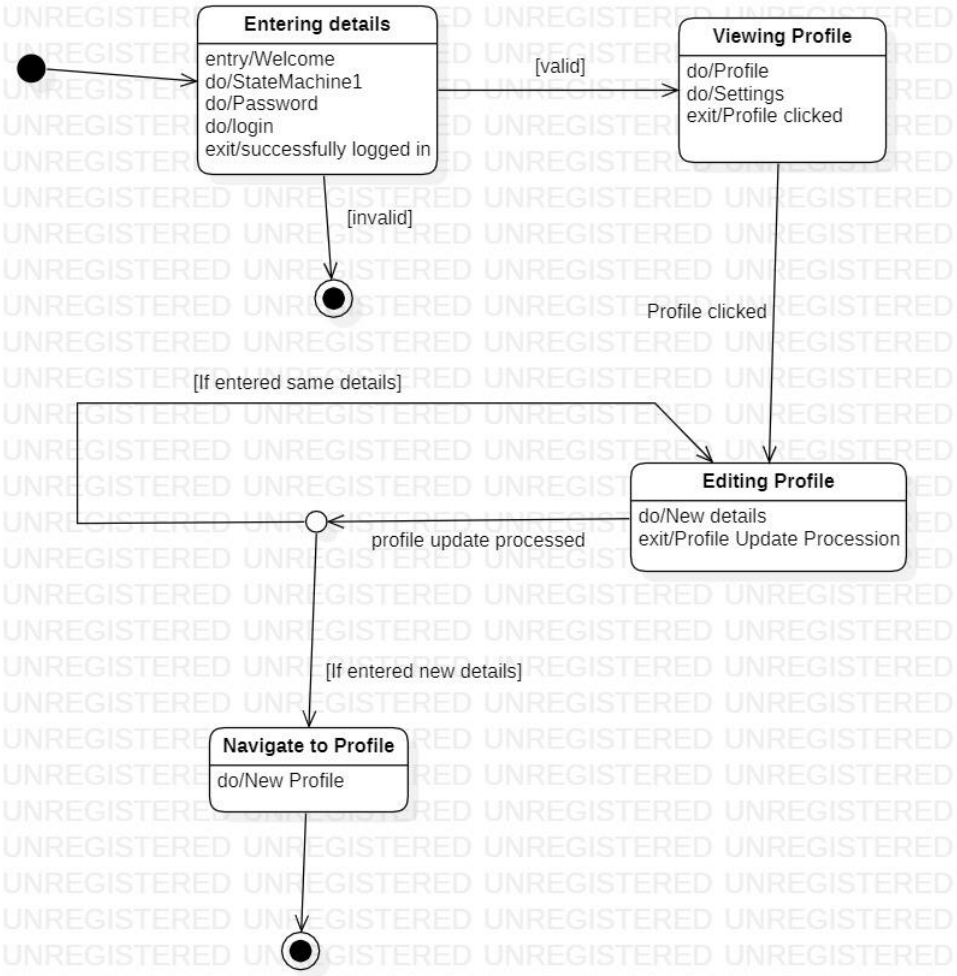


Fig 2.10 Edit profile

2. Feedback and complain

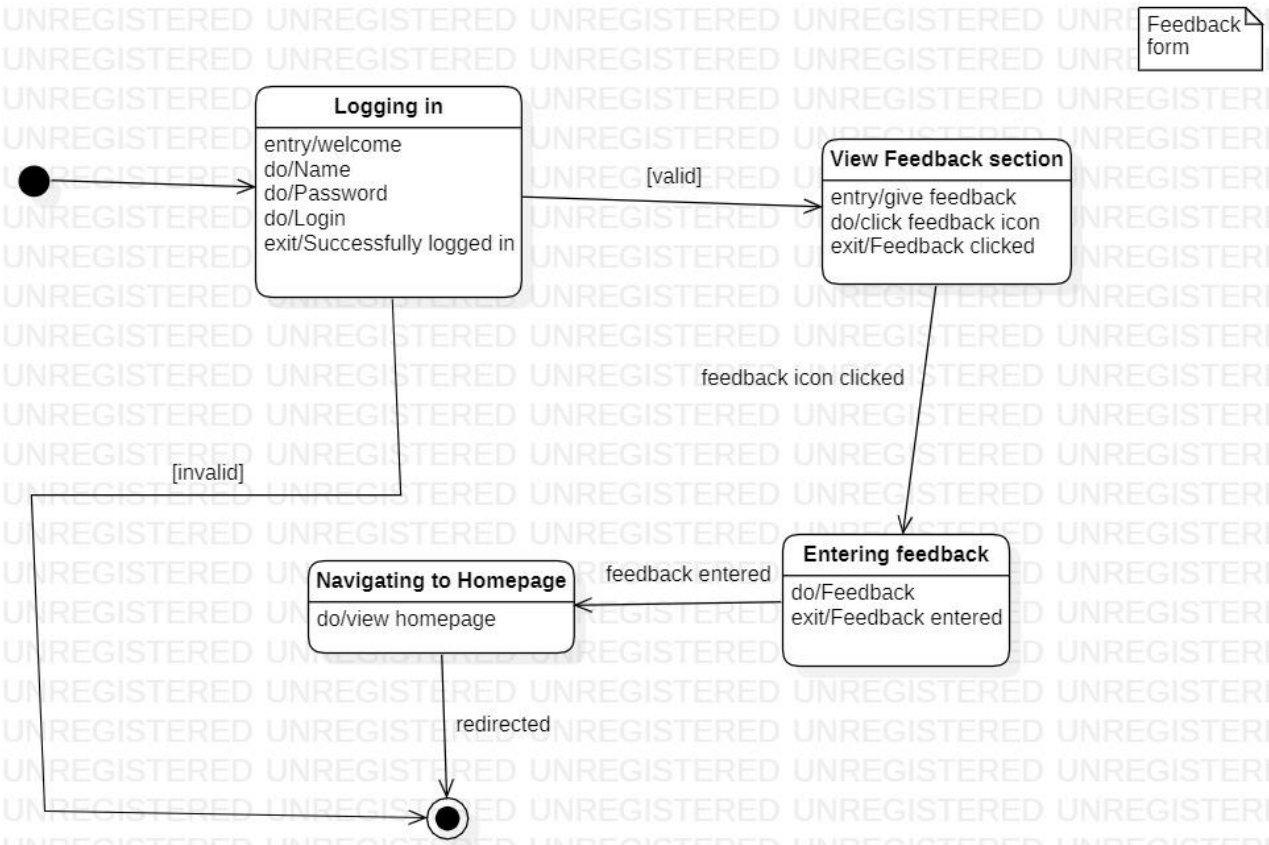


Fig 2.11 Feedback and complain

2.5.3.7 Activity diagram

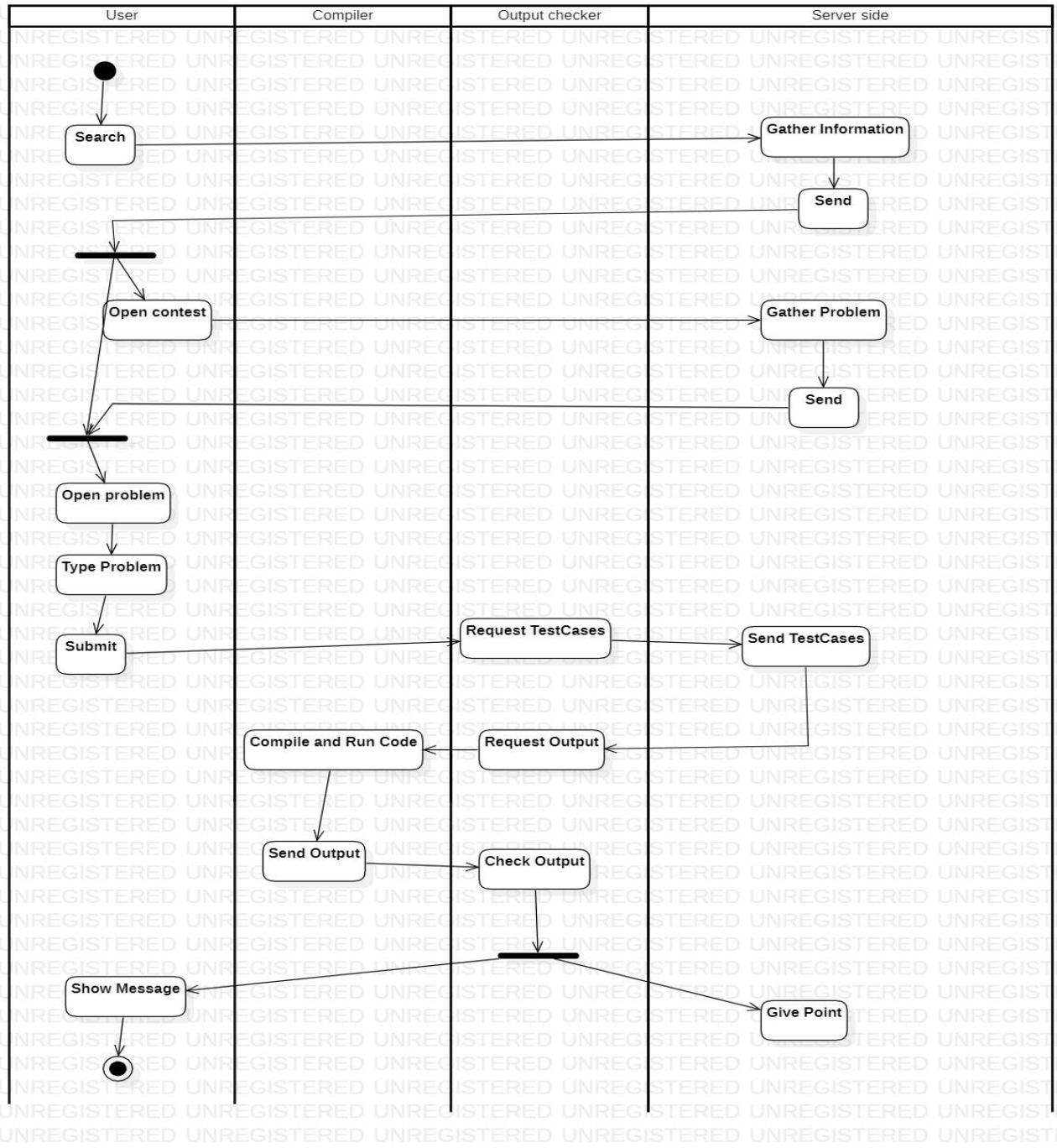


Fig 2.12 Activity diagram

2.5.3.8 Component diagram & deployment diagram

Deployment diagram:

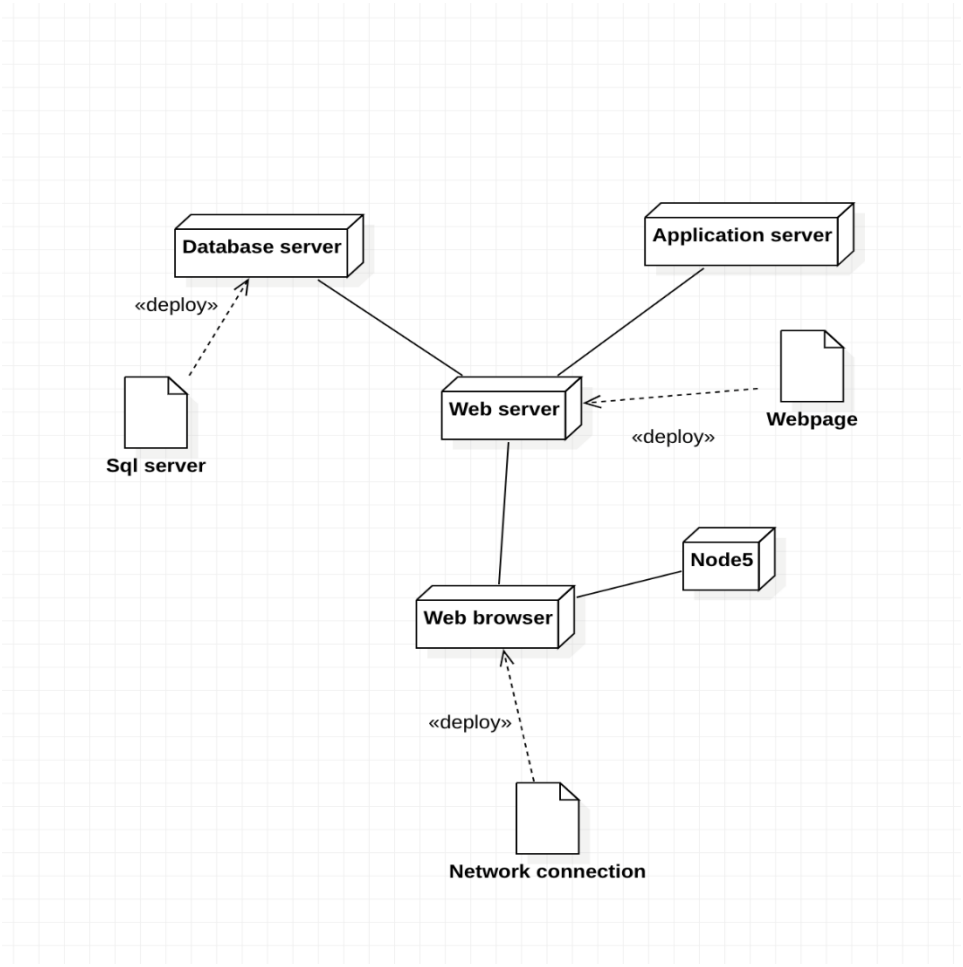


Fig 2.13 deployment diagram

Component diagram:

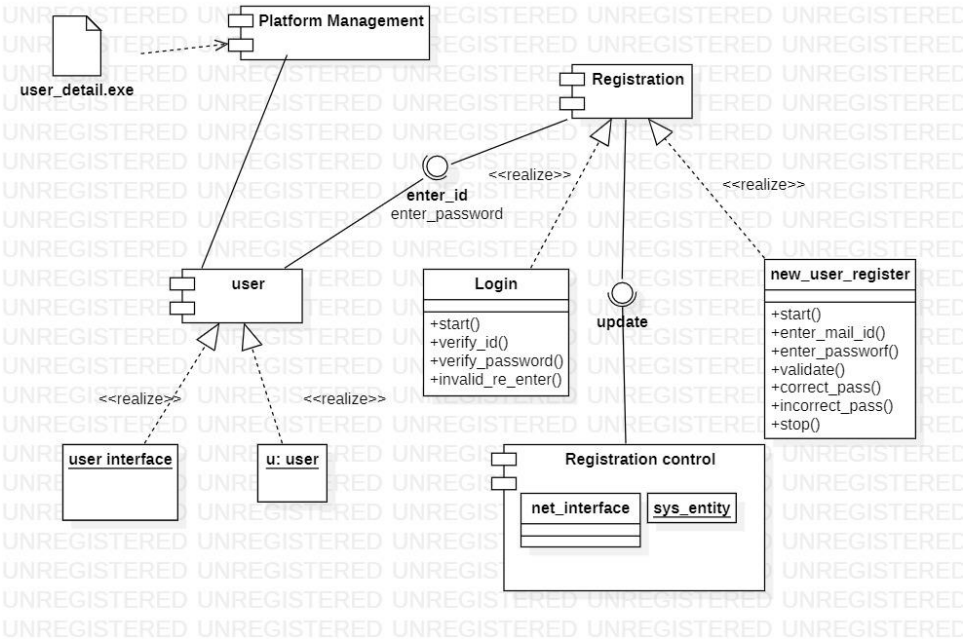


Fig 2.14 Component diagram

2.6 Initial set of requirements

HARDWARE REQUIREMENTS

Processor: Pentium IV and above

RAM: 512 MB or above

Hard Disk: 40 GB or above

Input Devices: Keyboard and Mouse

Output Devices: Monitor, Printer

SOFTWARE REQUIREMENTS

Operating System: Windows XP or higher

Memory: The software is expected to use not more than 150 MB of RAM and 0.75 GB of internal storage for the server client application. However a database server to store all the book

and customer details is required as per the needs of the online book store.

Formally, all functional requirements can be subdivided into two broad categories: those that allow the user to study from site at the stage of coding and those that “push” them to make a purchase of the related courses.

The first category includes the development of search tools, sorting, filtering, navigation, as well as the visual component of the site.

The second category includes the study of user course interactions, the application form for coding, all kinds of elements of preset code, programming language help etc.

Non functional requirements:

1. A website should be capable enough to handle 20 million users with affecting its performance
2. The software should be portable. So moving from one OS to other OS does not create any problem.
3. Privacy of information, the export of restricted technologies, intellectual property rights, etc. should be audited.

Functional requirements:

1. Fr1: As a user, it will need a separate sign-up page with name, date of birth, qualification, an email address, phone number and a password as a first time user.

2. Fr2: As a Server_side, it will need a password length longer than 15 characters. If the user enters more than 15 characters while choosing the password, an error message shall ask the user to correct it.

3. Fr3: User need a captcha to be solved every time while logging in

4. Fr4: Server_side need a check box to be checked for verifying whether it is a bot or a human.

5. Fr5: User shall manage the profile by editing, deleting, updating the profile name, email address, password, qualification.

Non-functional requirements:

1. Nf1: The system should allow scrolling one page up or down in a window shall take at most 1 second.

2. Nf2: As a Payment service provider, I shall provide the safe and secure transaction between user and system by keeping encryptions.

3. Nf3: As an auditor, I shall review and verify the accuracy of financial records and ensure that the system complies with tax laws.

4. Nf4: The system should be able to process 100 payment transactions per second in peak load.

5. Nf5: The system should ensure that no more than 1 per 10000 transactions shall result in a failure requiring a system restart.

6. Nf6: The system shall allow installation of a new version shall leave all database contents and all personal settings unchanged.

7. Nf7: The system should ensure that estimated loss of data in case of a disk crash shall be less than 0.1%

3. OBJECT ORIENTED ANALYSIS WORKFLOW

3.1 Refinement of Object-Oriented Requirement Workflow

Most conventional or traditional methods are found to restrict programmers from thinking out of the box. One of the sayings is very accurate i.e., Answering the question is much easier than framing one, users when exposed to different types of test cases can meet every possible question that needs a correct answer. Due to their confined knowledge, many jobseekers lack understanding the real-world problem-solving skills. This platform was created keeping in mind that everyone who codes here has two motives, one is to learn the concept by applying and testing it as code and another one is vice versa. Practice makes a man perfect, but it is very important to know how to progress, through this journey. How to challenge ourselves to meet the dream-big companies. Online coding platforms also make a channel for many companies who are in search of efficient programmers. In this way, companies can even get a great talented programmer who was once sitting in a corner of his/her room somewhere else in the world. Talents are scattered everywhere; online platforms are acting as a powerful weapon to spotlight

them. The motive behind this coding platform is to expose the programmer community to the real world, sharing knowledge and helping the companies to find them.

3.2 Flow chart for extracting classes

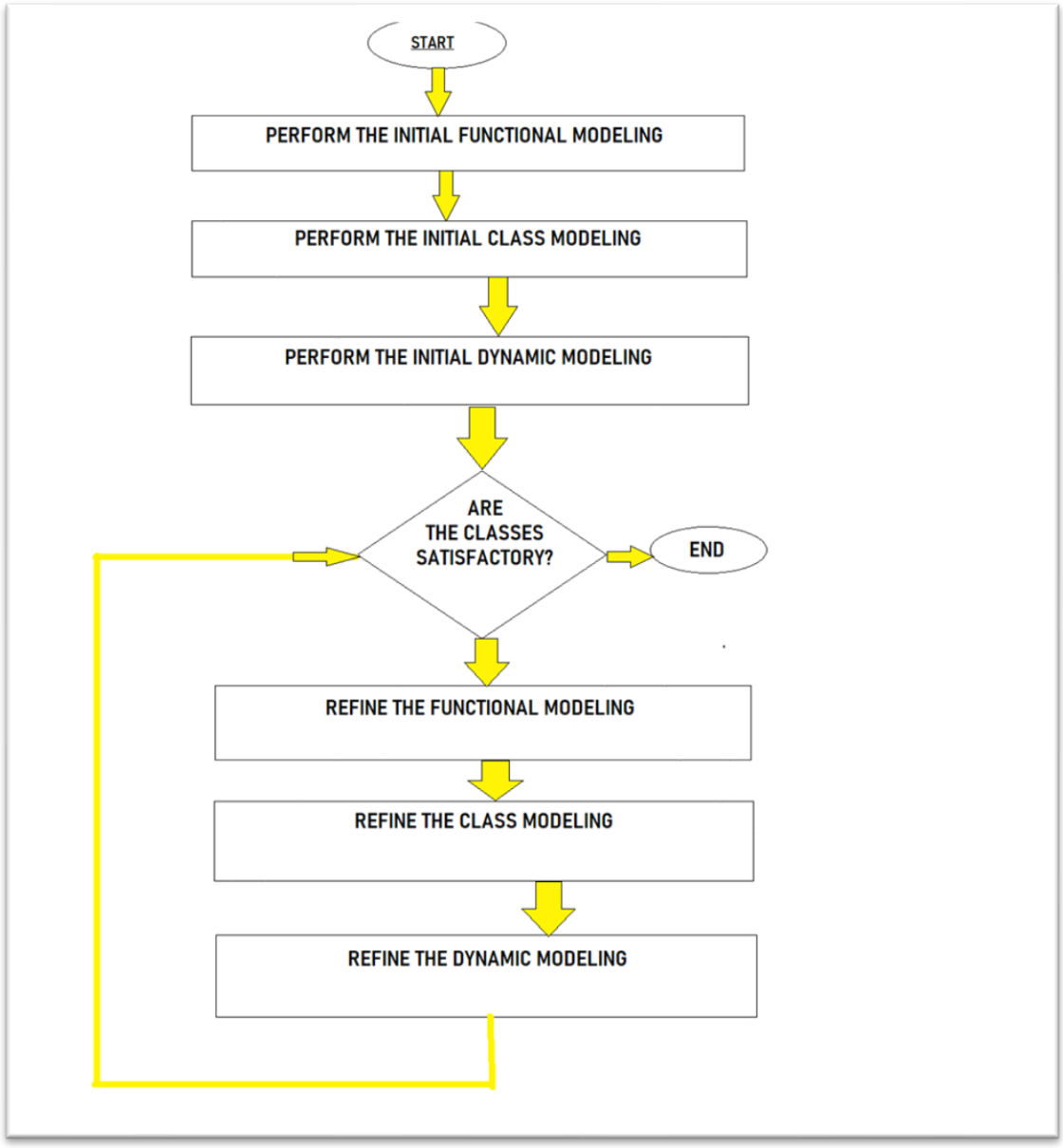


Fig 3.1 Flow chart for extracting classes

3.3 Functional Modelling

3.3.1 Normal Scenario

3.3.1.1.USE CASE SPECIFICATION: LOGIN

LOGIN
1. This User/company/educational institution will enter his/her login credentials which are provided to them after registration. 2. The system will validate the login credentials entered by them. 3. If the login credentials are valid, the system will allow you to login in to the system. 4. The use case ends.

Table 3.1 USE CASE SPECIFICATION: LOGIN

3.3.1.2. USE CASE SPECIFICATION: EDIT-PROFILE.

EDIT-PROFILE
1. After logging in successfully, the end user is given an option to edit their profile. 2. end-users can edit their profile. 3. For confirmation, an OTP will be sent to user's registered mobile number. 4. If the OTP is valid, the profile of the user will be updated. 5. The use case ends.

Table 3.2 USE CASE SPECIFICATION: EDIT-PROFILE.

3.3.1.3. USE CASE SPECIFICATION: MANAGE_PROBLEM_QUESTIONS

MANAGE_PROBLEM_QUESTIONS
1. This use case begins when the Educational institution logs on to the Online Coding Platform by entering his/her password. 2. The system verifies that the password entered is correct (E-1), and prompts the user to proceed with the following activity: ADD, EDIT, DELETE. 3. If the activity selected is ADD, the S1: ADD problem question sub flow is performed. 4. If the activity selected is EDIT, the S2: EDIT problem question sub flow is performed. 5. If the activity selected is DELETE, the S3: DELETE problem question sub flow is performed.

Table 3.3 USE CASE SPECIFICATION: MANAGE_PROBLEM_QUESTIONS

3.3.1.4. USE CASE SPECIFICATION: COMPLAINT_AND_QUERY

COMPLAINT_AND_QUERY
1. This use case begins when the Educational institution logs on to the Online Coding Platform by entering his/her password. 2. The system verifies that the password entered is correct (E-1), and then prompts the user to give complaints, suggestions. 3. The user has to type content as complaints, suggestions, queries at the textbox present in the screen(E-2). 4. The use case ends.

Table 3.4 USE CASE SPECIFICATION: COMPLAINT_AND_QUERY

3.3.1.5. USE CASE SPECIFICATION: VIEW

VIEW
1. This use case begins when the educational institution logs on to the Online Coding Platform by entering his/her password. 2. The system verifies that the password entered is correct(E-1). the end user can proceed with the following activity: INTERVIEW, COURSE, PROFILE, TIMER. 3. If the activity selected is INTERVIEW, the S1: view interview sub flow is performed 4. if the activity selected is COURSE, the S2: View courses sub flow is performed 5. if the activity selected is PROFILE, the S3: View profile sub flow is performed 6. if the activity selected is TIMER, the S4: View timer sub flow is performed

Table 3.5 USE CASE SPECIFICATION: VIEW

3.3.2 Exceptional Scenario

3.3.2.1.USE CASE SPECIFICATION: LOGIN

LOGIN
E-1: An invalid username is provided. The end-user can re-enter the valid username or terminate the use case. E-2: An invalid password is provided. The end user can re-enter the valid password or terminate the use case. E-3: System does not allow the End-user to login due to server failure. End-user can login after some time

Table 3.6 USE CASE SPECIFICATION: LOGIN

3.3.2.2. USE CASE SPECIFICATION: EDIT-PROFILE.

EDIT-PROFILE.
E-1: An invalid information is given. The end-user can re-enter the information or terminate the use case. E-2: If the end-user enters an information which is same as another user's information, they have to re-enter his information correctly. E-3: An invalid OTP is provided. The OTP will become invalid after some time and new OTP will be sent 3 times

Table 3.7 USE CASE SPECIFICATION: EDIT-PROFILE.

3.3.2.3. USE CASE SPECIFICATION: MANAGE_PROBLEM_QUESTIONS

MANAGE_PROBLEM_QUESTIONS
E-1: An invalid user password is entered. Then the user can re-enter the password or end the use case. E-2: An invalid question code is found. And hence the user can try again to modify or end the use case. E-3: An invalid description length is detected and hence the user can try again or end the use case. E-4: An invalid Test case count is found and hence the user can try again to modify or end the use case.

Table 3.8 USE CASE SPECIFICATION: MANAGE_PROBLEM_QUESTIONS

3.3.2.4. USE CASE SPECIFICATION: COMPLAINT_AND_QUERY

COMPLAINT_AND_QUERY
E-1: An invalid user password is entered. Then the user can re-enter the password or end the use case. E-2: An invalid complaint, suggestion, queries is entered as it is empty. Then the user has to type the content or end the use case

.

Table 3.9 USE CASE SPECIFICATION: COMPLAINT_AND_QUERY

3.3.2.5. USE CASE SPECIFICATION: VIEW

VIEW
E1: An invalid user password is entered. Then the user can re-enter the password or end the use case. E2: An invalid interview code is found and hence the user can try again to modify or end the use case. E3: An invalid username, the profile is not displayed to the end user. The end -user gets the message of no results on search. Use case begins again

.

Table 3.10 USE CASE SPECIFICATION: VIEW

3.3.3 Extended Scenario

3.3.3.1.USE CASE SPECIFICATION: LOGIN

LOGIN
Sub Flow Forgot password 1. This will be selected by the end-user when the valid password is not known.

Table 3.11 USE CASE SPECIFICATION: LOGIN

3.3.3.2. USE CASE SPECIFICATION: EDIT-PROFILE.

EDIT-PROFILE.
Security: As this contains many personal information, this must be secured properly.

Table 3.12 USE CASE SPECIFICATION: EDIT-PROFILE.

3.3.3.3. USE CASE SPECIFICATION: MANAGE_PROBLEM_QUESTIONS

MANAGE_PROBLEM_QUESTIONS
Sub Flow S-1: Add problem question The question paper details screen is displayed containing question id(for question identification),question code(for domain identification), Question description, timer, test cases. The system then verifies the question code(E-2), question description length(E-3) and Test case count(E-4). And then the Educational institution will make questions visible and link with Server_side. Then the use case begins again. S-2: Edit problem question The system displays the question paper details screen containing all the previously added details by Educational institution(user). Then Educational institutions can edit the question id, question code, question description, timer, test cases, re-upload question description. System checks for any changes updated and verifies the question code(E-2), question description length(E-3) and Test case count(E-4) and links the updated details to the Educational institution. Then the use case begins again. S-3: Delete problem question The system displays the screen which contains all the previous details entered by the Education institution. The Education institution can delete all the registered details including the question id, question code, question description, test cases and the system removes all the records linked to the educational institution. The use case then begins again.

Table 3.13 USE CASE SPECIFICATION: MANAGE_PROBLEM_QUESTIONS

3.3.3.4. USE CASE SPECIFICATION: COMPLAINT_AND_QUERY

COMPLAINT_AND_QUERY
Sub Flow Give feedback Give review about our platform

Table 3.14 USE CASE SPECIFICATION: COMPLAINT_AND_QUERY

3.3.3.5. USE CASE SPECIFICATION: VIEW

VIEW
Sub Flow S1: View interview that have been scheduled by the company Each interview is scheduled and formulated by the company which are made available for the view of end users. Each interview is identified by the unique id which are placed under the concerning domain with the question code (for domain identification). End users are able to view the event description, timings and company details. System displays the list of interviews under the particular filter applied. The system then verifies the interview code(E-2). Then the use case begins again. S2: View courses offered by the website or educational institution The system displays the screen which contains the list of courses offered End-user can view a particular course by filtering it or choosing it from the shown interface by its course id (unique) and course code(for domain identification). System checks for any courses by cross checking course code(E2). The use case begins again. S3: View profile

The end-user can view the list of profiles of other end -users. System verifies the profile username(E-3) and links to the particular profile details and displays them on screen. Then the use case begins again. S4: View timer The system allows the end users mainly the students and the job seekers who participate in competitive coding context/quiz. Timers can be customized to be shown or hidden. The use case ends once the session ends

Table 3.15 USE CASE SPECIFICATION: VIEW

3.4 Class Modelling

Refined Class diagram

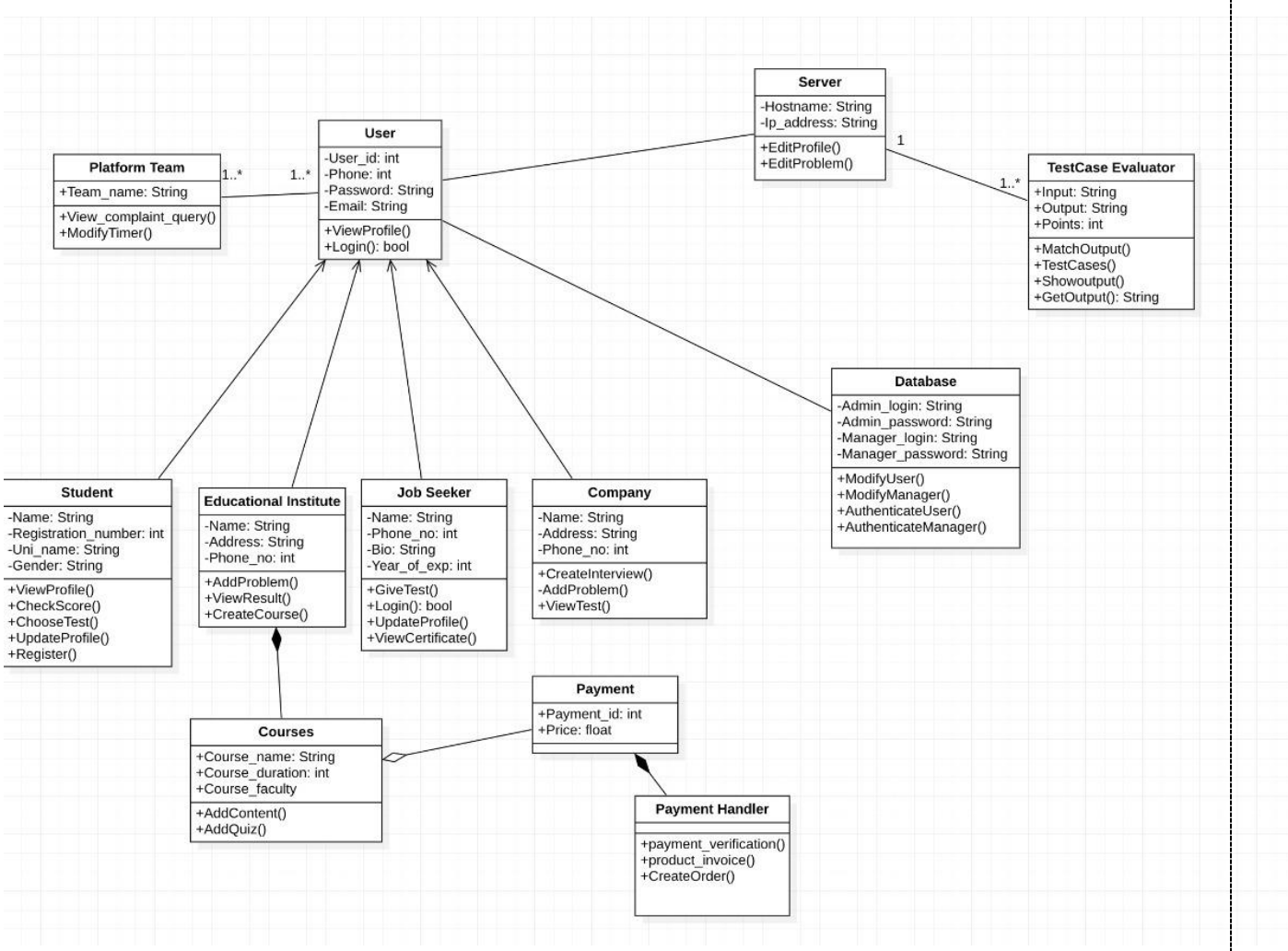


Fig 3.2 Class Modelling

3.5 Dynamic Modelling

3.5.1. State chart diagram:

1. Edit profile

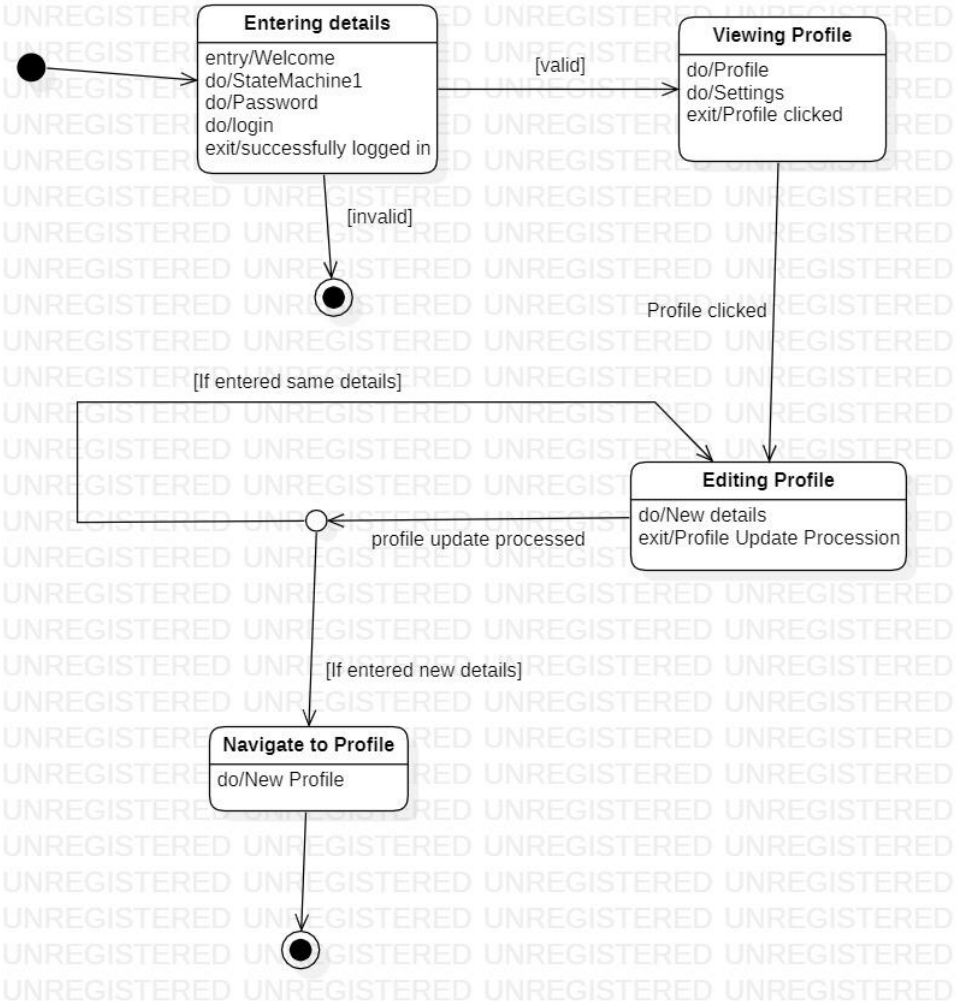


Fig 3.3 Edit profile

2. Feedback and complain

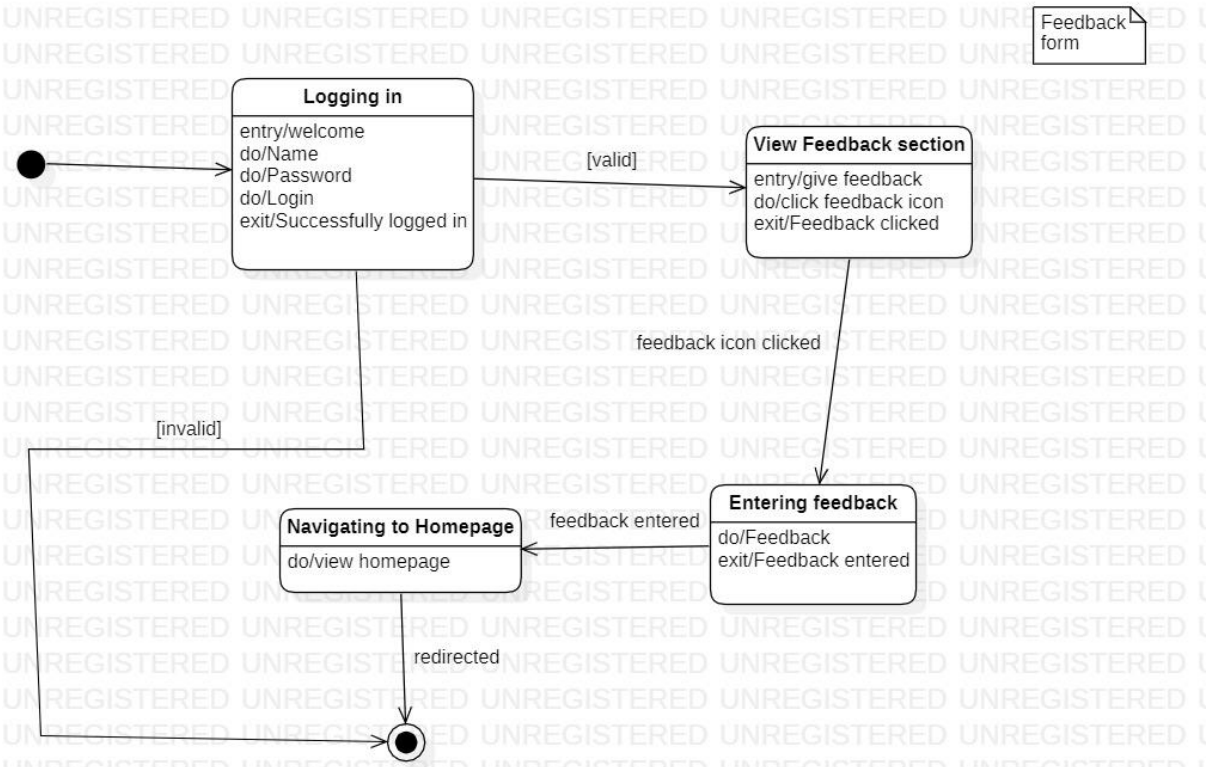


Fig 3.4 Feedback and complain

3. Manage questions

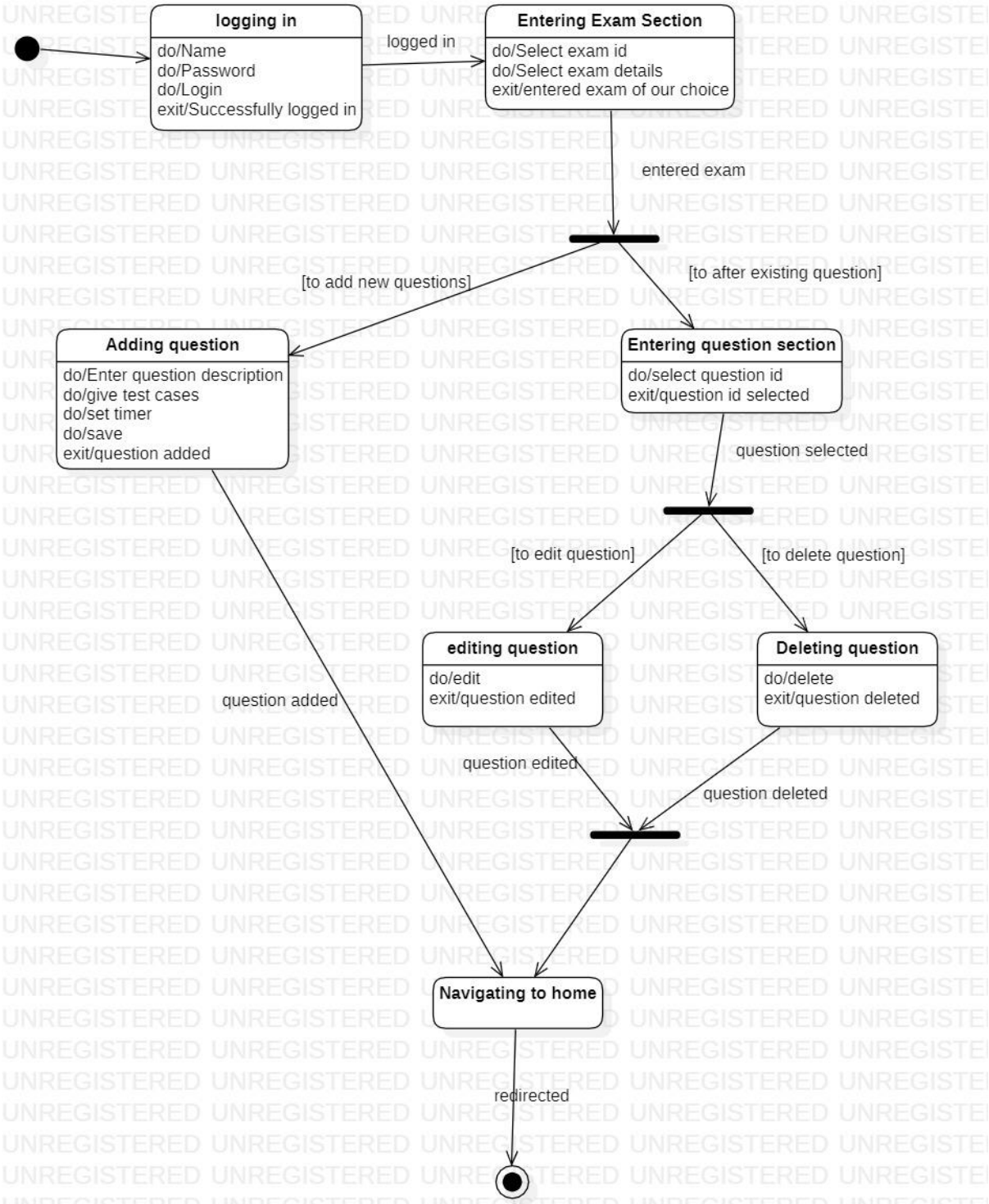


Fig 3.5 Manage questions

4. Register

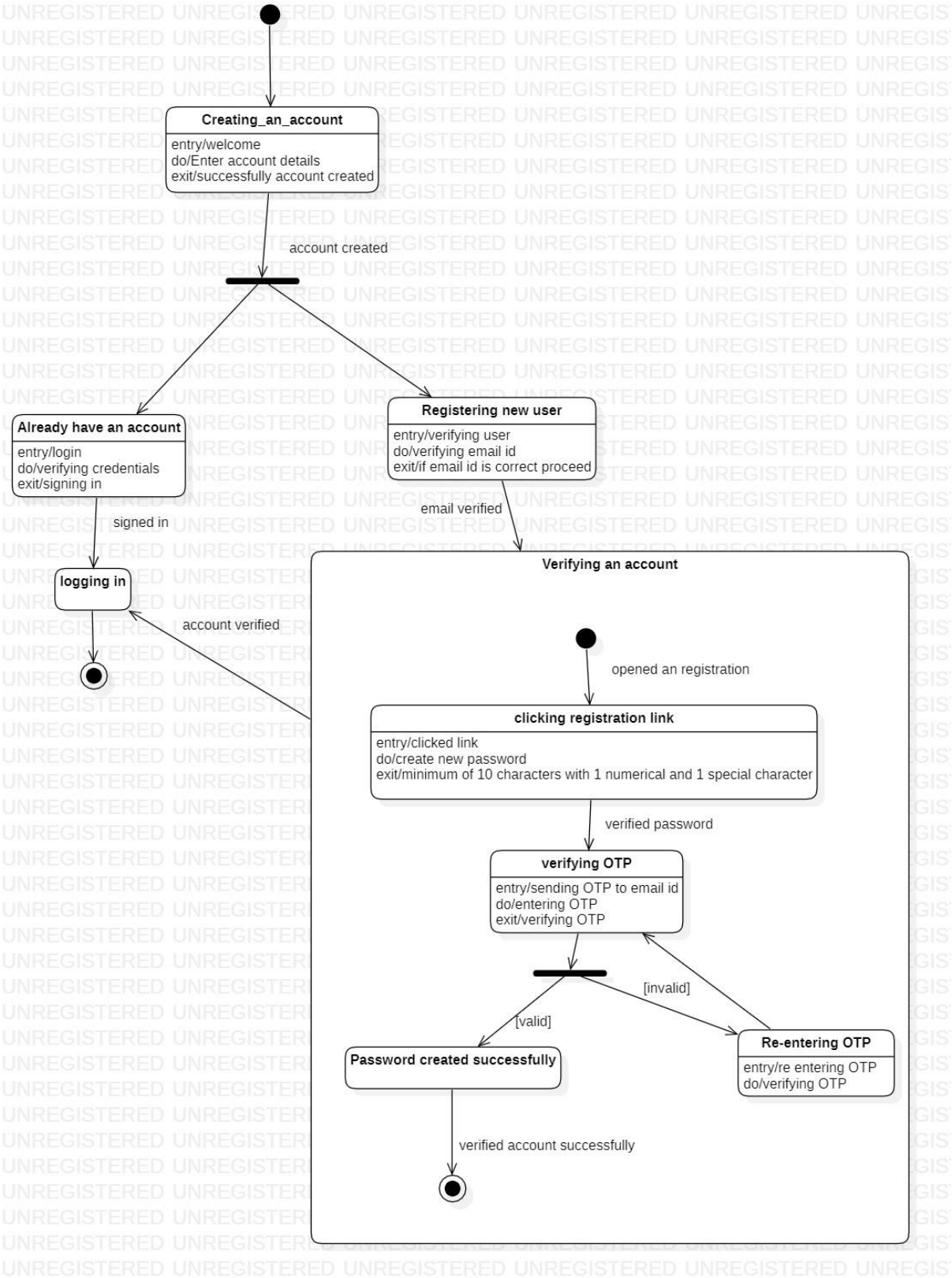


Fig 3.6 Register

5. Enter examination portal

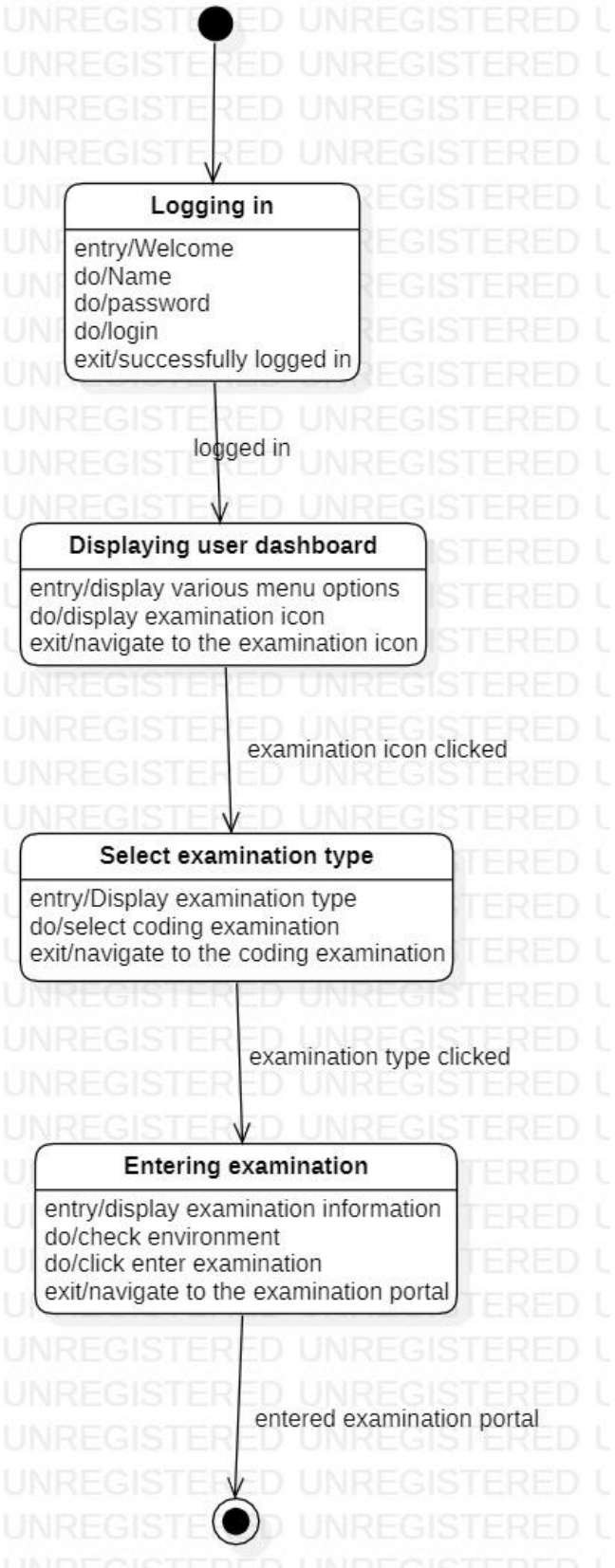


Fig 3.6 Enter examination portal

3.5.2. Activity Diagram

1. Edit profile

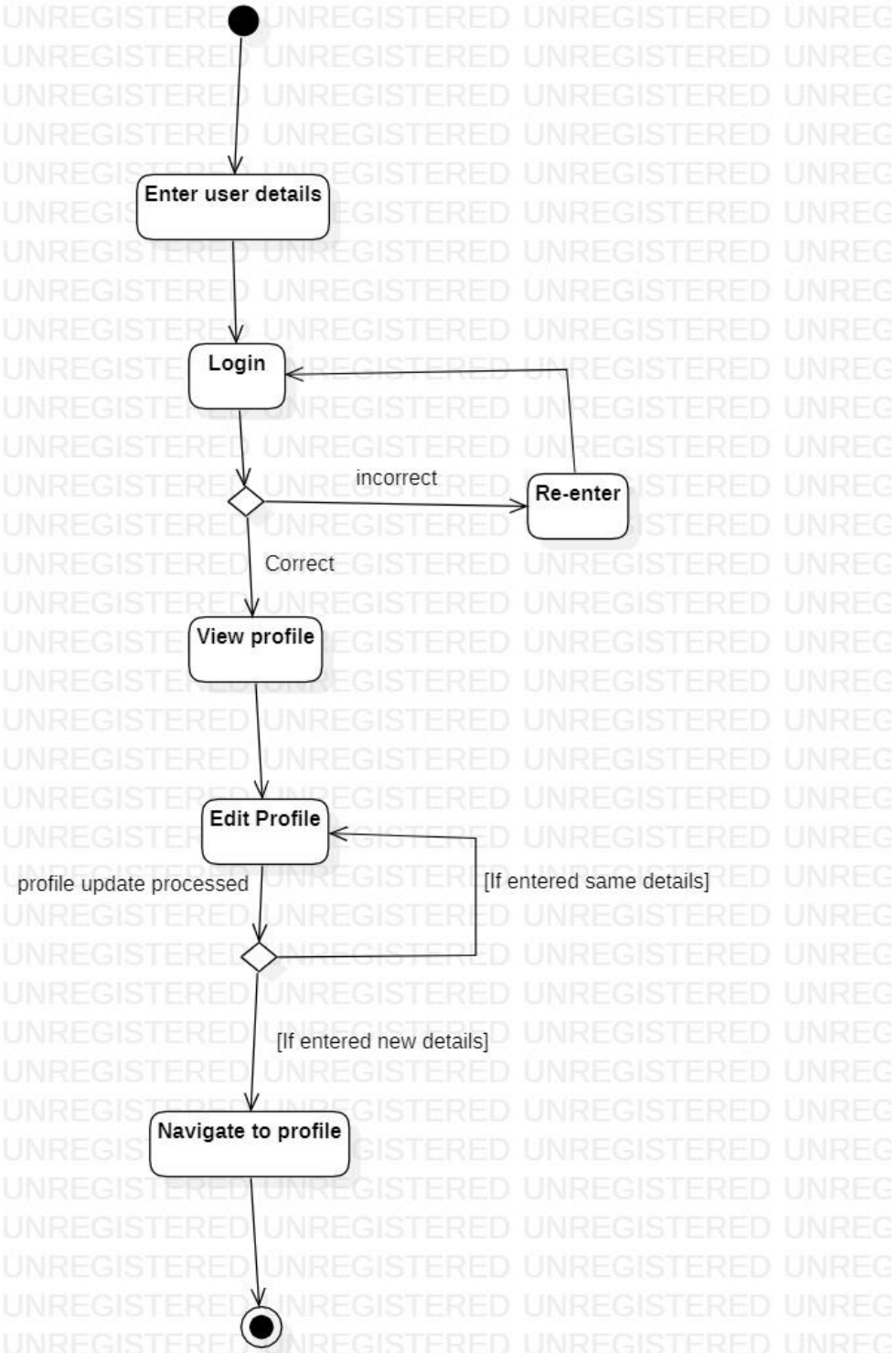


Fig 3.7 Edit profile

2. Feedback and complain

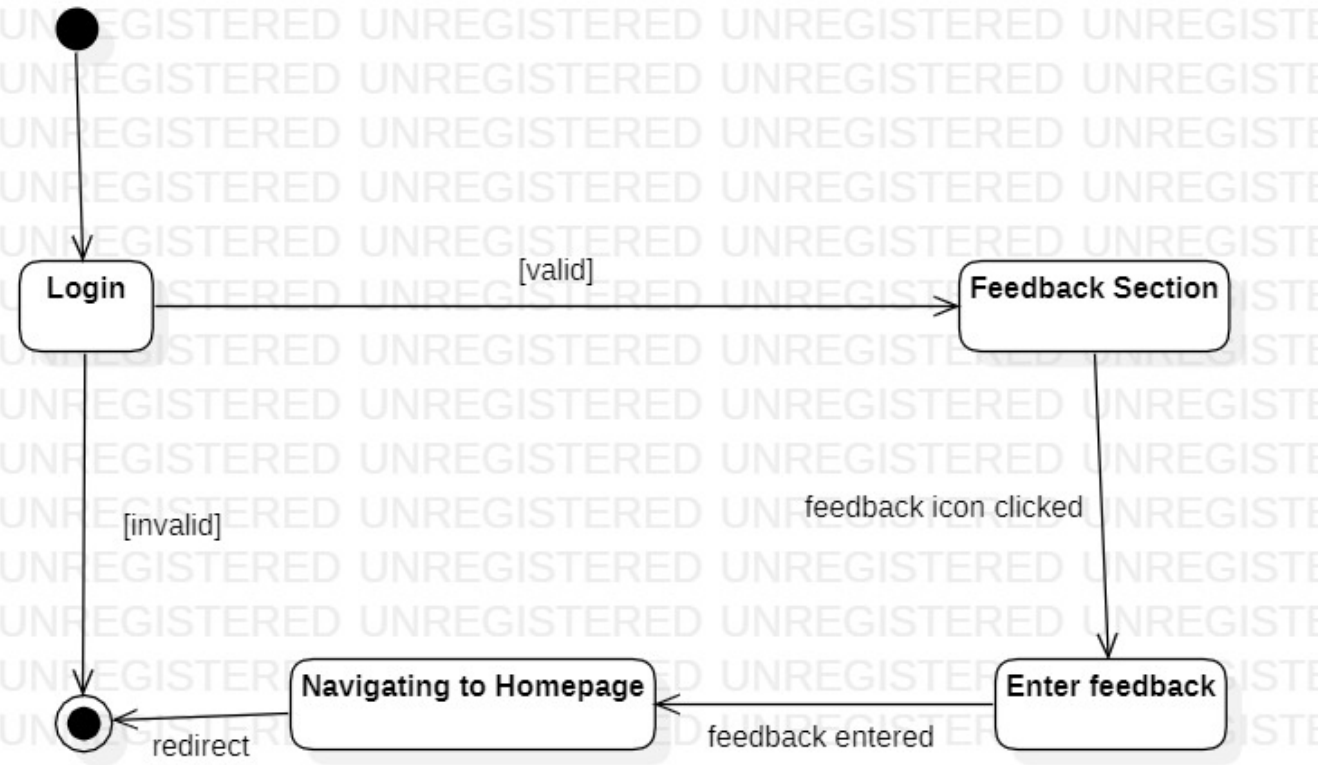


Fig 3.8 Feedback and complain

3. Manage questions

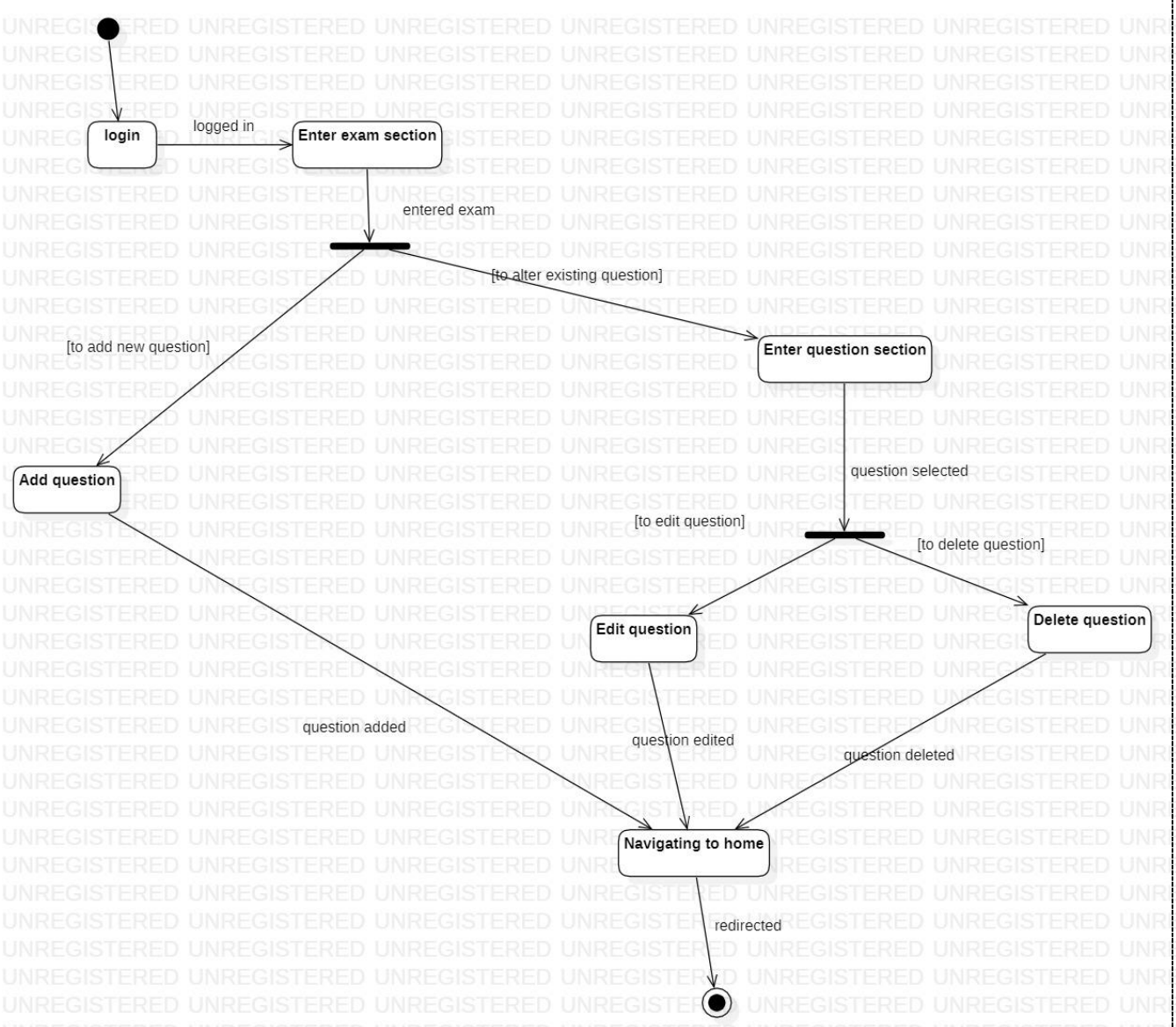


Fig 3.9 Manage questions

4. Register

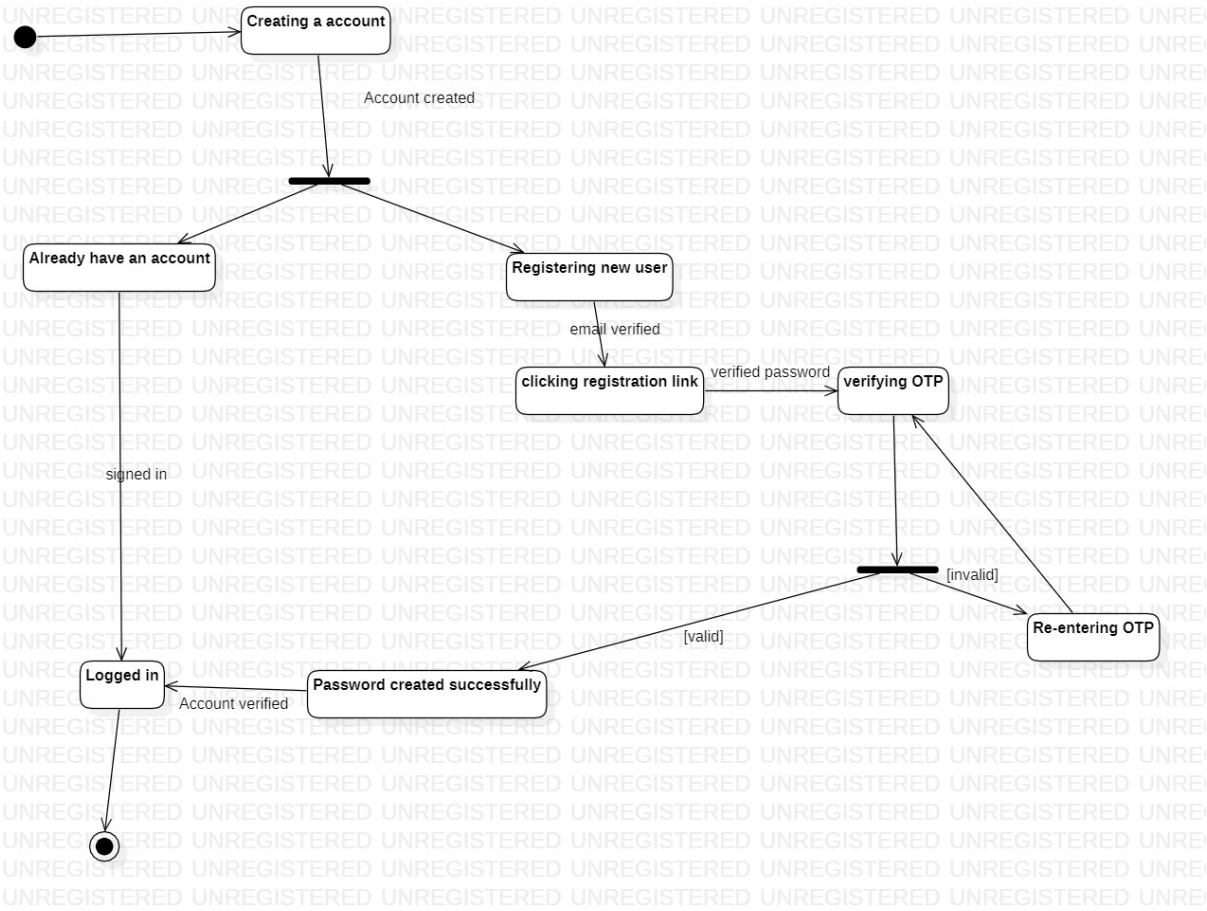


Fig 3.10 Register

5. Enter examination portal

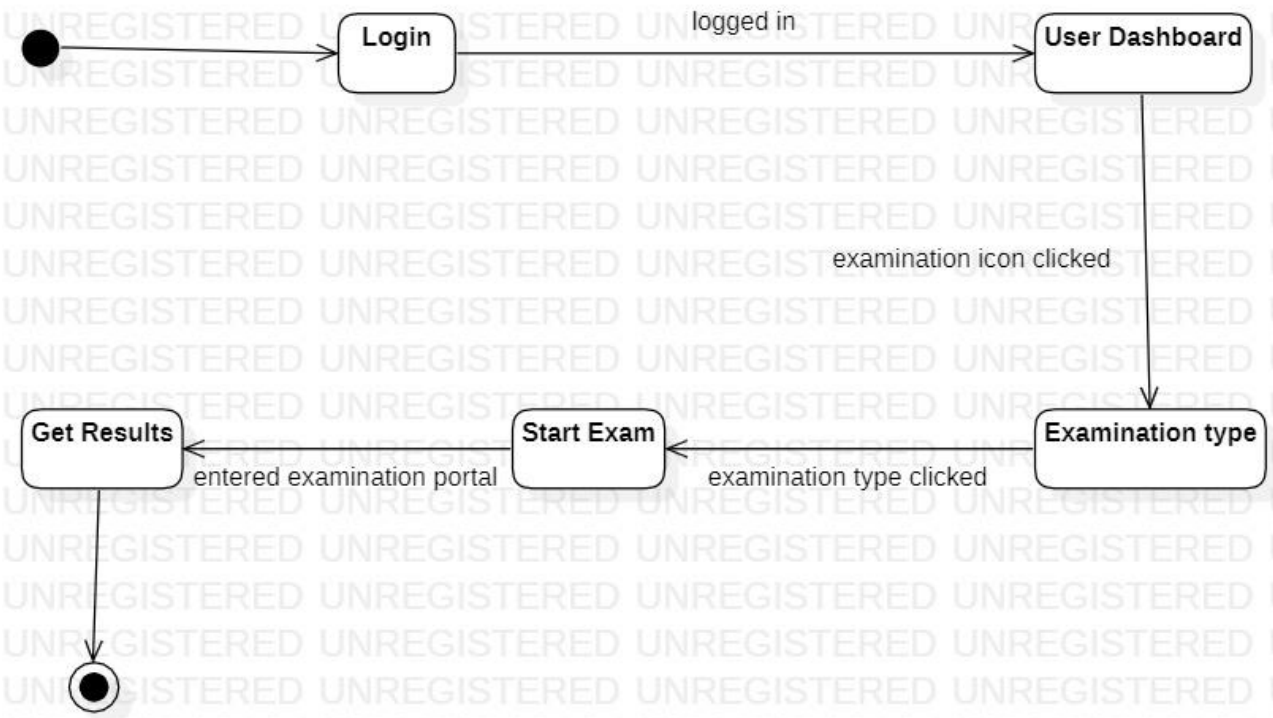


Fig 3.11 Enter examination portal

6. Exam Process

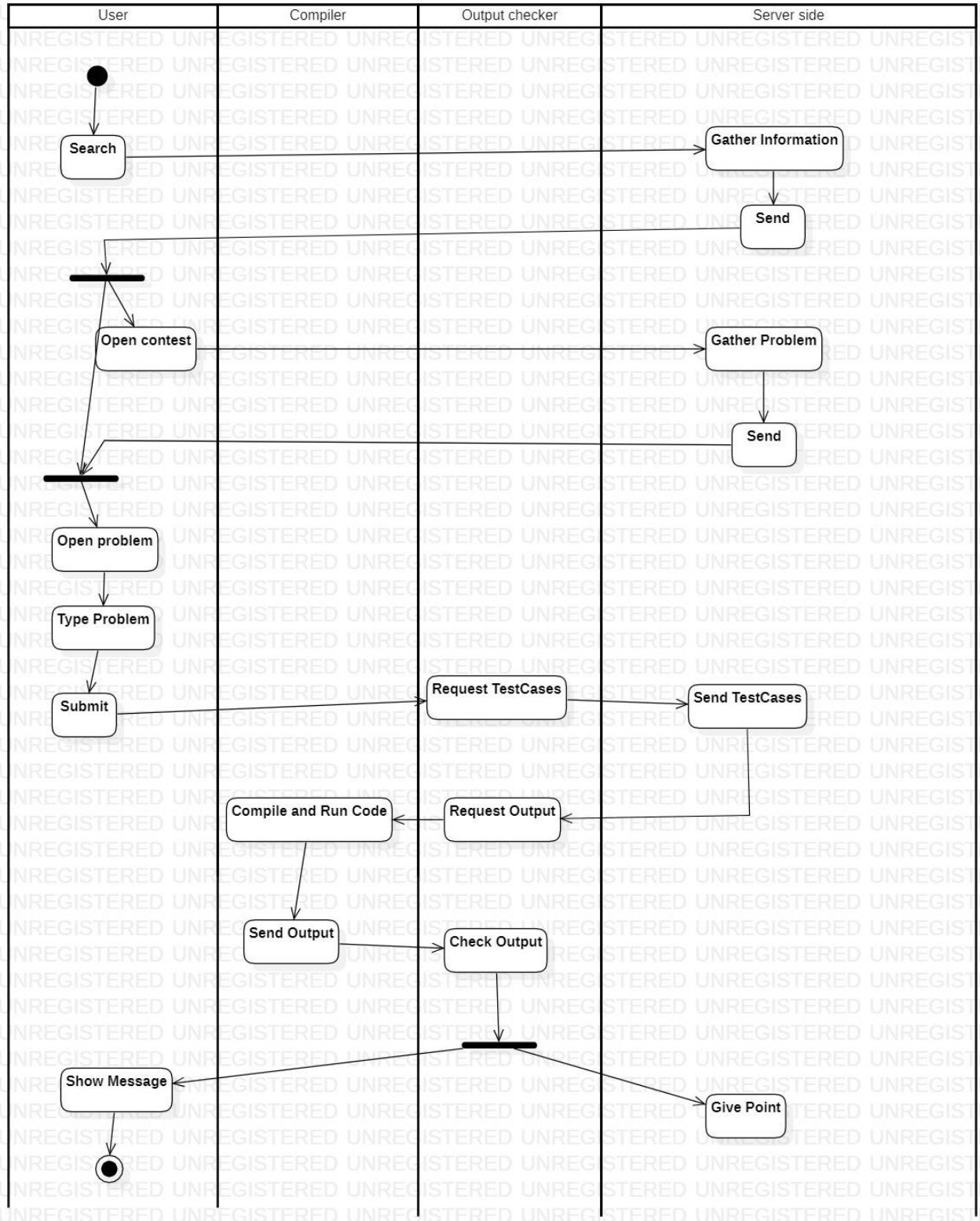


Fig 3.12 Exam Process

3.5.3. Sequence Diagram

1. Login

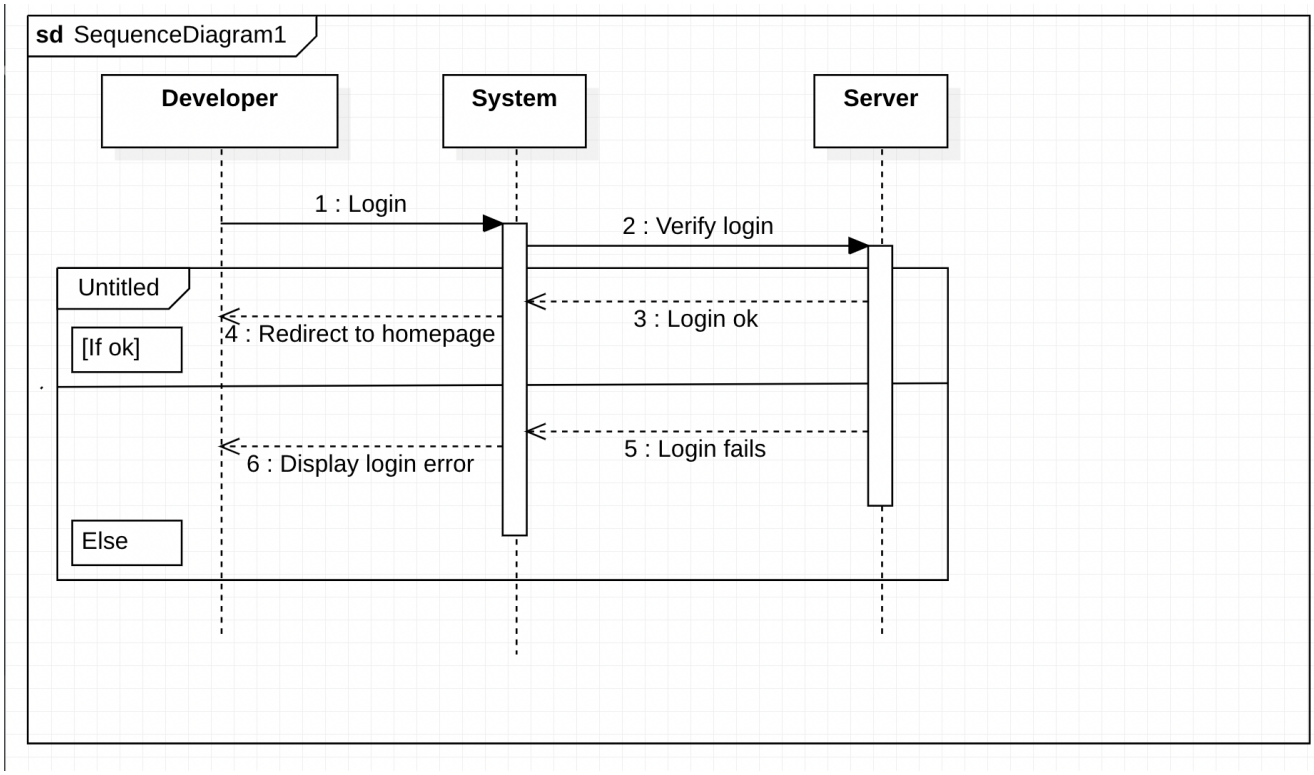


Fig 3.13 Login

2. Participate contest:

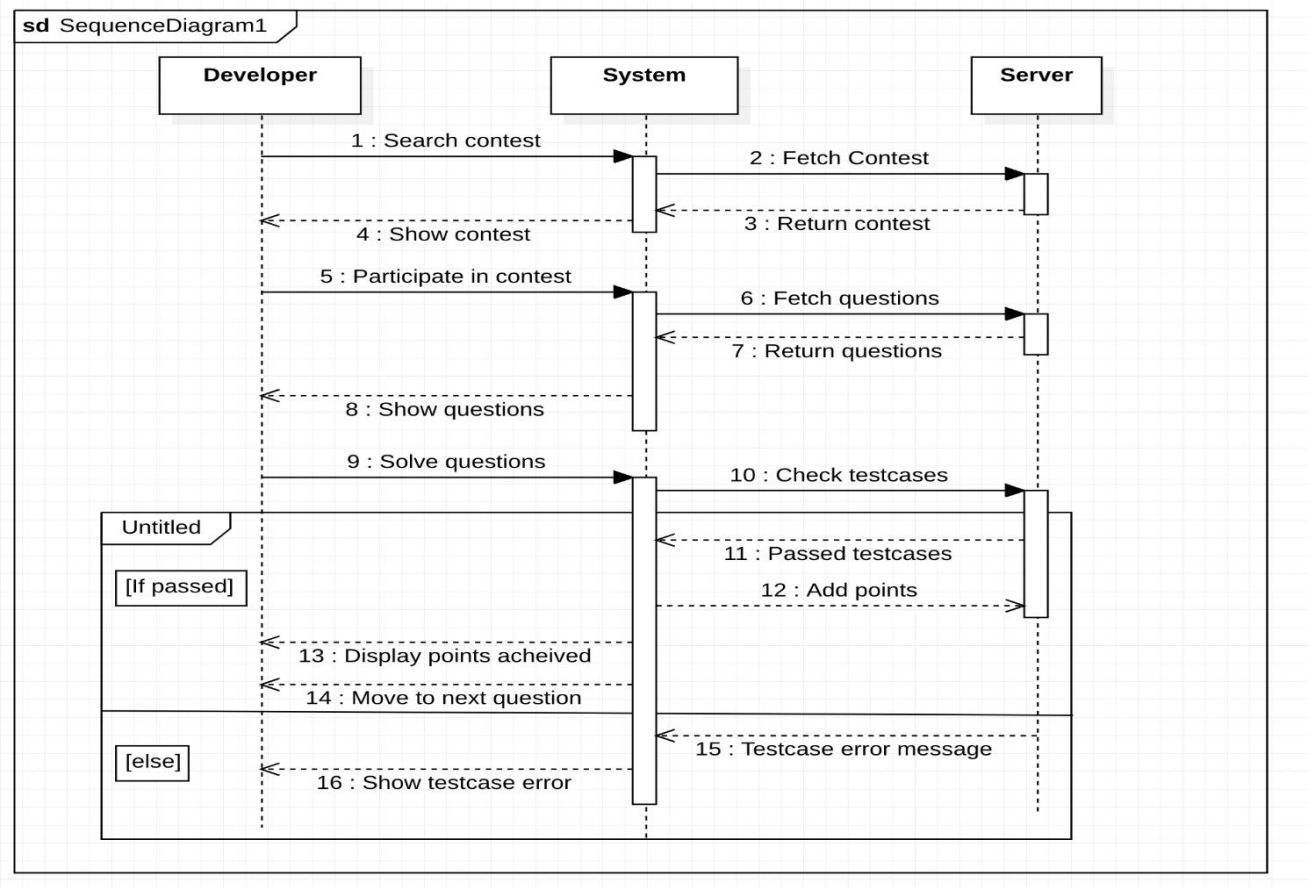


Fig 3.14 Participate contest:

3. Search problem:

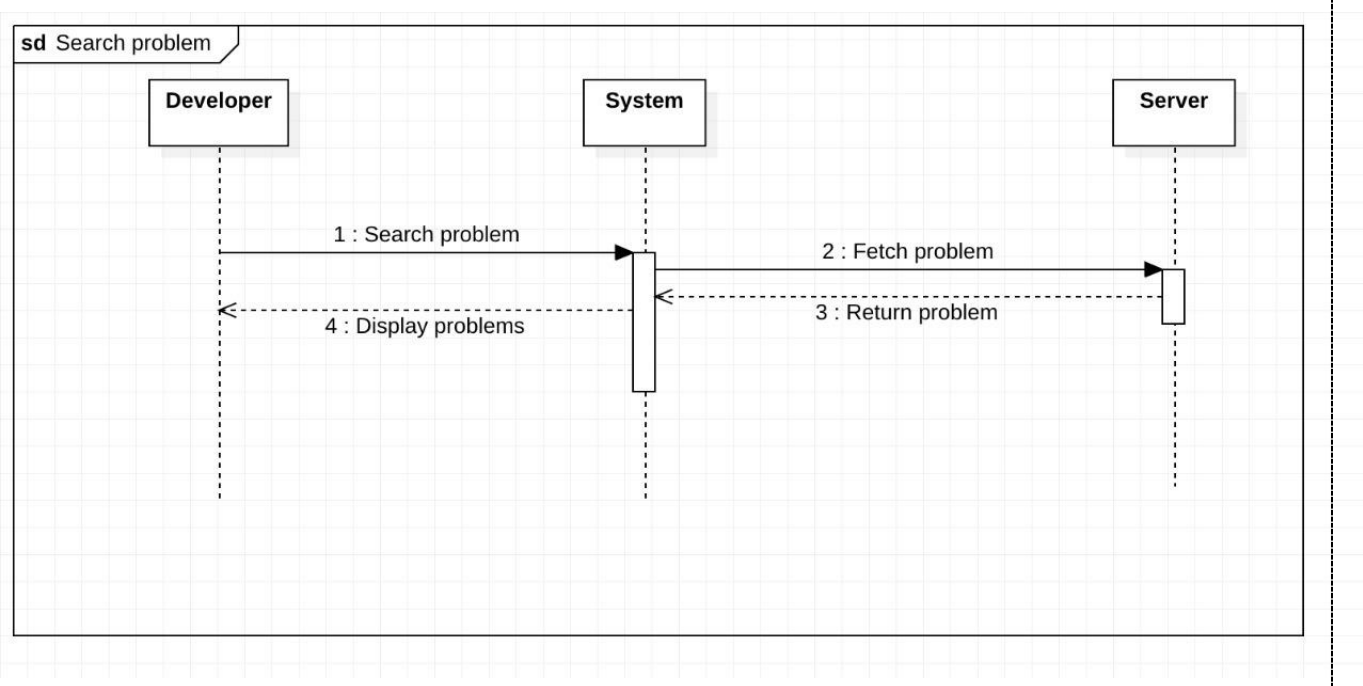


Fig 3.15 Search problem:

4. Solve problem:

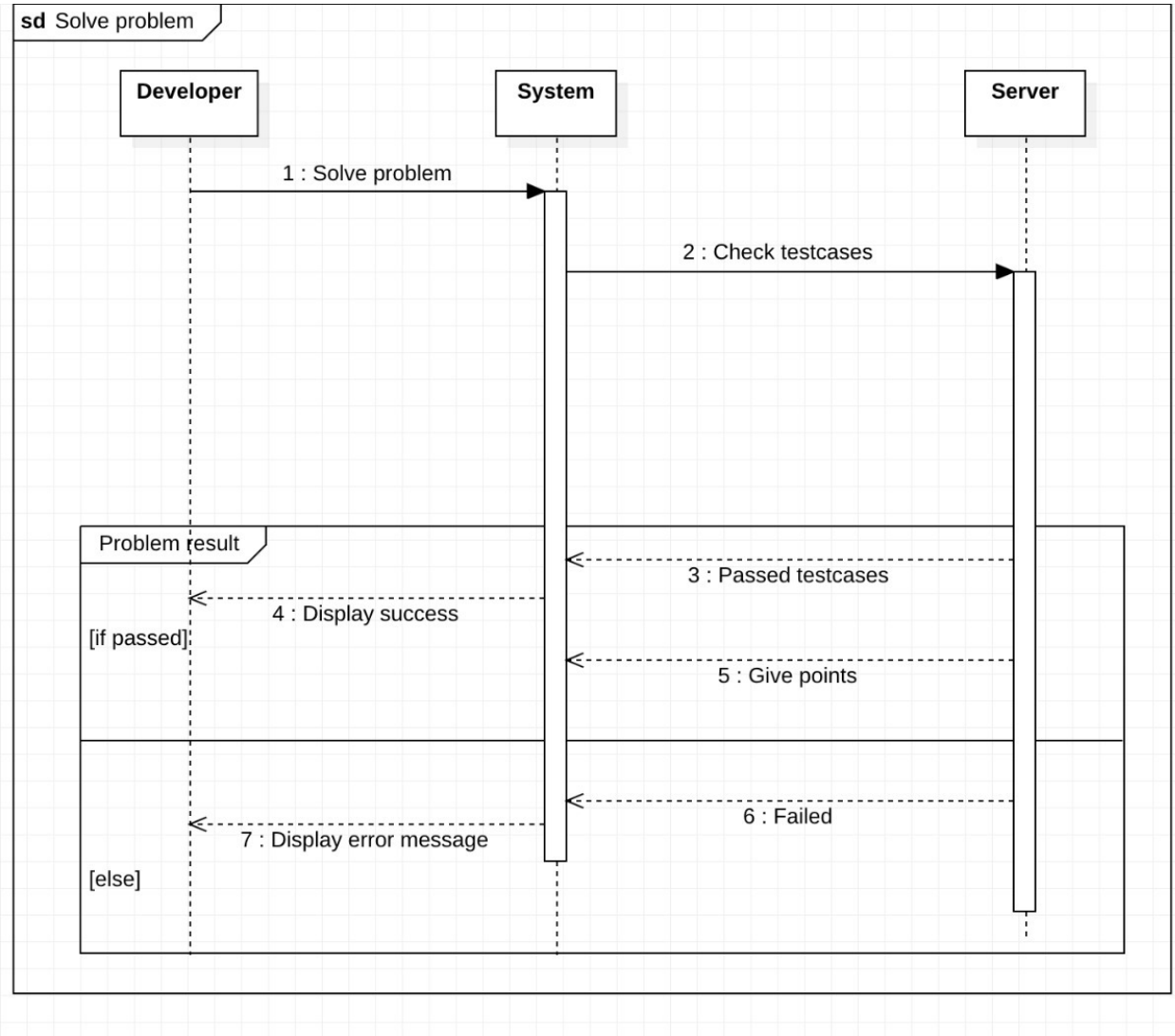


Fig 3.16 Solve problem:

5. View Profile:

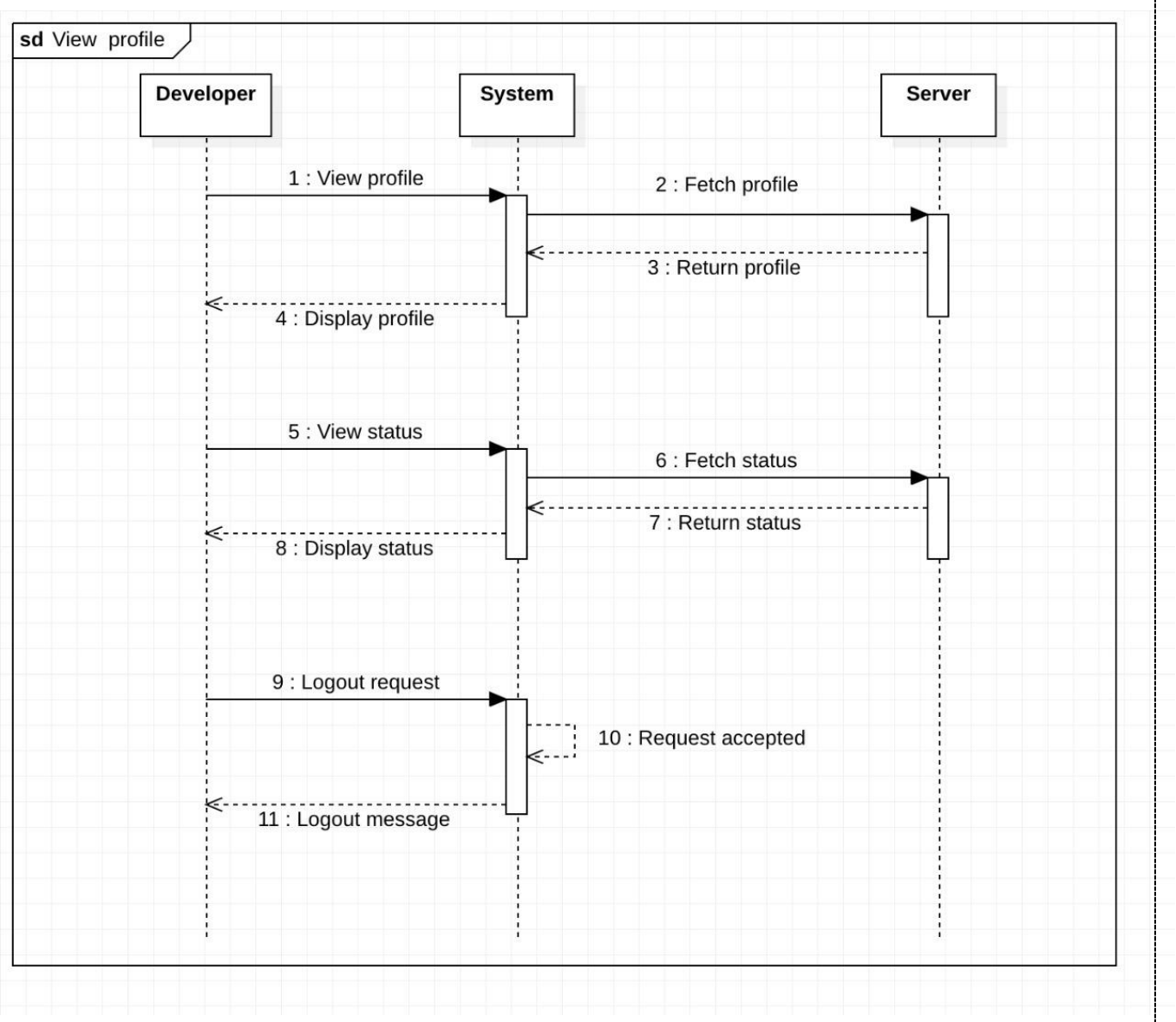


Fig 3.17 View Profile:

3.5.4.Collaboration diagram

1. Login

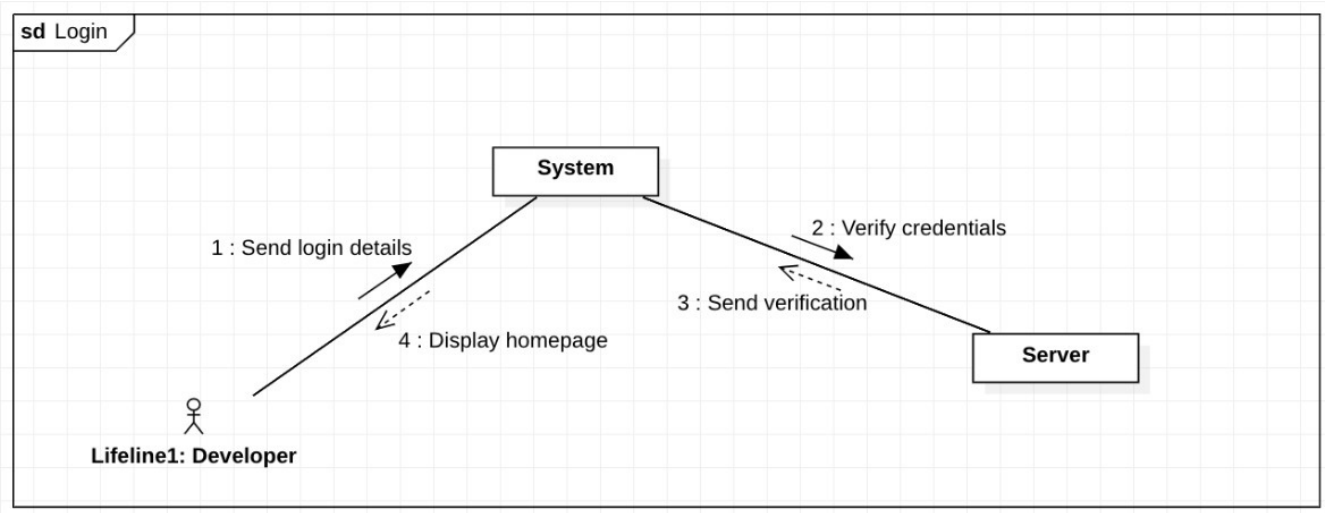


Fig 3.18 Login

2. Participate contest:

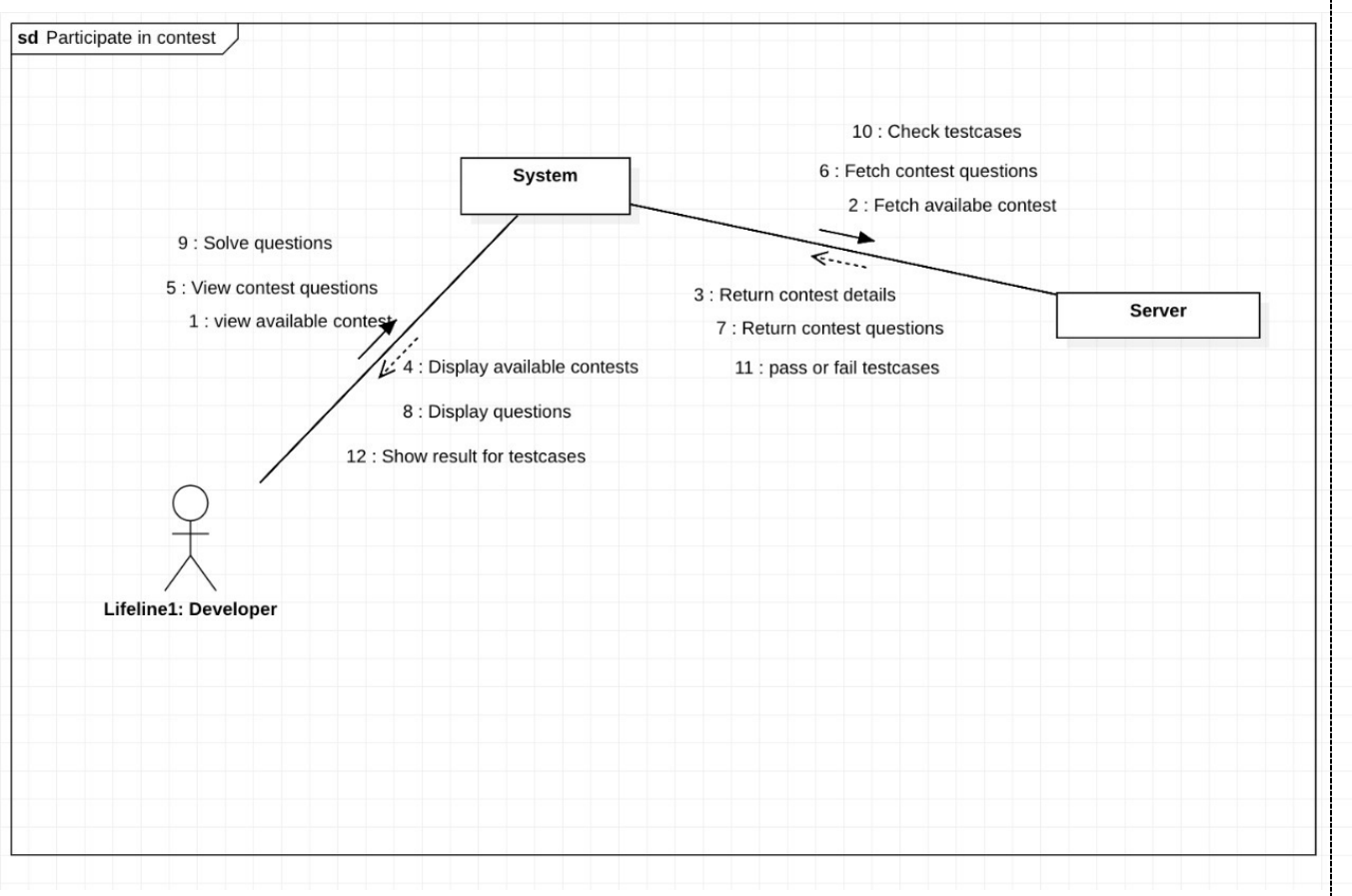


Fig 3.19 Participate contest:

3. Search problem:

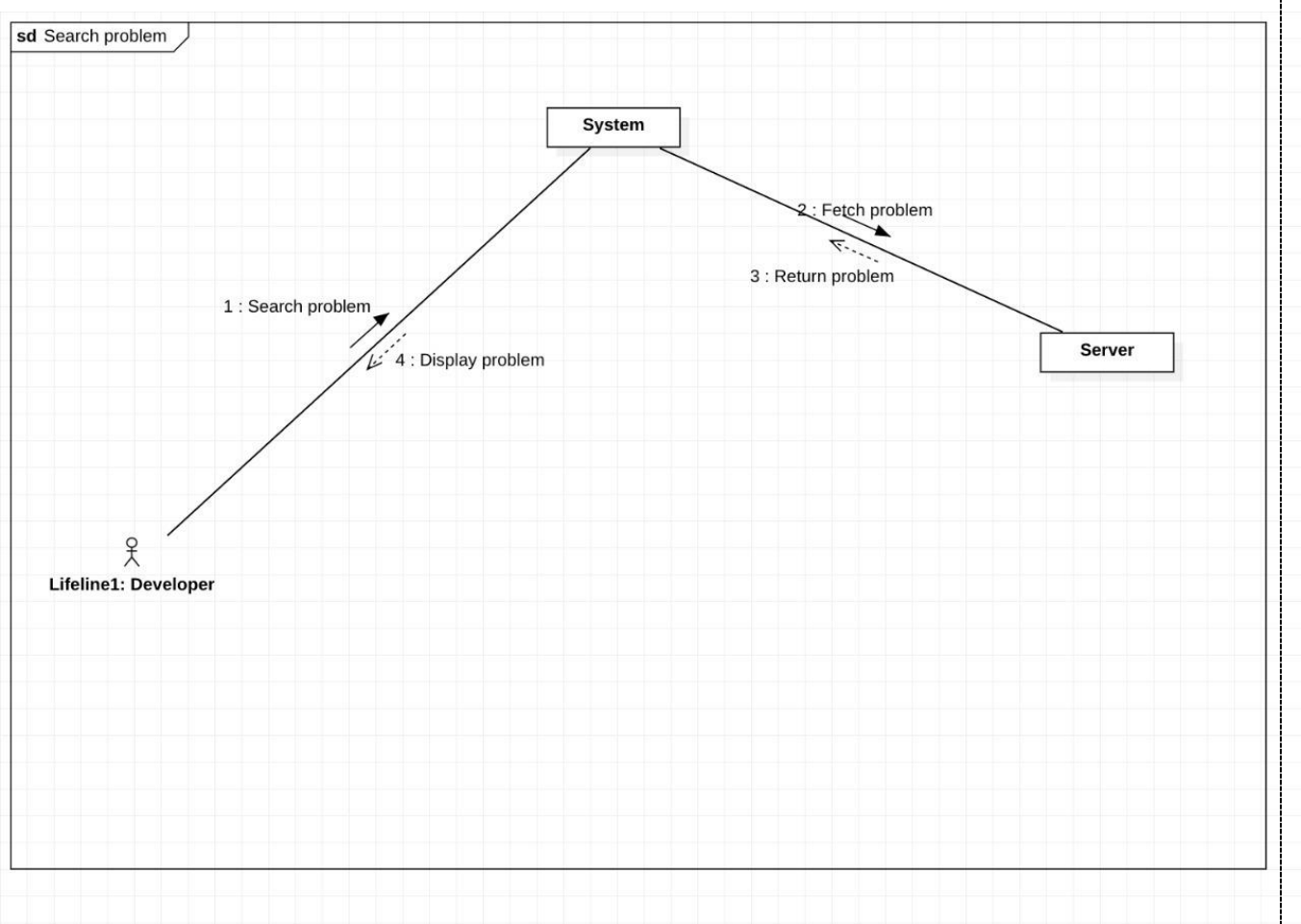


Fig 3.20 Search problem:

4. Solve problem:

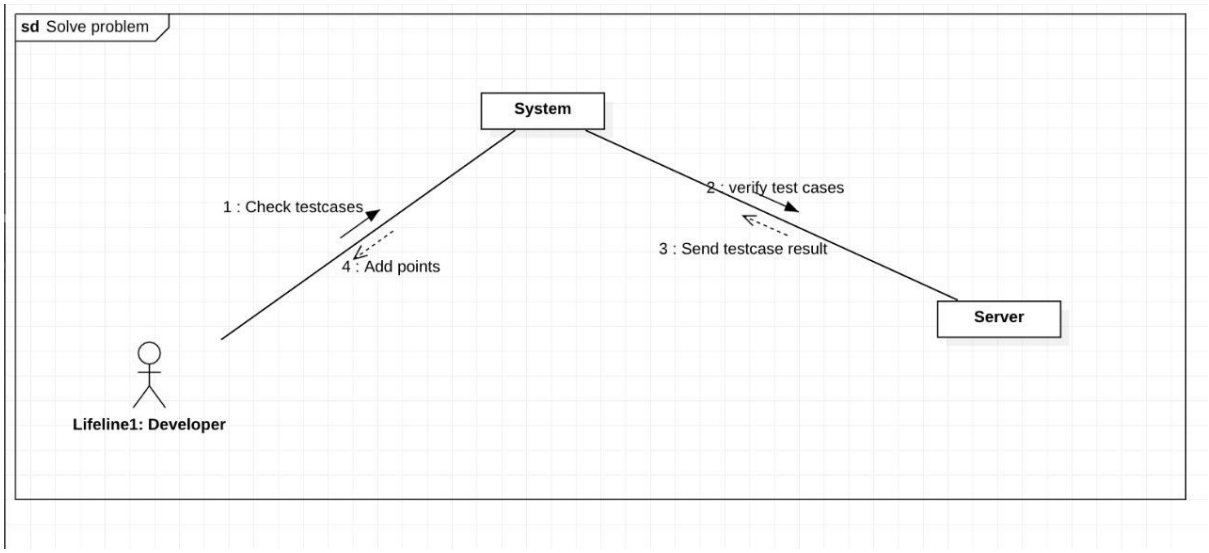


Fig 3.21 Solve problem:

5. View Profile:

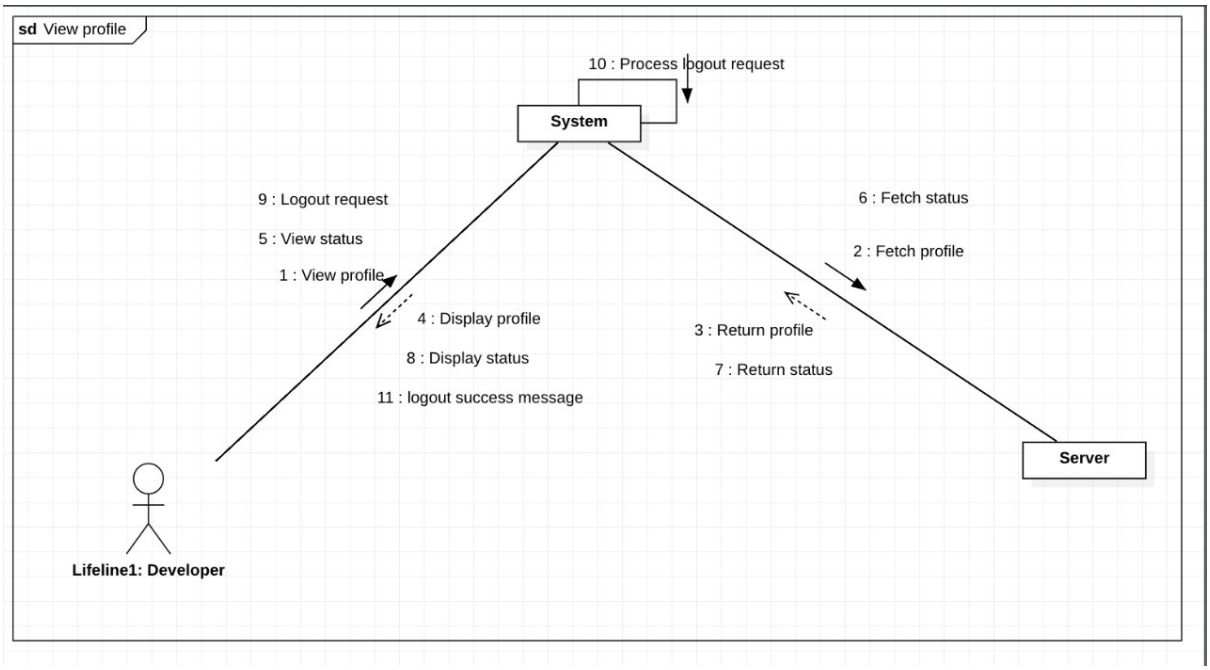


Fig 3.22 View Profile:

4. OBJECT ORIENTED DESIGN WORKFLOW

4.1 Why need the Design Workflow?

Object-oriented design (OOD) is the process of using an object-oriented methodology to design a computing system or application. This technique enables the implementation of a software solution based on the concepts of objects.

In object-oriented system design and development, OOD helps in designing the system architecture or layout - usually after completion of an object-oriented analysis (OOA). The designed system is later created or programmed using object-oriented based techniques and/or an object-oriented programming language (OOPL).

The OOD process takes the conceptual systems model, use cases, system relational model, user interface (UI) and other analysis data as input from the OOA phase. This is used in OOD to identify, define and design systems classes and objects, as well as their relationship, interface and implementation.

OOD serves as part of the object-oriented programming (OOP) process or lifecycle.

4.2 Formats of the Attributes

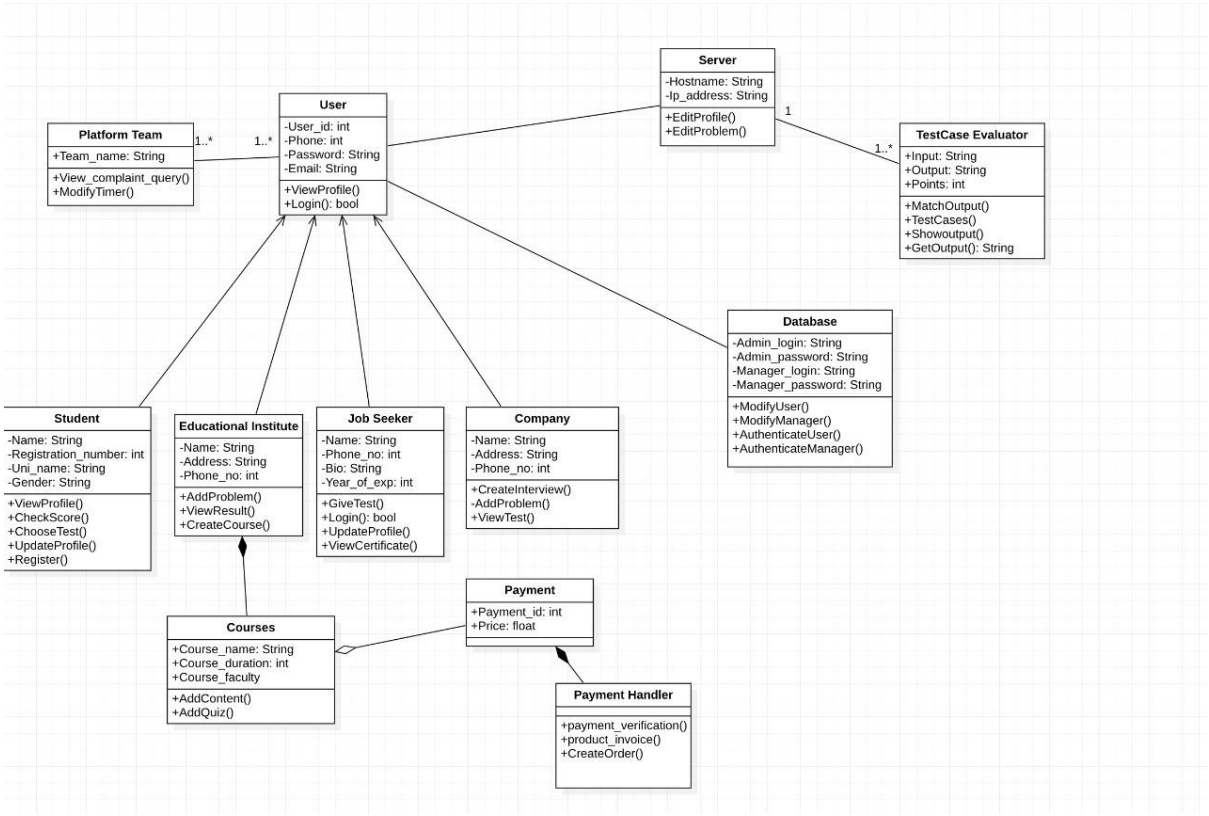


Fig 4.1 Formats of the Attributes

4.3 Allocation of Operations to Classes

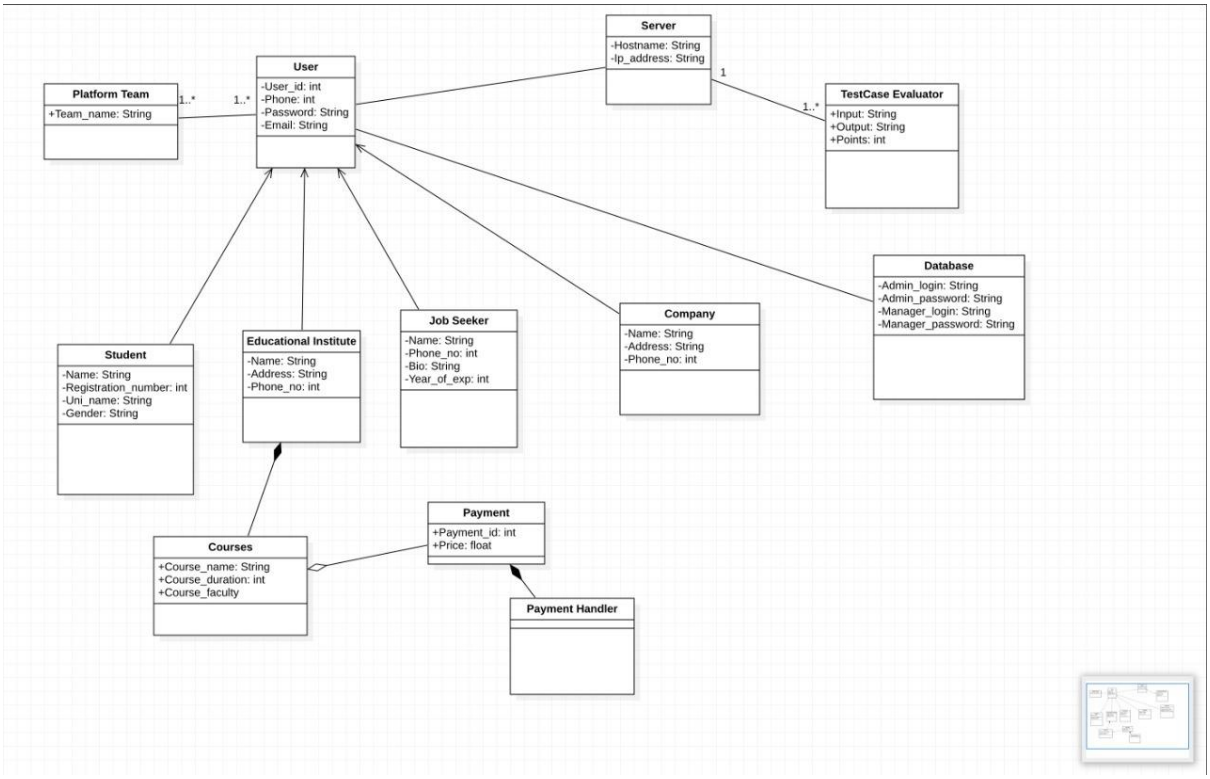


Fig 4.2 Allocation of Operations to Classes

4.4 Allocation of Operations

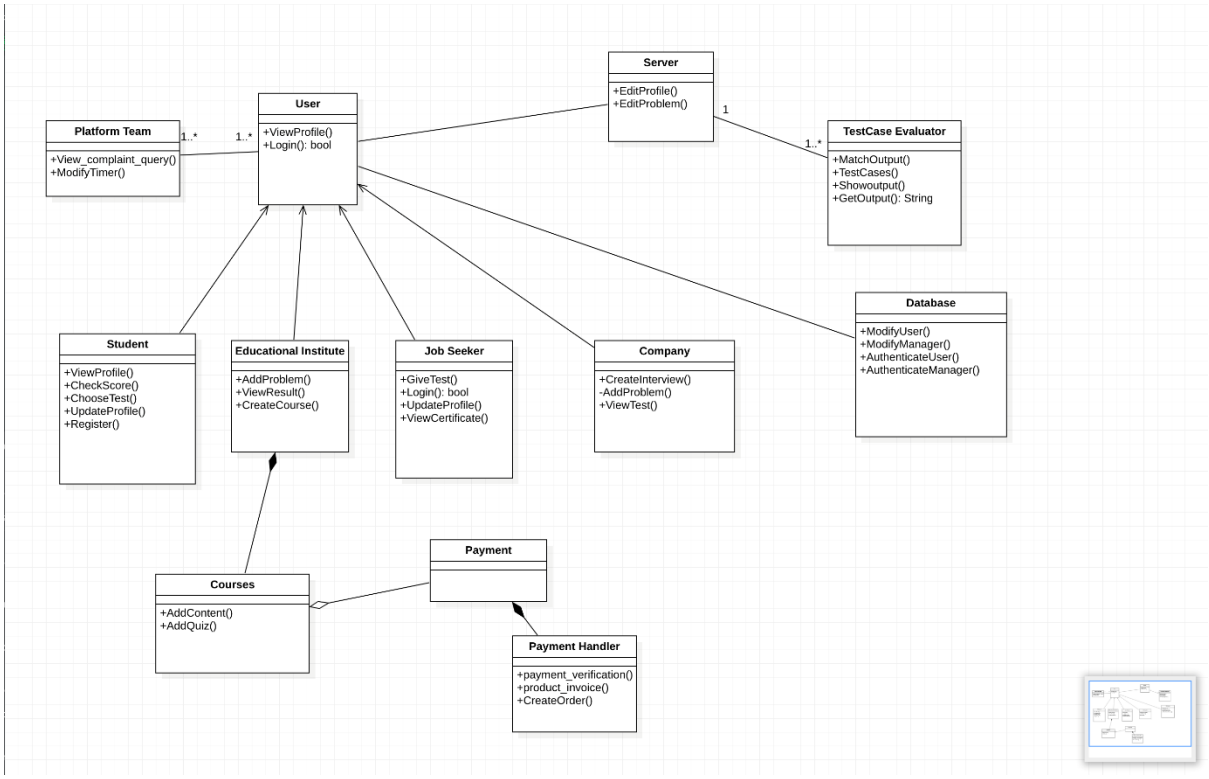


Fig 4.3 Allocation of Operations

5. OBJECT ORIENTED IMPLEMENTATION WORKFLOW

5.1 Components Diagram

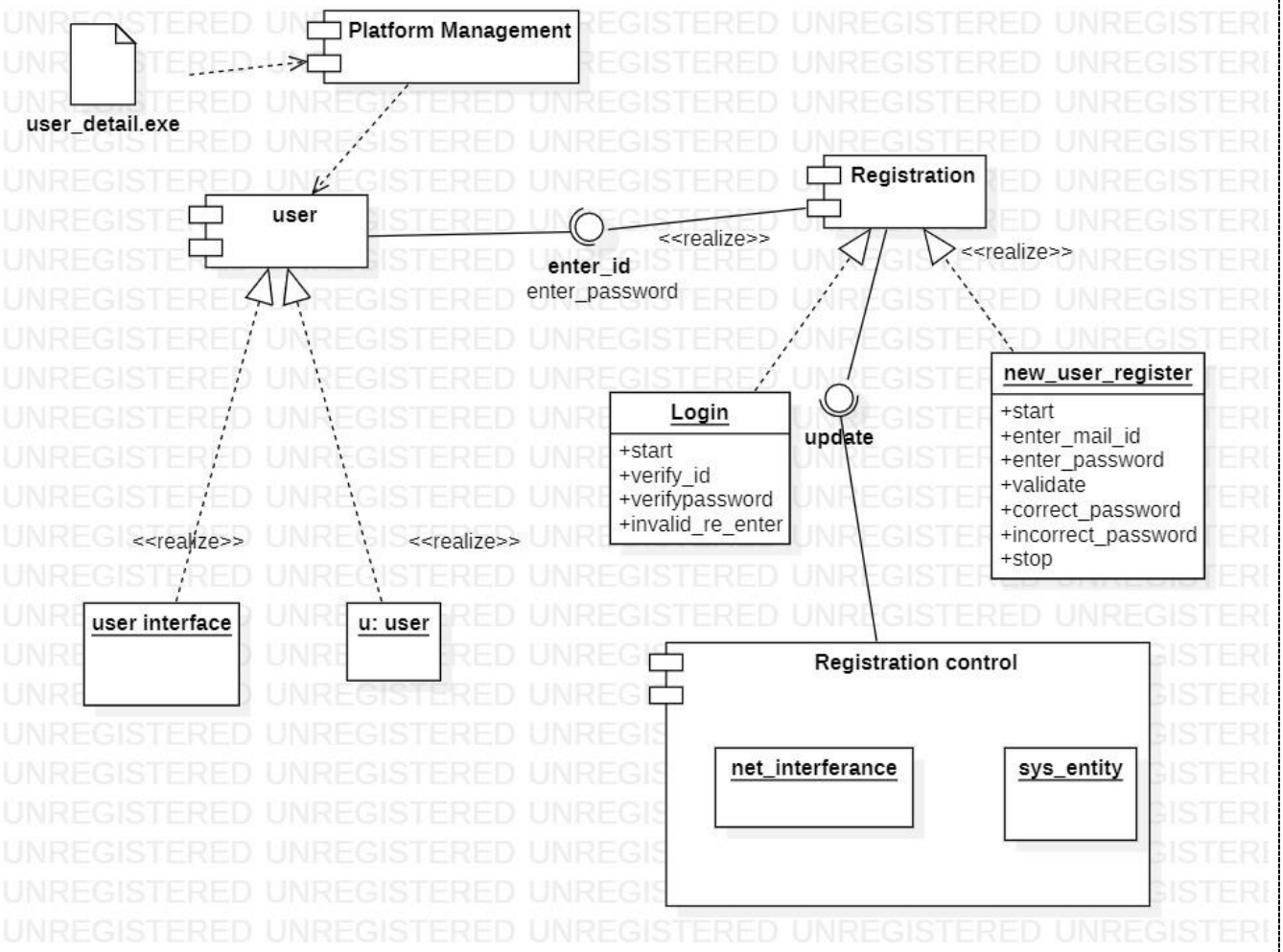


Fig 5.1 Components Diagram

5.2 Deployment Diagram

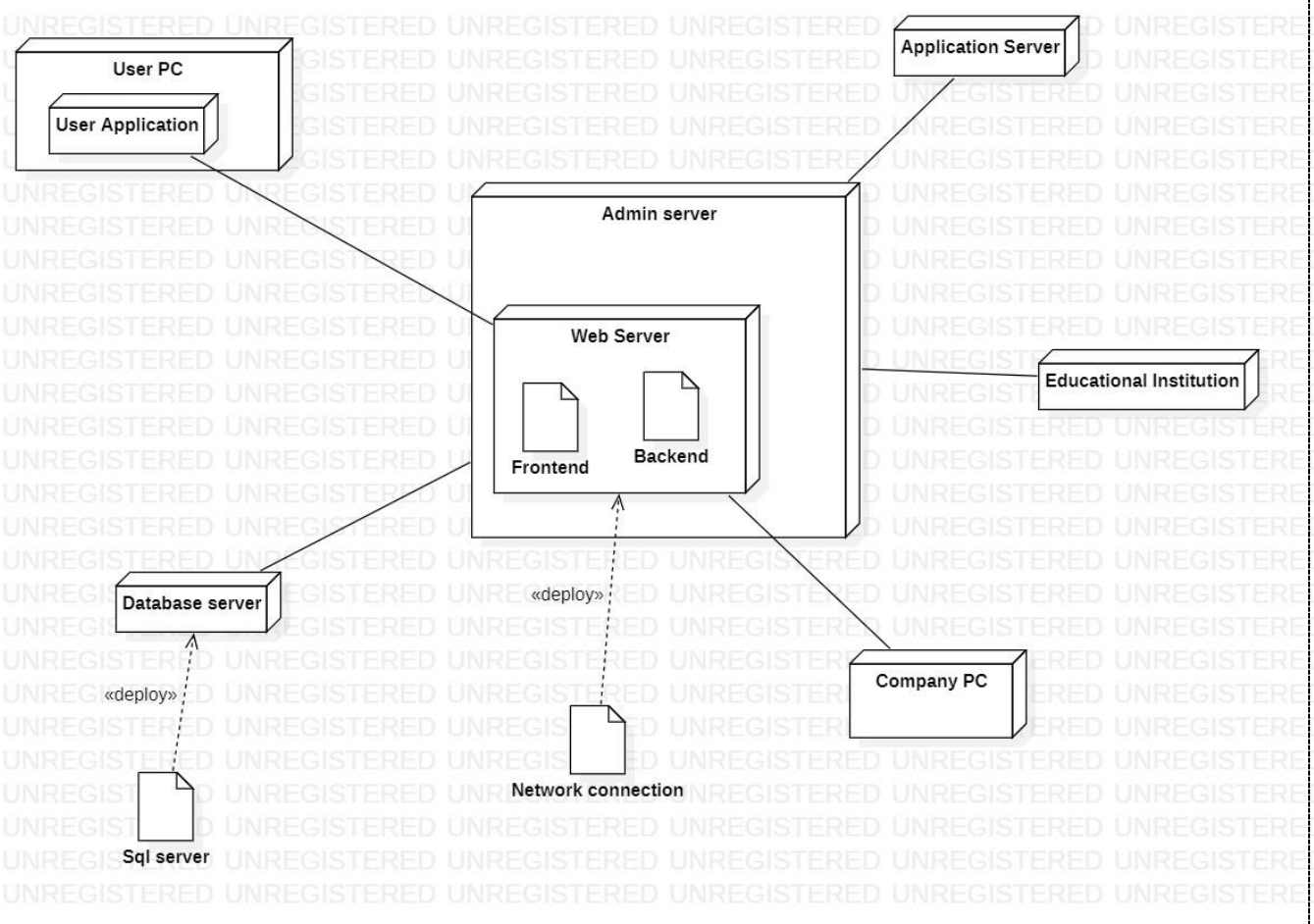


Fig 5.2 Deployment Diagram

6. Conclusion

A Coding Platform for students may sound as simple as any online IDE's having the same functionalities. But this project, upon implementation and regular maintenance, would have a significant and drastic change towards their career preparation .This would rapidly increase the number of placements happening and would enhance the pool of skilled resources in the college. A panel of staffs could be given the responsibility of managing this project and the project can be extended to be developed as a full scale product. Hence, technical placement training by outside vendors can be completely eliminated in a step by step manner. This would probably urge all the students as well as the staffs to set their skill range as high as possible. This will help the students take a better step in their careers. Student can eliminate the fear to code as they would start cultivating the habit of coding right from the beginning. Integrating the features in one single platform brings in an uniformity in which students learn rather than using multiple IDEs to code and learn.

Future research and commercial development needs to better emphasize personalized support and precise, contextualized feedback and explore ways of explaining to learners why and when to use particular coding concepts. Teachers must be very selective in their use of materials, focusing on the more evidence-based tutorials, particularly the educational games. All educational games in the list provide hierarchical structure, immediate feedback, and opportunities that learners actively write code and use subsequent knowledge for coding throughout the tutorial.

7. REFERENCES

[1]. M. Ilahi-Amri, L. Cheniti-Belcadhi, and R. Braham, “A framework for competence based e-assessment.” Interaction Design and Architecture(s) Journal (IxD&A), vol. 32, pp. 189–204, 2017.

[2]. Alexey Pelykh (2020), ‘Code Architecture for Web Apps: How to Choose’

[3]. S. Combe´fis and J. Wautelet, “Programming trainings and informatics teaching through online contests.” Olympiads in Informatics, vol. 21, 2014.Schach and Stephen R. (2003), "An Introduction to Object-Oriented Systems Analysis and Design with UML and the Unified Process", Tata McGraw Hill

[4]. Athanasios Drigas (2018), “Online Learning Facilities to Support Coding and Robotics Courses for Youth”

[5]. Grady Booch, “Object Oriented Analysis and Design with Application”, Addison Wesley

[6]. Ali Bahrami (2008), “Object Oriented Systems Development”, McGraw-Hill, 1999

[7]. I. Jacobson et al (2000) , “The Unified Software Development Process”, Addison Wesley

[8]. Zhu, J., Warner, J., Gordon, M., White, J., Zanelatto, R., & Guo, P. J. (2015, October). Toward a domain-specific visual discussion forum for learning computer programming: An empirical study of a popular MOOC forum. In Visual Languages and Human-Centric Computing (VL/HCC), 2015 IEEE Symposium on (pp. 101-109). IEEE.

[9]. Bishop, J., Horspool, R.N., Xie, T., Tillmann, N. and de Halleux, J. 2015. Code Hunt: Experience with coding contests at scale. 398–407.