COMSATS University Islamabad,
Park Road, Chak Shahzad, Islamabad Pakistan

SOFTWARE DESIGN DESCRIPTION

(SDS DOCUMENT)

for

HATH MILAO

Version 1.1

By

Qaiser Ali Khan	**CIIT/FA16-BSE-127/ISB**
Muhammad Haroon Javed	**CIIT/FA16 -BSE-110/ISB**

Supervisor

Mr. Qasim Malik

Bachelor of Science in Software Engineering (2016-2020)

Table of Contents

Revision History

Name	Date	Reason for changes	Version

Application Evaluation History

Comments (by committee) ***include the ones given at scope time both in doc and presentation**	**Action Taken**
Activity Diagram Need Revision	We revised the activity diagrams and updated with new changes. Activity diagrams that we changed are Add Job and Internship, Add Challenge, Add Education, Add Event, Add Experience, and Add Final Year Project.
Class Diagram Need Revision	We revised class diagram and implemented new changes. The changes we made in class diagram is to move trackjobless function in Job Class.

Supervised by

Mr. Qasim Malik

Signature______________

1. Introduction

Students and alumni register or login into the system. Once students and alumni login they will see the dashboard page in which different section will show like job, companies, etc. They will be able to view, save and apply for jobs and internships. Students and alumni will be able to view and create a CV. Students and alumni have a message section in which they will contact with companies and career centers. They have a powerful search filter for the jobs and internships. Students and alumni will be able to view update his/her profile in which their skills and other information are shown, etc. Students and alumni will see and save companies and on-campus events. Students and alumni will be able to attempt challenges which will be posted by company. The score of attempt challenge will be shown in challenge attempt section. Students will be able to apply for different projects which are offered by industry. Companies will register or login they will see the dashboard page in which different section will show like job, students, etc. Companies will be able to post jobs and internships which will show on the job section of thousands of students and alumni. Companies will be able to see the jobs and internships applicant profile. Companies will be able to create a different type of events. Companies will be able to manage all jobs fair on this platform. Companies will be able to contact directly with students and alumni. Career center of university register or login into the system. After that, on their track jobless section, they will see all the student of their own university those are jobless and looking for jobs. Career Center will be able to see the profile of their university students and alumni. The career center will be able to contact directly with students. The career center will be able to post different type of events like job fair, etc. It will show to thousands of companies and students. Students, alumni, companies and career center have a dashboard in which they will be able to see the statistics of different things.

2. Design methodology and software process model

We are using Object Oriented Programming (OOP) methodology for our project Hath Milao. OOP is a programming pattern that uses abstraction as classes and objects in order to create models based real world environment. An object-oriented application has a collection of objects. These objects requests services by passing messages. Objects of object-oriented application are also capable of processing data and receiving messages. Aim of using this methodology is to provide modular structure of programs, increase re-usability, easier to maintain.

1. **Modularity**

   Classes can be written and maintained independently. It allows better organization of code. Breaking a code into chunks makes it easy to tell what this piece of code does
2. **Information Hiding**

   Interaction with only class object's method hides detail of internal structure and implementation of that class which secures our product.

3. **Code Reusability**

   Functionality of one class can be used in different classes. This allows programmers to program fast and test/debug a program easily

3. System overview

3.1 Architectural design

System Architecture is basically a conceptual model that defines the structure, behavior, and views of a system. System architecture is formal description and representation of the system. Architecture provides the infrastructure for the system. There are different architectures to be followed but we are following Model View Controller (MVC). MVC has three different concerns that are as following.

1. **Model**

   Model is the part of the application that handles the logic for the application data. It maintains the data of the application. Model objects retrieve and store model state in a database.

2. **View**

   The view is the part of the application that handles the display of data. Most often the views are created from the model data. User interacts with the UI.

3. **Controller**

   Controller handles the user request. Typically, user interact with View, which in-turn raises appropriate URL request, this request will be handled by a controller. The controller renders the appropriate view with the model data as a response.

We are using the MVC because our system has a lot of interface views. Each user has different views and MVC helps in providing the different views. It makes it easier to understand the system and makes changing in the code easier as it provides the different presentation. The MVC separation helps you manage complex application, because you can focus on one aspect a time. For example, you can focus on the view without depending on the business logic. The MVC separation also simplifies group development.

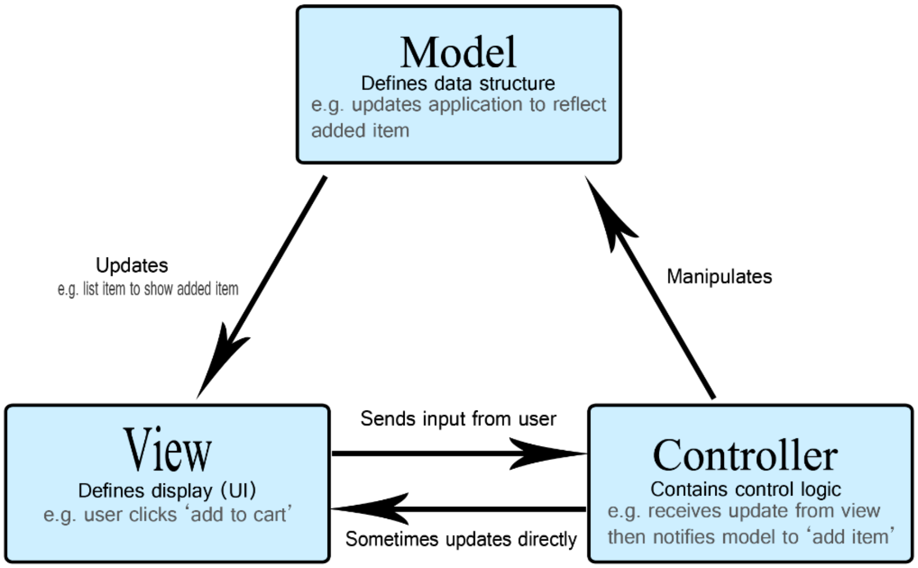

Model
Defines data structure
e.g. updates application to reflect added item
Updates
e.g. list item to show added item
Manipulates
View
Defines display (UI)
e.g. user clicks 'add to cart'
Sends input from user
Sometimes updates directly
Controller
Contains control logic
e.g. receives update from view then notifies model to 'add item'

Figure 1: Architecture diagram

3.2 Block Diagram

A block diagram is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. Following is the block diagram of Hath Milao displaying relationships with the main components of the system.

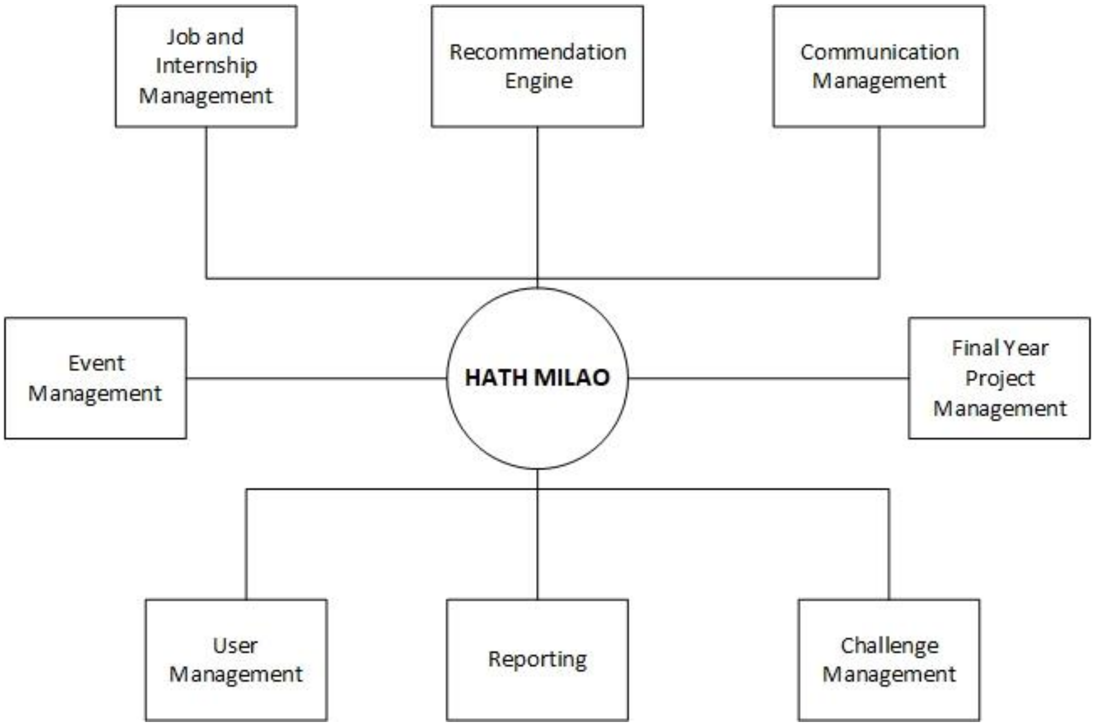


Figure 2: Block Diagram of Hath Milao

3.3 Activity Diagrams

Following are the activity diagram of hath milao.

3.3.1 Add Challenge Activity Diagram

This activity diagram demonstrates how to add challenge in a system.

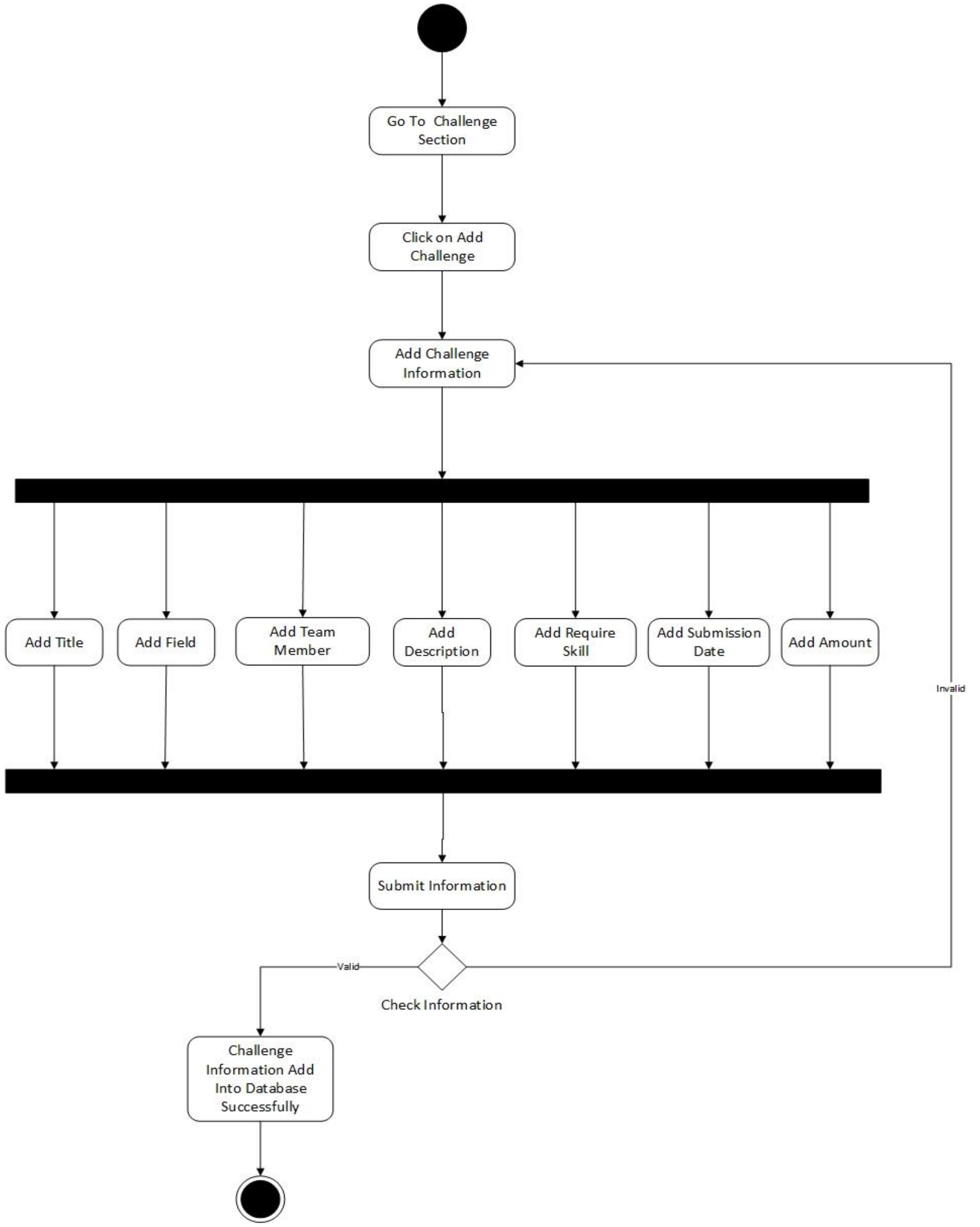


Figure 3: Add Challenge Activity Diagram

3.3.2 Add Education Activity Diagram

This activity diagram demonstrates how to add education in your profile as student.

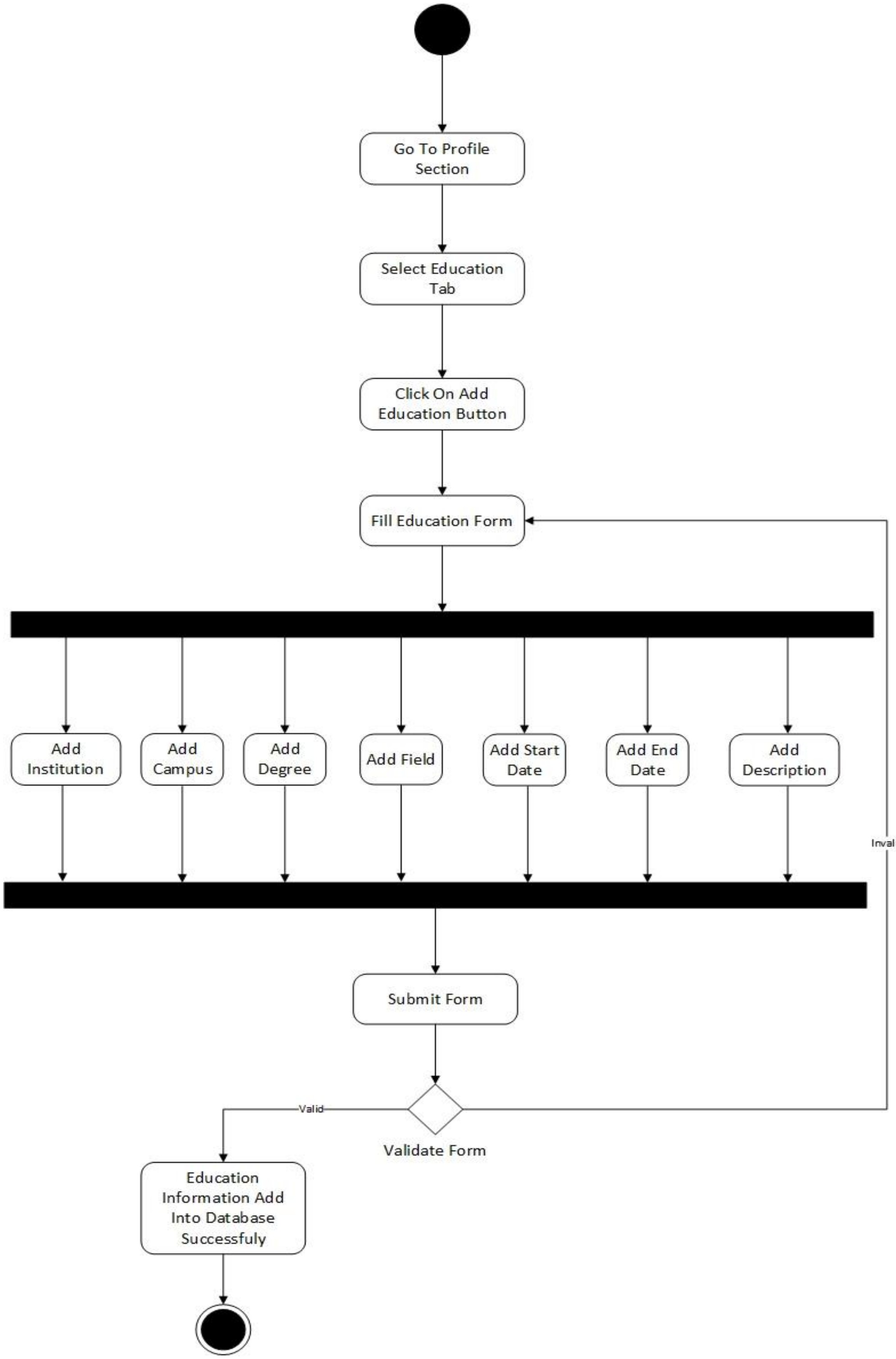


Figure 4: Add Education Activity Diagram

3.3.3 Add Event Activity Diagram

This activity diagram demonstrates how to add event in a system

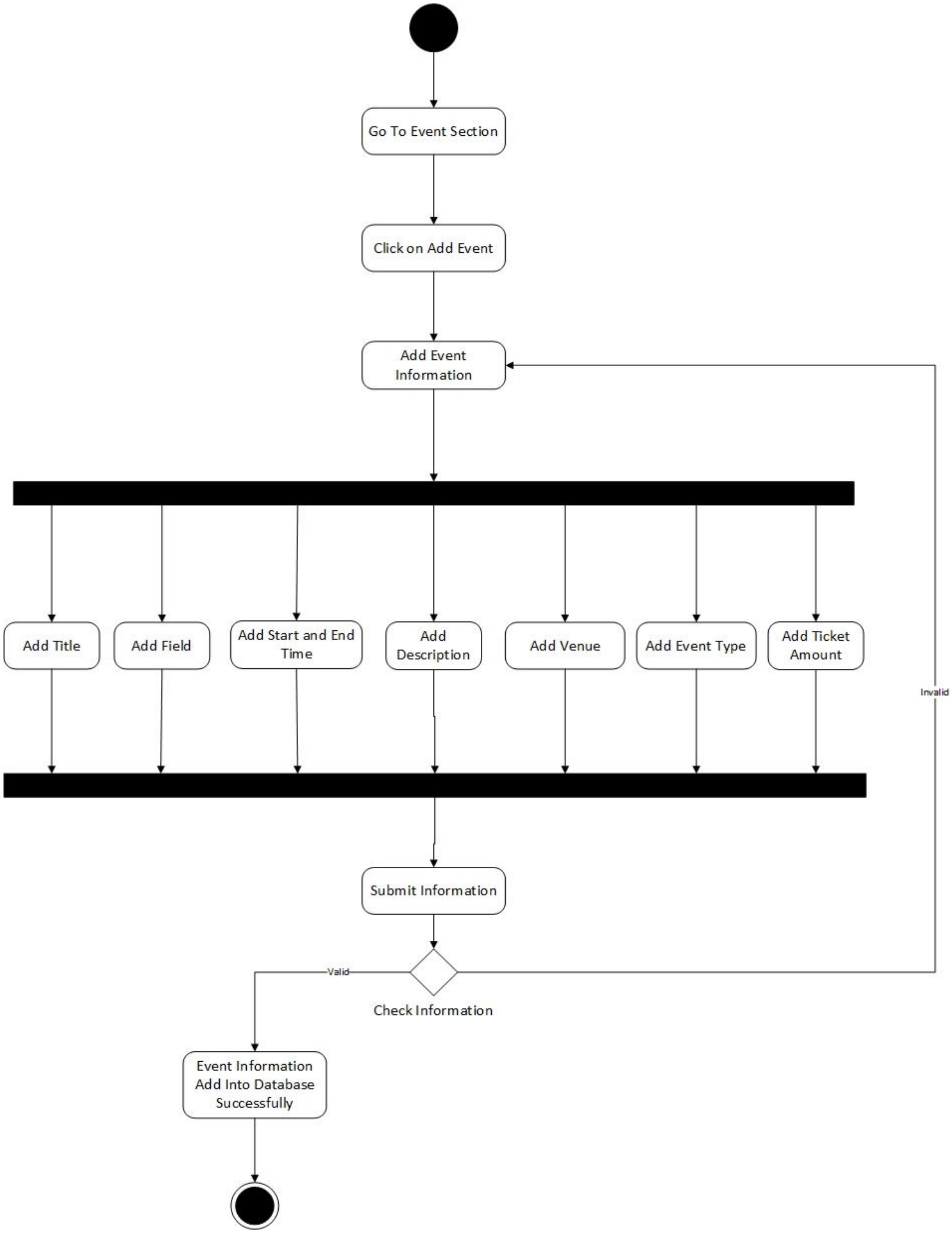


Figure 5: Add Event Activity Diagram

3.3.4 Add Experience Activity Diagram

This activity diagram demonstrates how to add experience in you profile as a student.

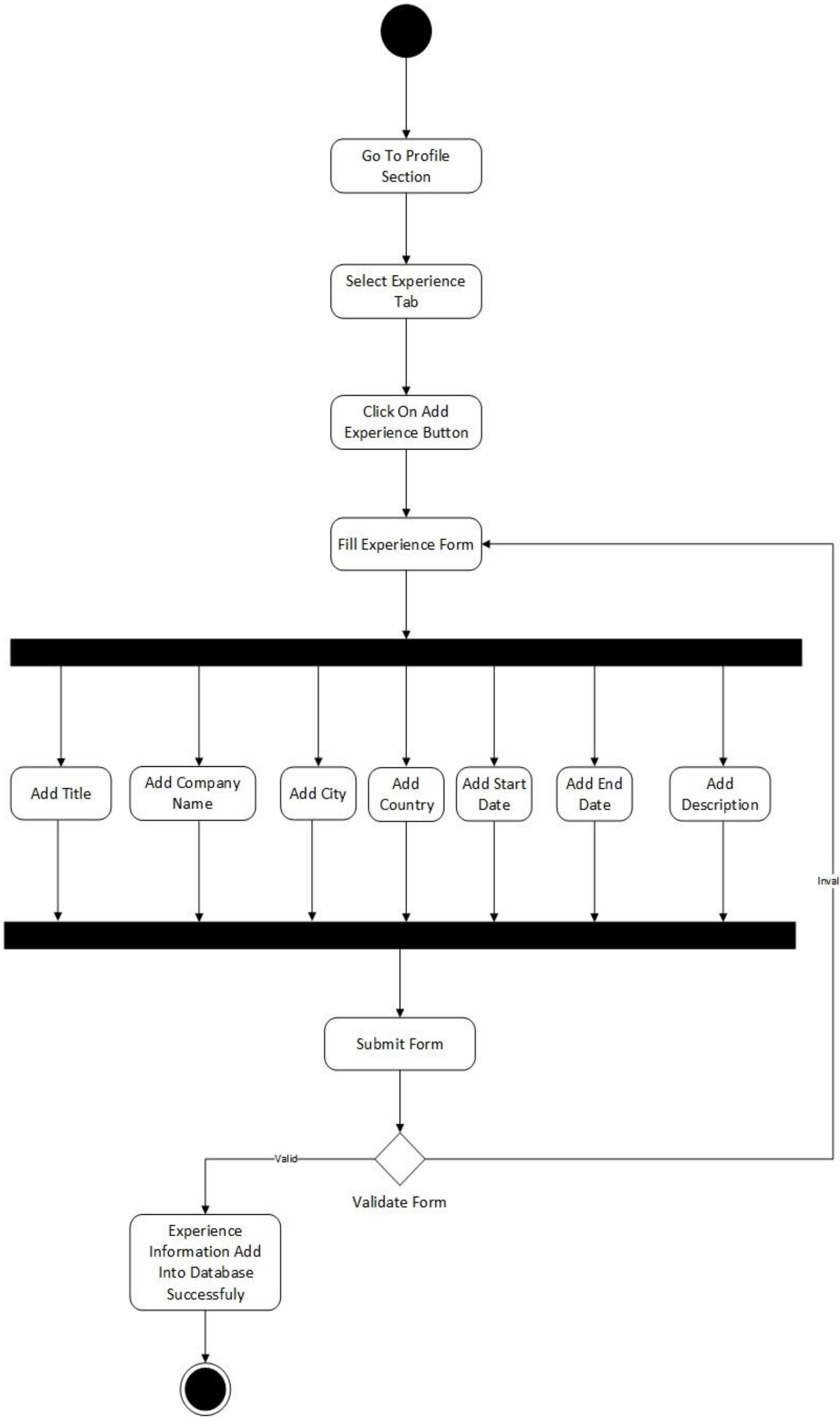


Figure 6: Add Experience Activity Diagram

3.3.5 Add Final Year Project Activity Diagram

This activity diagram demonstrates how to add final year project in a system

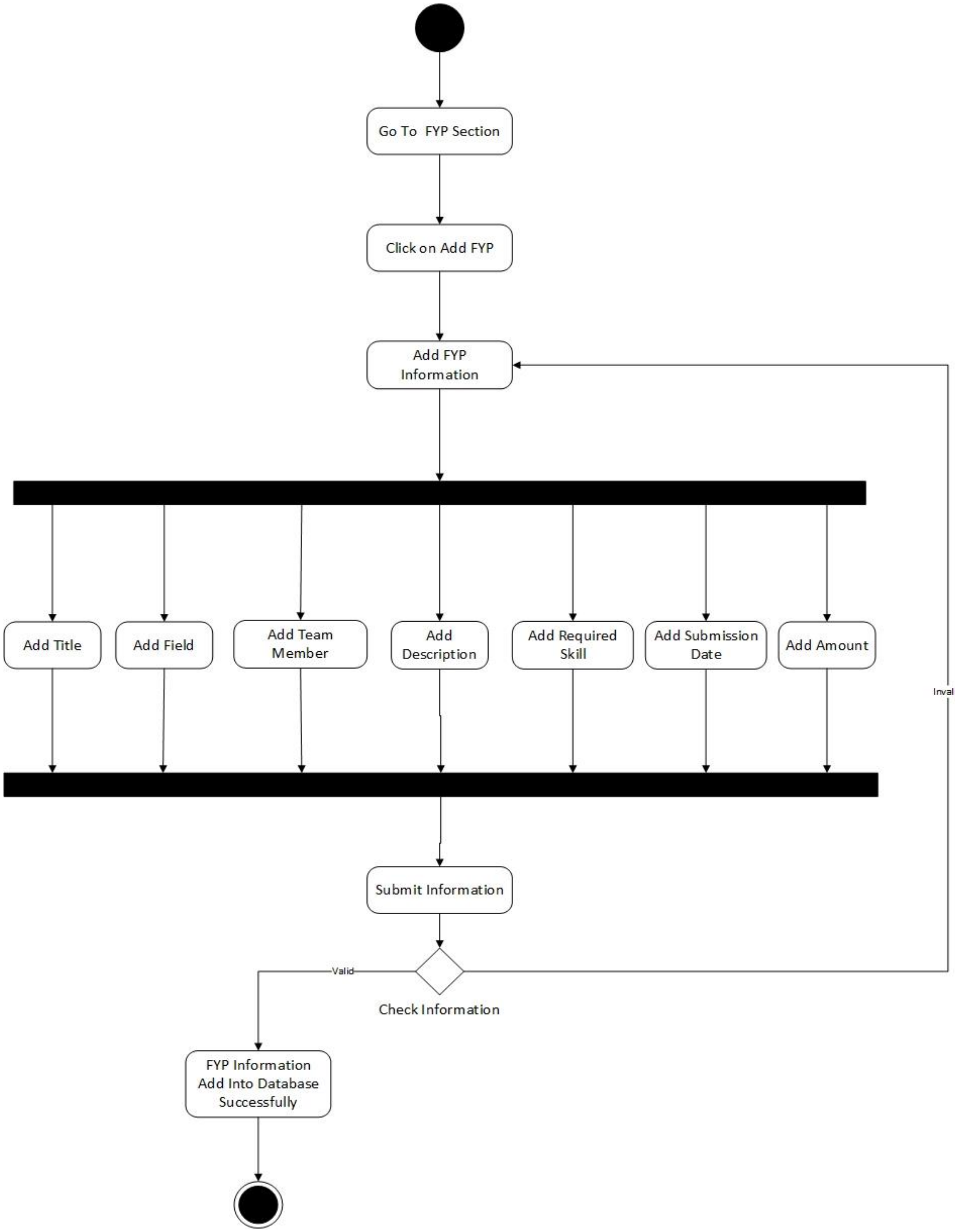


Figure 7: Add Final Year Project Activity Diagram

3.3.6 Add Job and Internship Activity Diagram

This activity diagram demonstrates how to add job and internship in a system

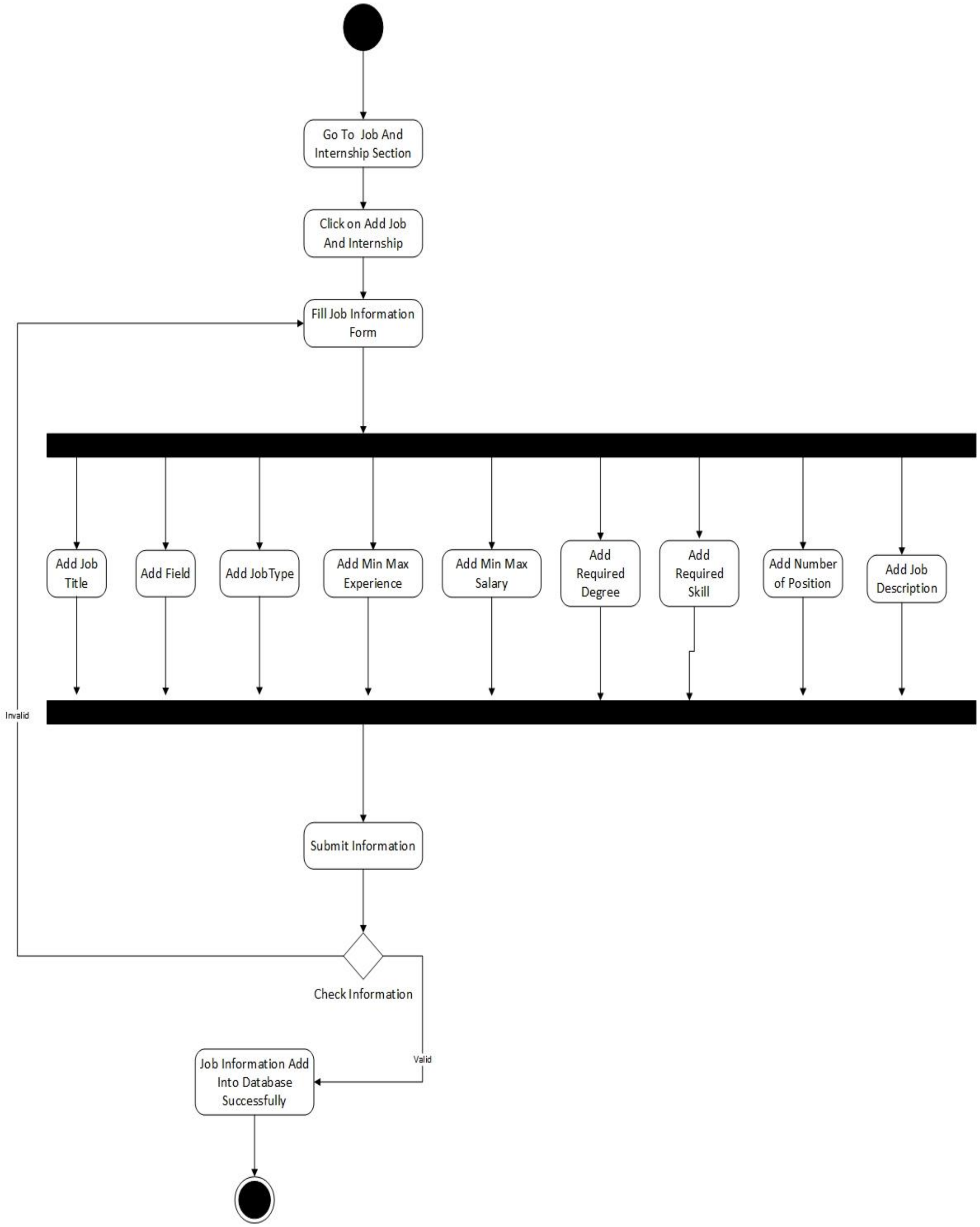


Figure 8: Add Job and Internship Activity Diagram

3.3.7 Cancel Apply Job and Internship Activity Diagram

This activity diagram demonstrates how to cancel apply job and internship.

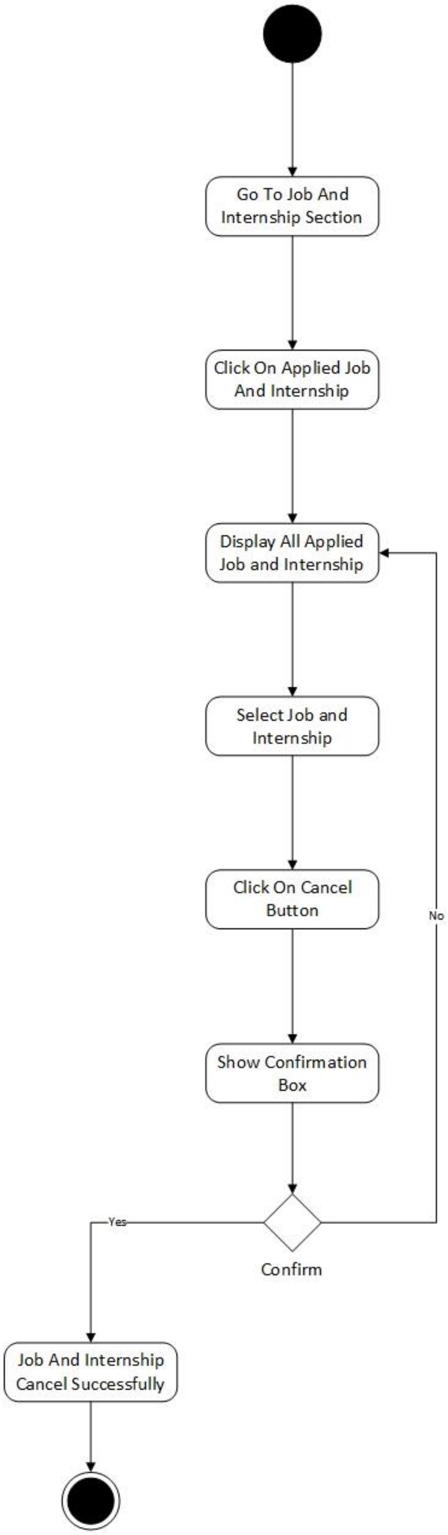

Go To Job And Internship Section
Click On Applied Job And Internship
Display All Applied Job and Internship
Select Job and Internship
Click On Cancel Button
No
Show Confirmation Box
Yes
Confirm
Job And Internship Cancel Successfully

Figure 9: Cancel Apply Job and Internship Activity Diagram

3.3.8 Delete Challenge Activity Diagram

This activity diagram demonstrates how to delete challenge in a system.

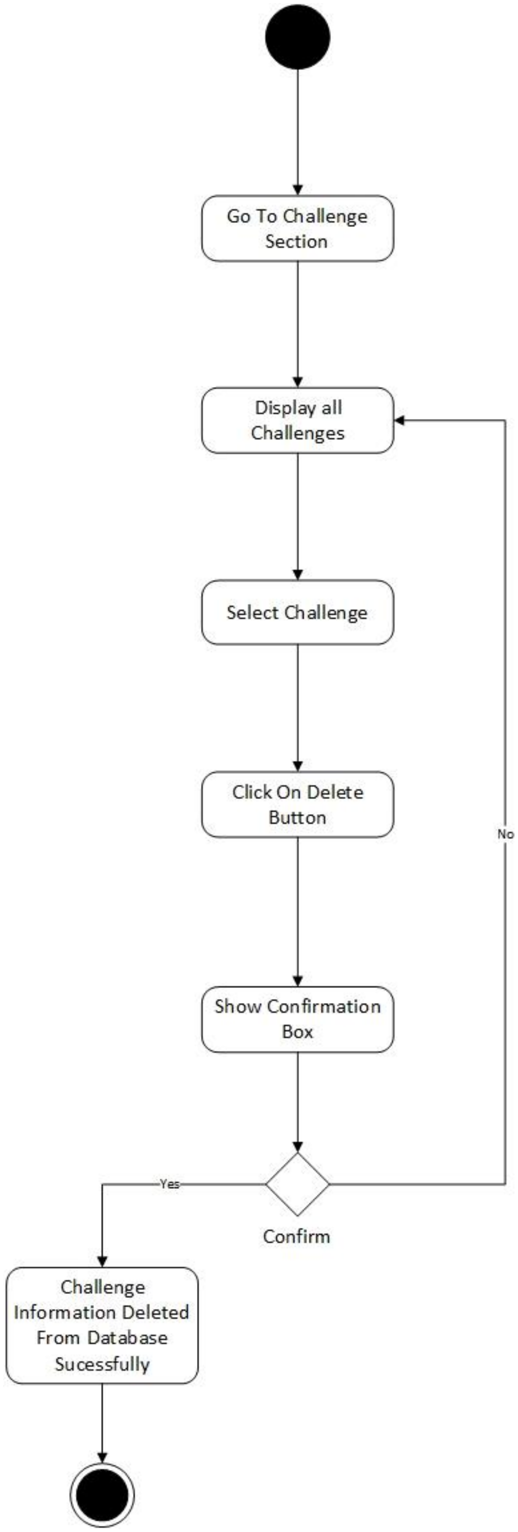


Figure 10: Delete Challenge Activity Diagram

3.3.9 Delete Job and Internship Activity Diagram

This activity diagram demonstrates how to delete Job and Internship in a system.

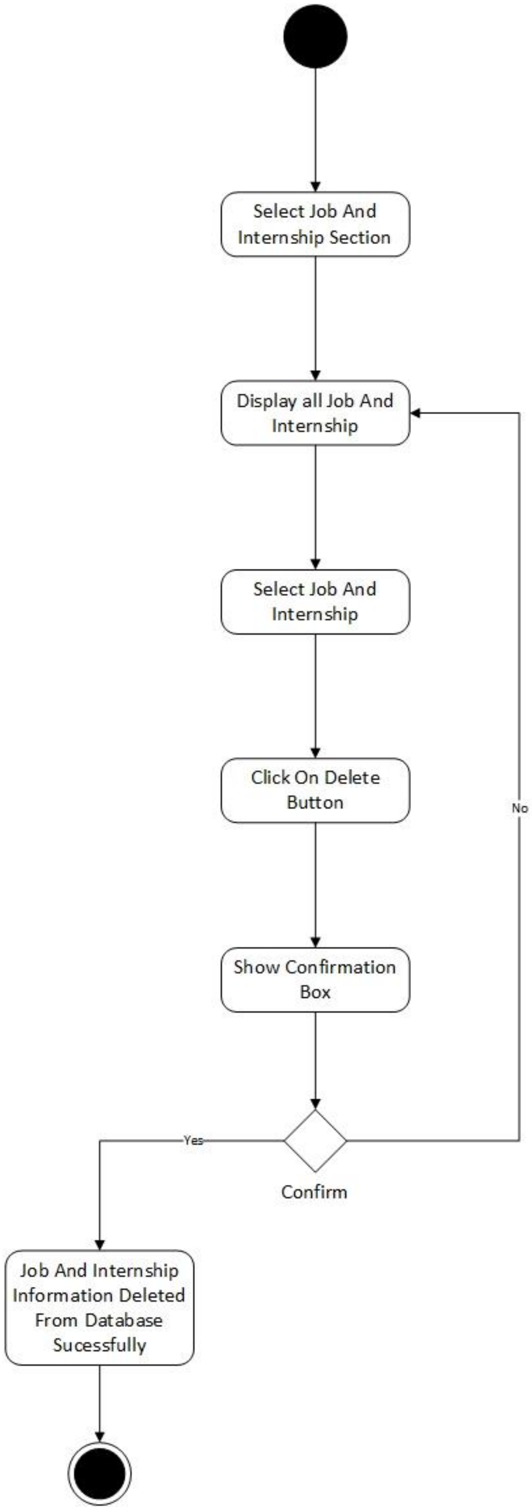

Select Job And Internship Section
Display all Job And Internship
Select Job And Internship
Click On Delete Button
Show Confirmation Box
No
Yes
Confirm
Job And Internship Information Deleted From Database Sucessfully

Figure 11: Delete Job and Internship Activity Diagram

3.3.10 Update Challenge Activity Diagram

This activity diagram demonstrates how to update challenge in a system.

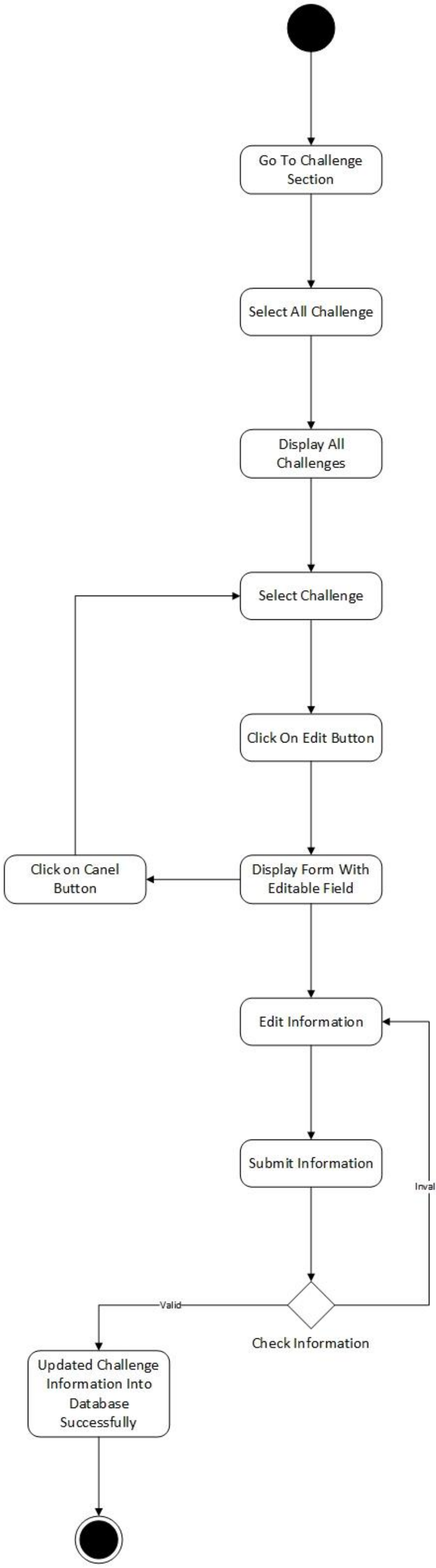


Figure 12: Update Challenge Activity Diagram

3.3.11 Update Event Activity Diagram

This activity diagram demonstrates how to update event in a system.

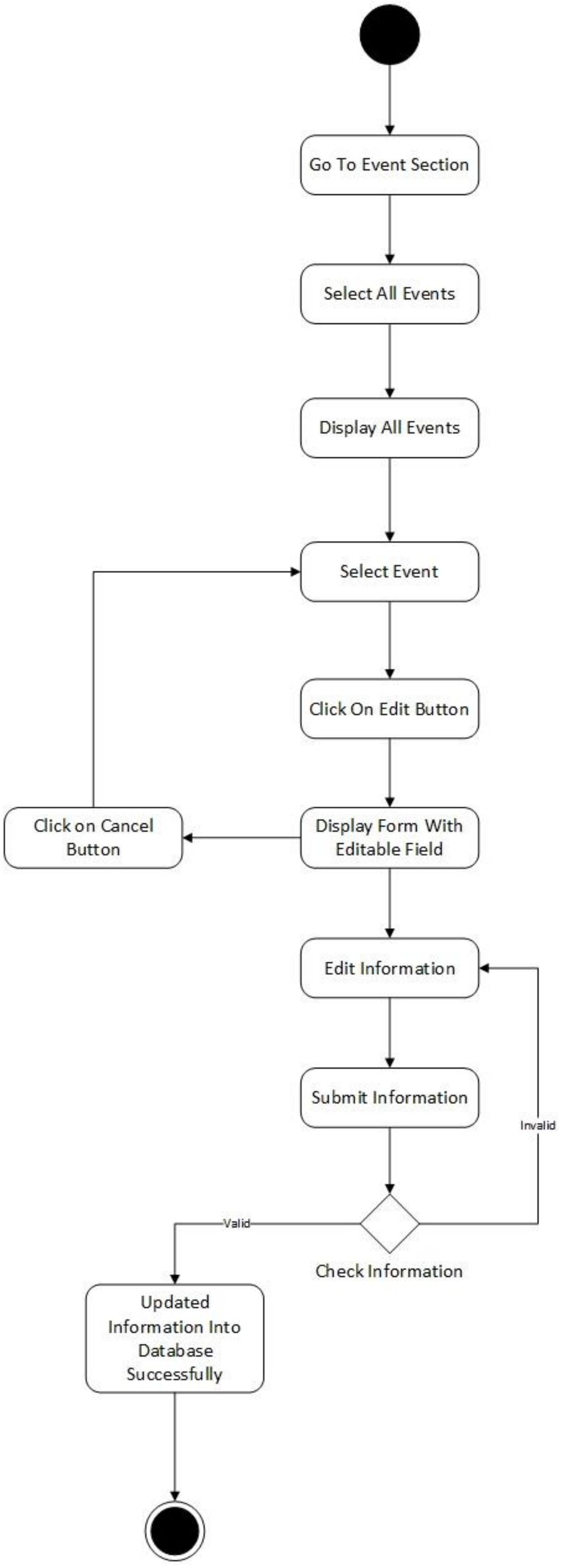


Figure 13: Update Event Activity Diagram

3.3.12 Update Final Year Project Activity Diagram

This activity diagram demonstrates how to update final year project in a system.

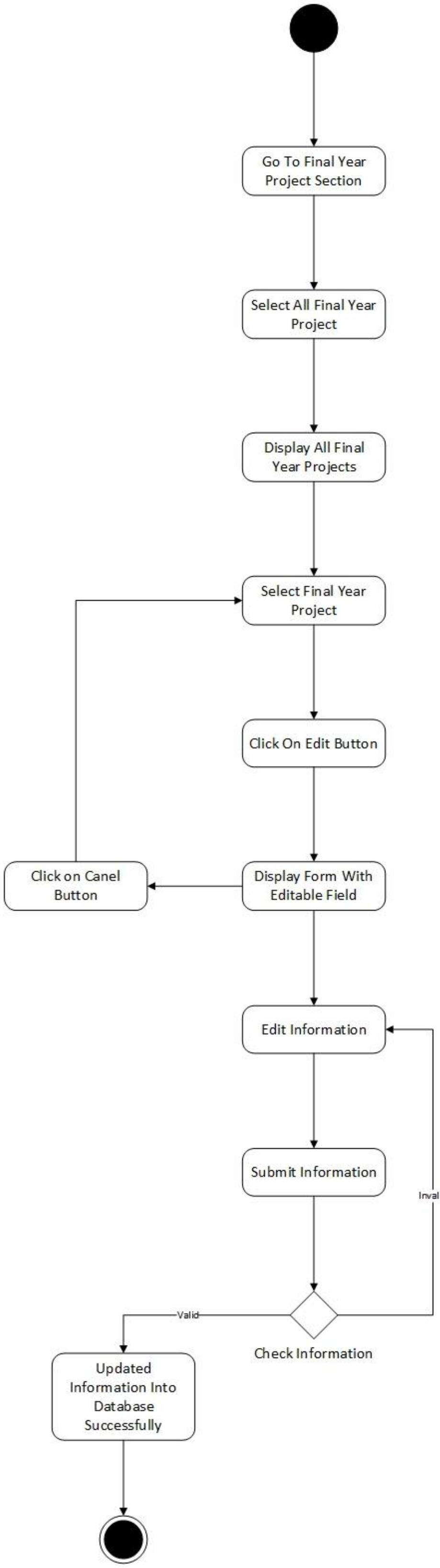

Go To Final Year Project Section
Select All Final Year Project
Display All Final Year Projects
Select Final Year Project
Click On Edit Button
Click on Canel Button
Display Form With Editable Field
Edit Information
Submit Information
Invalid
Valid
Check Information
Updated Information Into Database Successfully

Figure 14: Update Final Year Project Activity Diagram

3.3.13 Follow Company Activity Diagram

This activity diagram demonstrates how to follow a company.

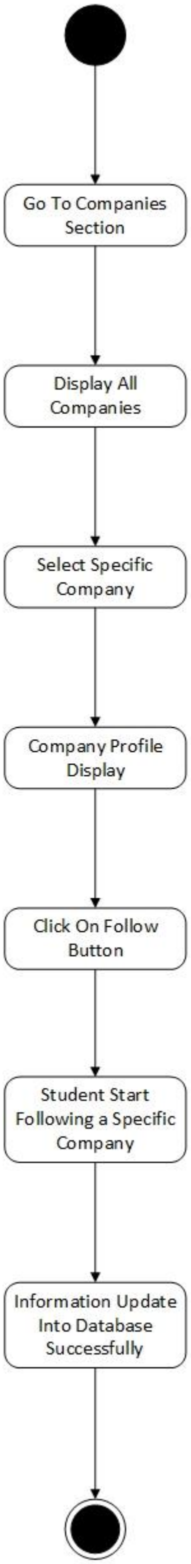


Figure 15: Follow Company Activity Diagram

3.3.14 Submit Solution Activity Diagram

This activity diagram demonstrates how to submit solution of challenge.

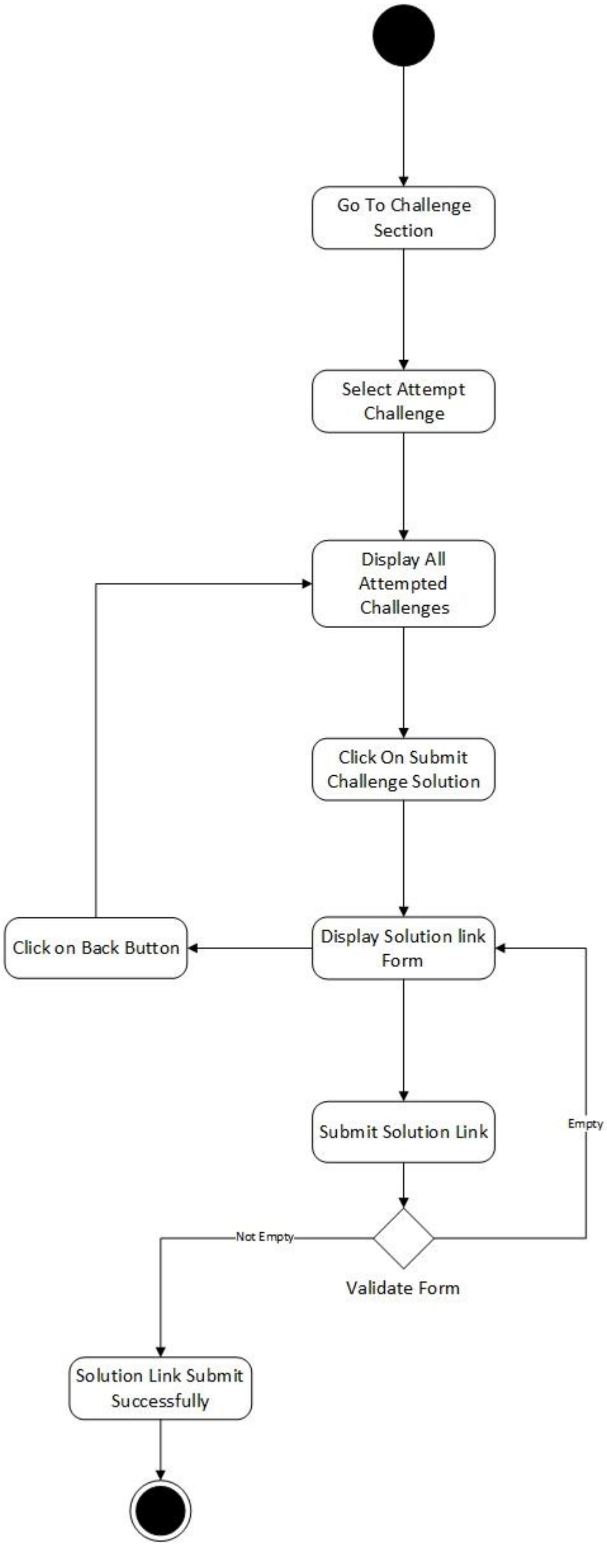

Go To Challenge Section
Select Attempt Challenge
Display All Attempted Challenges
Click On Submit Challenge Solution
Click on Back Button
Display Solution link Form
Submit Solution Link
Empty
Not Empty
Validate Form
Solution Link Submit Successfully

Figure 16: Submit Solution Activity Diagram

3.3.15 Update Education Activity Diagram

This activity diagram demonstrates how to update education in your profile as a student.

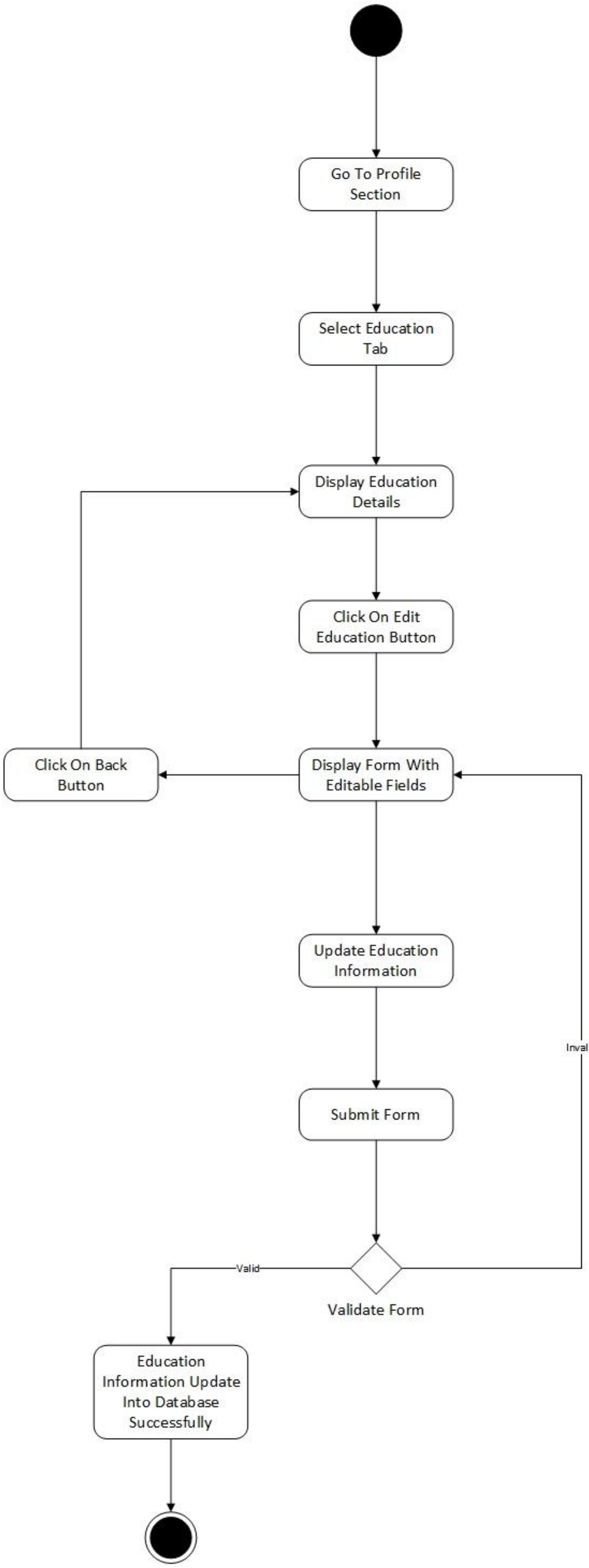

Go To Profile Section
Select Education Tab
Display Education Details
Click On Edit Education Button
Click On Back Button
Display Form With Editable Fields
Update Education Information
Invalid
Submit Form
Valid
Validate Form
Education Information Update Into Database Successfully

Figure 17: Update Education Activity Diagram

3.3.16 Update Experience Activity Diagram

This activity diagram demonstrates how to update experience in your profile as a student.

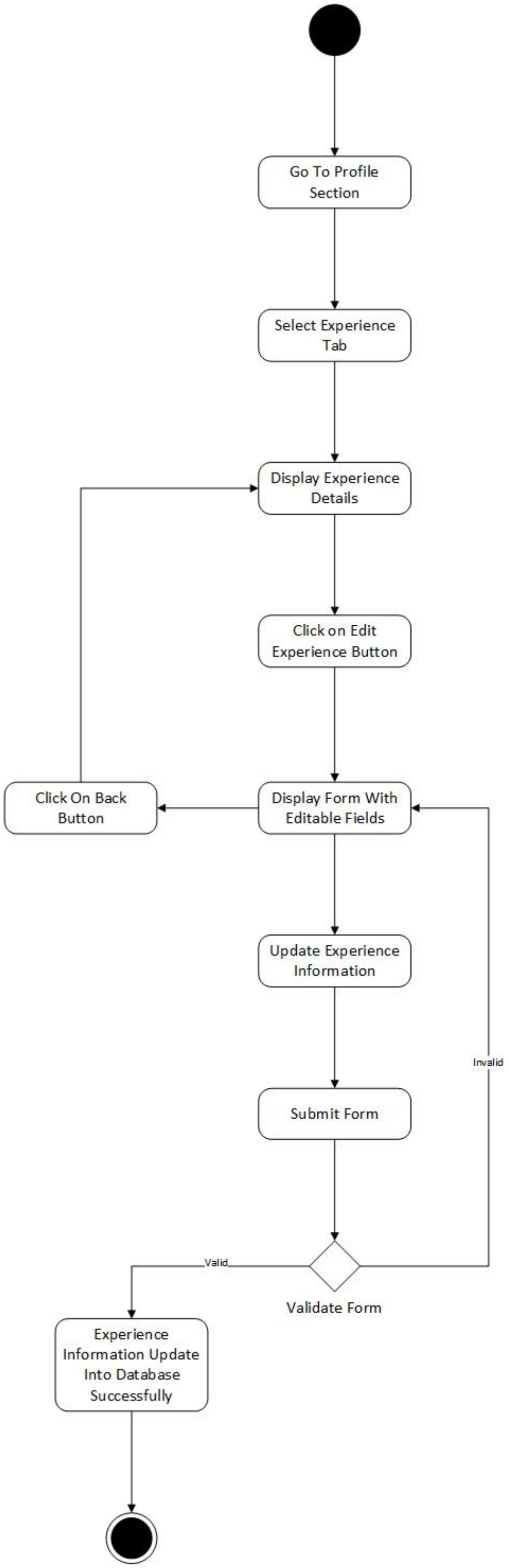

Go To Profile Section
Select Experience Tab
Display Experience Details
Click on Edit Experience Button
Click On Back Button
Display Form With Editable Fields
Update Experience Information
Invalid
Submit Form
Valid
Validate Form
Experience Information Update Into Database Successfully

Figure 18: Update Experience Activity Diagram

3.3.17 View Challenge Activity Diagram

This activity diagram demonstrates how to view challenge.

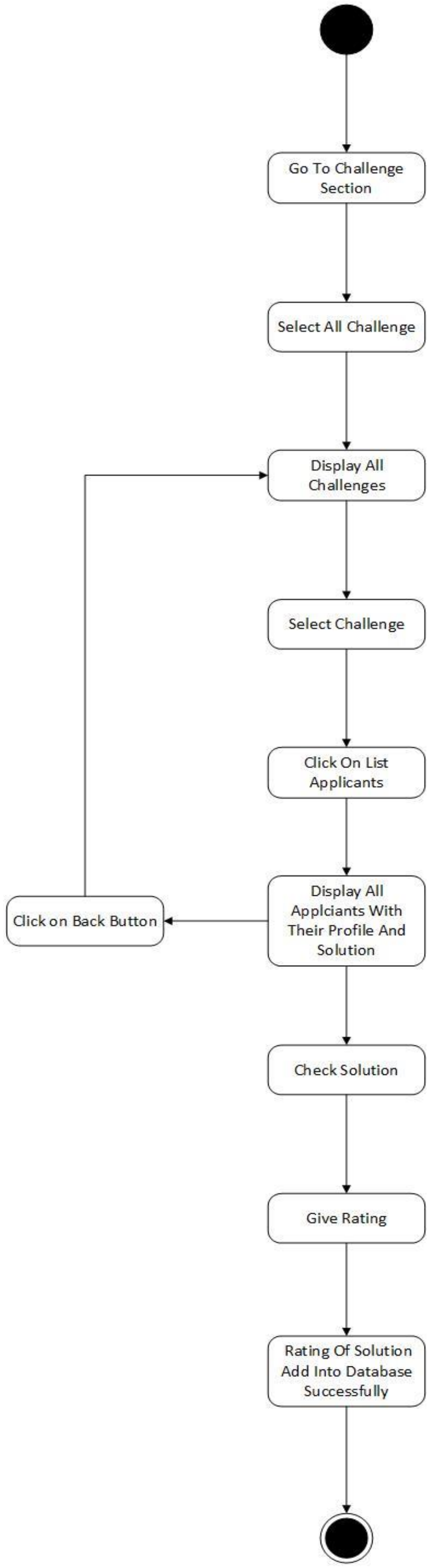


Figure 19: View Challenge Activity Diagram

3.3.18 View Job and Internship Applicants Activity Diagram

This activity diagram demonstrates how to view job and internship applicants' as a company.

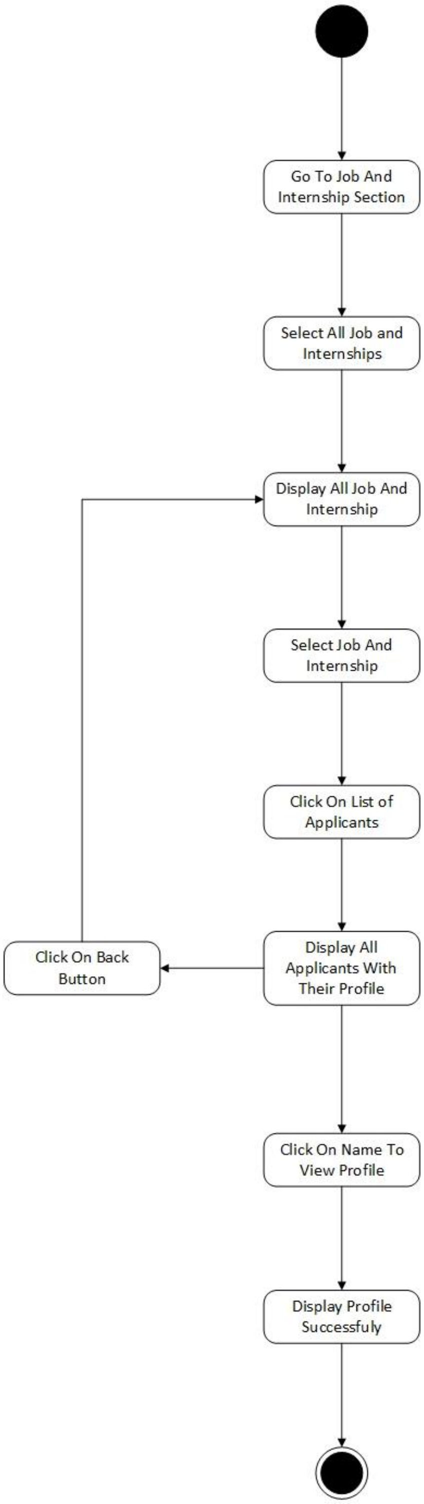

Go To Job And Internship Section
Select All Job and Internships
Display All Job And Internship
Select Job And Internship
Click On List of Applicants
Display All Applicants With Their Profile
Click On Back Button
Click On Name To View Profile
Display Profile Successfuly

Figure 20: View Job and Internship Applicants Activity Diagram

3.3.19 Show Job Fair Activity Diagram

This activity diagram demonstrates how to see job fair.

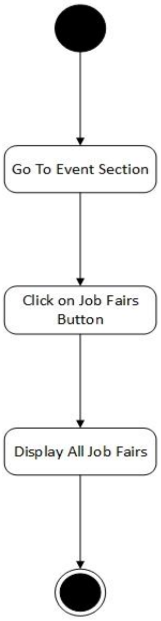


Figure 21: Show Job Fair Activity Diagram

3.3.20 Show Jobless Activity Diagram

This activity diagram demonstrates how to show jobless student.

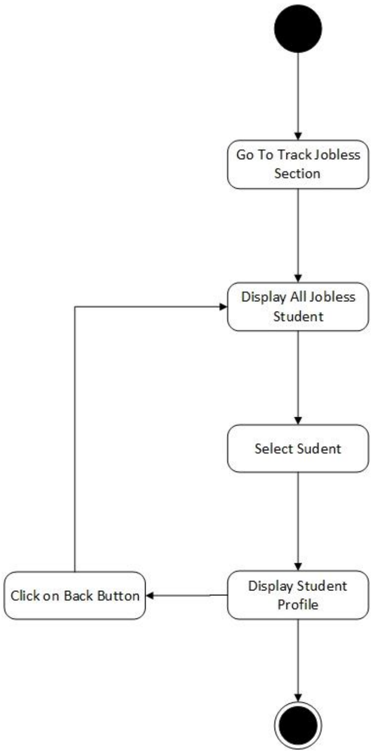


Figure 22: Show Jobless Student Activity Diagram

4. Design models

4.1 Class Diagram

Following is class diagram of hath milao.

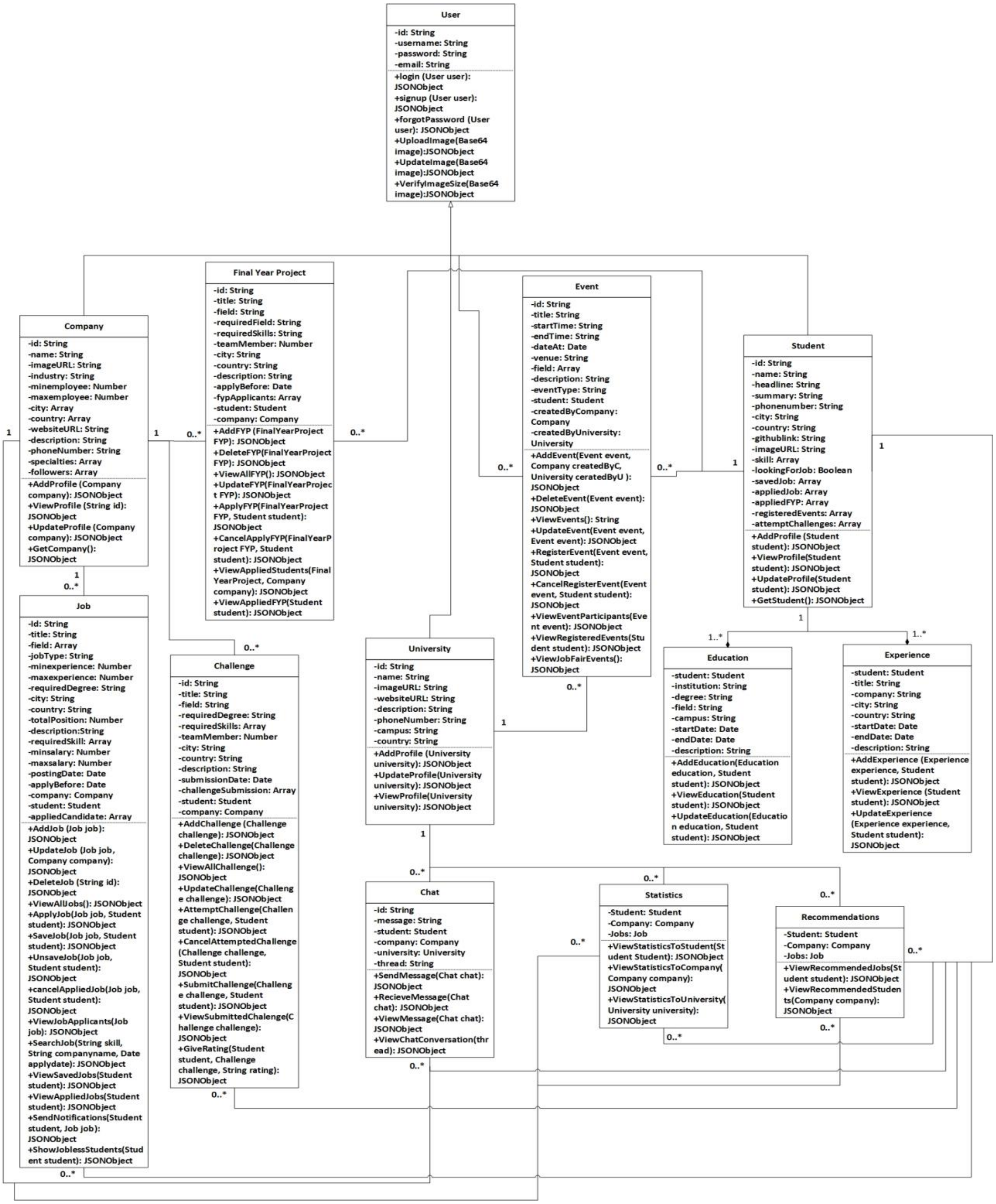

User
-id: String
-username: String
-password: String
-email: String
+login (User user): JSONObject
+signup (User user): JSONObject
+forgotPassword (User user): JSONObject
+UploadImage(Base64 image):JSONObject
+UpdateImage(Base64 image):JSONObject
+VerifyImageSize(Base64 image):JSONObject
Company
-id: String
-name: String
-imageURL: String
-industry: String
-minemployee: Number
-maxemployee: Number
-city: Array
-country: Array
-websiteURL: String
-description: String
-phoneNumber: String
-specialties: Array
-followers: Array
+AddProfile (Company company): JSONObject
+ViewProfile (String id): JSONObject
+UpdateProfile (Company company): JSONObject
+GetCompany(): JSONObject
Final Year Project
-id: String
-title: String
-field: String
-requiredField: String
-requiredSkills: String
-teamMember: Number
-city: String
-country: String
-description: String
-applyBefore: Date
-fypApplicants: Array
-student: Student
-company: Company
+AddFYP (FinalYearProject FYP): JSONObject
+DeleteFYP(FinalYearProject FYP): JSONObject
+ViewAllFYP(): JSONObject
+UpdateFYP(FinalYearProject FYP): JSONObject
+ApplyFYP(FinalYearProject FYP, Student student): JSONObject
+CancelApplyFYP(FinalYearProject FYP, Student student): JSONObject
+ViewAppliedStudents(FinalYearProject, Company company): JSONObject
+ViewAppliedFYP(Student student): JSONObject
Event
-id: String
-title: String
-startTime: String
-endTime: String
-dateAt: Date
-venue: String
-field: Array
-description: String
-eventType: String
-student: Student
-createdByCompany: Company
-createdByUniversity: University
+AddEvent(Event event, Company createdByC, University ceratedByU): JSONObject
+DeleteEvent(Event event): JSONObject
+ViewEvents(): String
+UpdateEvent(Event event, Event event): JSONObject
+RegisterEvent(Event event, Student student): JSONObject
+CancelRegisterEvent(Event event, Student student): JSONObject
+ViewEventParticipants(Event event): JSONObject
+ViewRegisteredEvents(Student student): JSONObject
+ViewJobFairEvents(): JSONObject
Student
-id: String
-name: String
-headline: String
-summary: String
-phonenumber: String
-city: String
-country: String
-githublink: String
-imageURL: String
-skill: Array
-lookingForJob: Boolean
-savedJob: Array
-appliedJob: Array
-appliedFYP: Array
-registeredEvents: Array
-attemptChallenges: Array
+AddProfile (Student student): JSONObject
+ViewProfile(Student student): JSONObject
+UpdateProfile(Student student): JSONObject
+GetStudent(): JSONObject
Job
-id: String
-title: String
-field: Array
-jobType: String
-minexperience: Number
-maxexperience: Number
-requiredDegree: String
-city: String
-country: String
-totalPosition: Number
-description:String
-requiredSkill: Array
-minsalary: Number
-maxsalary: Number
-postingDate: Date
-applyBefore: Date
-company: Company
-student: Student
-appliedCandidate: Array
+AddJob (Job job): JSONObject
+UpdateJob (Job job, Company company): JSONObject
+DeleteJob (String id): JSONObject
+ViewAllJobs(): JSONObject
+ApplyJob(Job job, Student student): JSONObject
+SaveJob(Job job, Student student): JSONObject
+UnsaveJob(Job job, Student student): JSONObject
+cancelAppliedJob(Job job, Student student): JSONObject
+ViewJobApplicants(Job job): JSONObject
+SearchJob(String skill, String companyname, Date applydate): JSONObject
+ViewSavedJobs(Student student): JSONObject
+ViewAppliedJobs(Student student): JSONObject
+SendNotifications(Student student, Job job): JSONObject
+ShowJoblessStudents(Student student): JSONObject
Challenge
-id: String
-title: String
-field: String
-requiredDegree: String
-requiredSkills: Array
-teamMember: Number
-city: String
-country: String
-description: String
-submissionDate: Date
-challengeSubmission: Array
-student: Student
-company: Company
+AddChallenge (Challenge challenge): JSONObject
+DeleteChallenge(Challenge challenge): JSONObject
+ViewAllChallenge(): JSONObject
+UpdateChallenge(Challenge challenge): JSONObject
+AttemptChallenge(Challenge challenge, Student student): JSONObject
+CancelAttemptedChallenge (Challenge challenge, Student student): JSONObject
+SubmitChallenge(Challenge challenge, Student student): JSONObject
+ViewSubmittedChalenge(Challenge challenge): JSONObject
+GiveRating(Student student, Challenge challenge, String rating): JSONObject
University
-id: String
-name: String
-imageURL: String
-websiteURL: String
-description: String
-phoneNumber: String
-campus: String
-country: String
+AddProfile (University university): JSONObject
+UpdateProfile(University university): JSONObject
+ViewProfile(University university): JSONObject
Education
-student: Student
-institution: String
-degree: String
-field: String
-campus: String
-startDate: Date
-endDate: Date
-description: String
+AddEducation(Education education, Student student): JSONObject
+ViewEducation(Student student): JSONObject
+UpdateEducation(Education education, Student student): JSONObject
Experience
-student: Student
-title: String
-company: String
-city: String
-country: String
-startDate: Date
-endDate: Date
-description: String
+AddExperience (Experience experience, Student student): JSONObject
+ViewExperience (Student student): JSONObject
+UpdateExperience (Experience experience, Student student): JSONObject
Chat
-id: String
-message: String
-student: Student
-company: Company
-university: University
-thread: String
+SendMessage(Chat chat): JSONObject
+RecieveMessage(Chat chat): JSONObject
+ViewMessage(Chat chat): JSONObject
+ViewChatConversation(thread): JSONObject
Statistics
-Student: Student
-Company: Company
-Jobs: Job
+ViewStatisticsToStudent(Student Student): JSONObject
+ViewStatisticsToCompany(Company company): JSONObject
+ViewStatisticsToUniversity(University university): JSONObject
Recommendations
-Student: Student
-Company: Company
-Jobs: Job
+ViewRecommendedJobs(Student student): JSONObject
+ViewRecommendedStudents(Company company): JSONObject
1
1
0..*
0..*
1
0..*
0..*
0..*
1
1
1..*
1..*
1
1
0..*
1
0..*
0..*
0..*
0..*
0..*
0..*
0..*
0..*
0..*
0..*

Figure 23: Class Diagram Hath Milao

4.2 Sequence Diagram

4.2.1 Company Challenge Management

In this diagram we explain the interaction of company with challenge class.

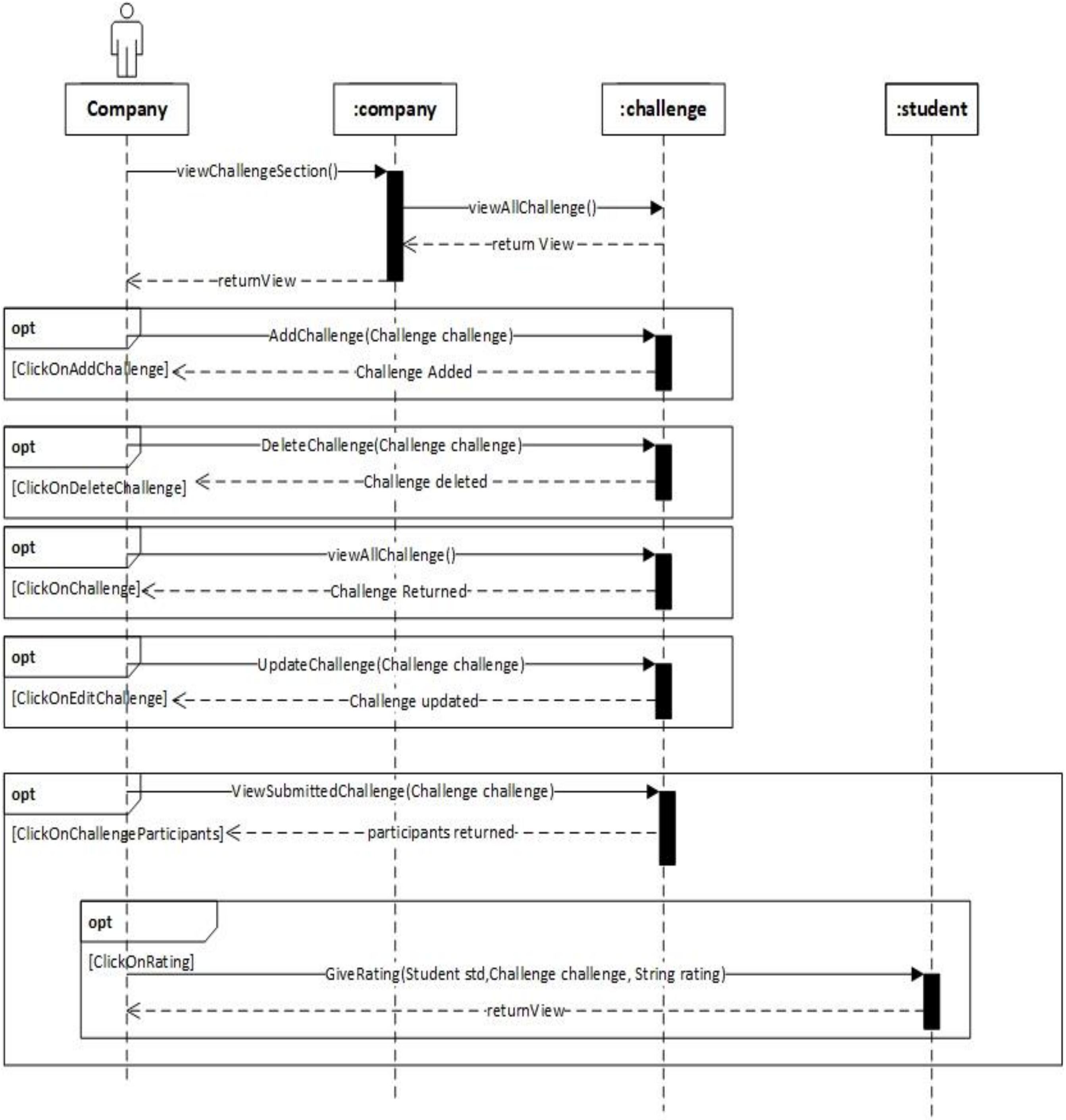


Figure 24: Company Challenge Management Sequence Diagram

4.2.2 Company Event Management

In this diagram we explain the interaction of company with event class.

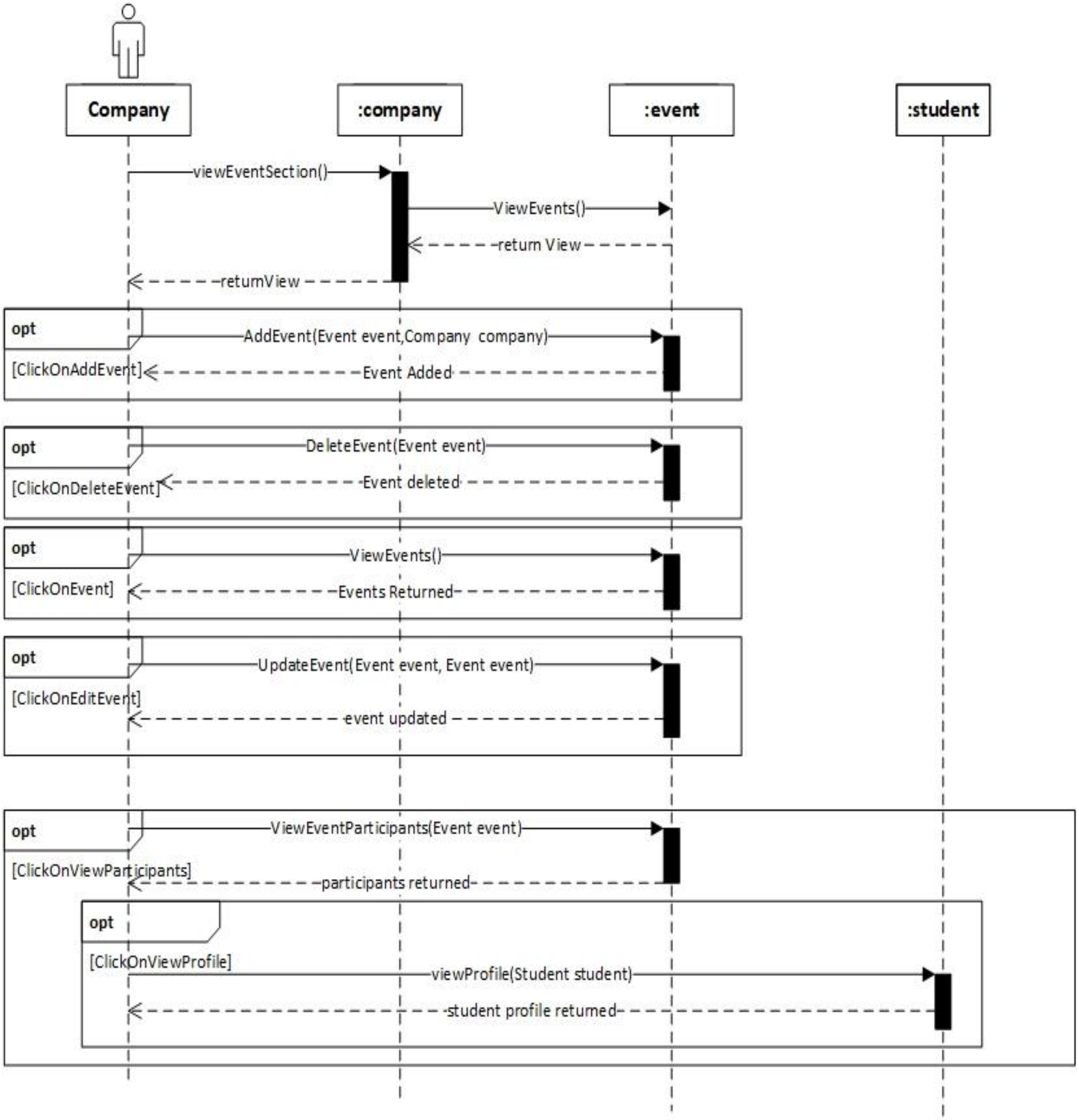


Figure 25: Company Event Sequence Management Diagram

4.2.3 Company Final Year Project Management

In this diagram we explain the interaction of company with final year project class.

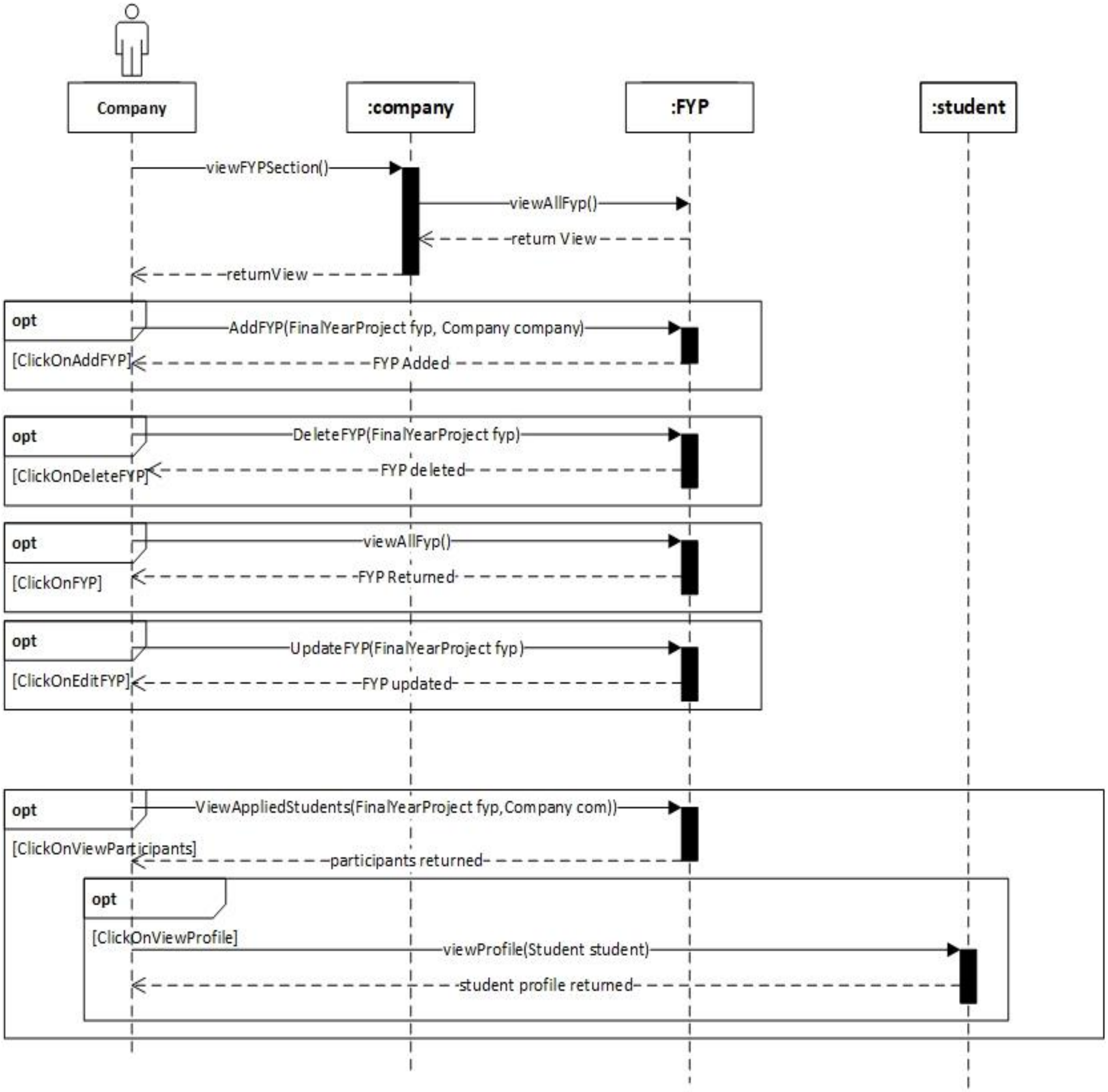


Figure 26: Company Final Year Project Management Sequence Diagram

4.2.4 Company Job and Internship Management

In this diagram we explain the interaction of company with job class

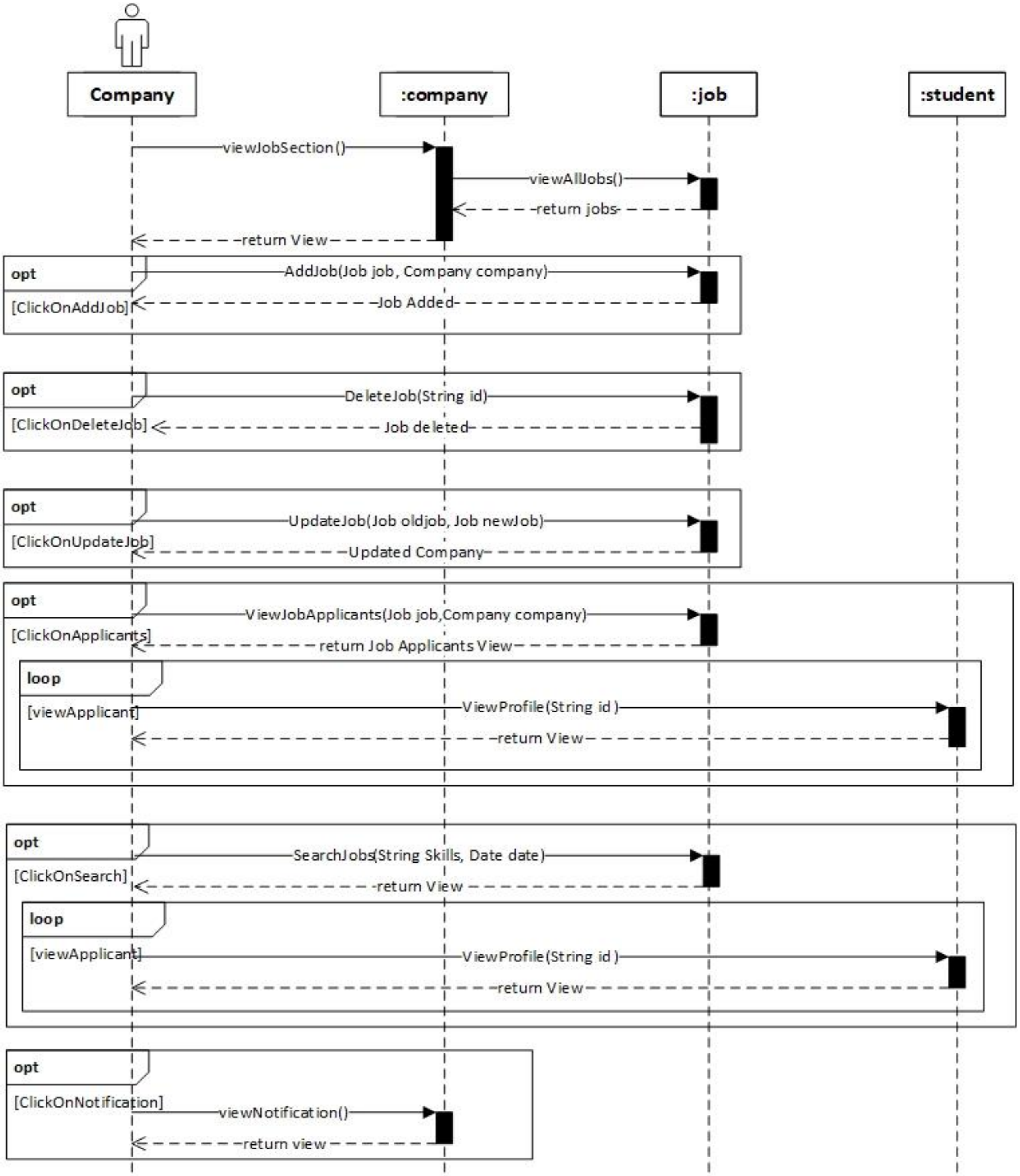


Figure 27: Company Job and Internship Management Sequence Diagram

4.2.5 Student Challenge Management

In this diagram we explain the interaction of student with challenge class

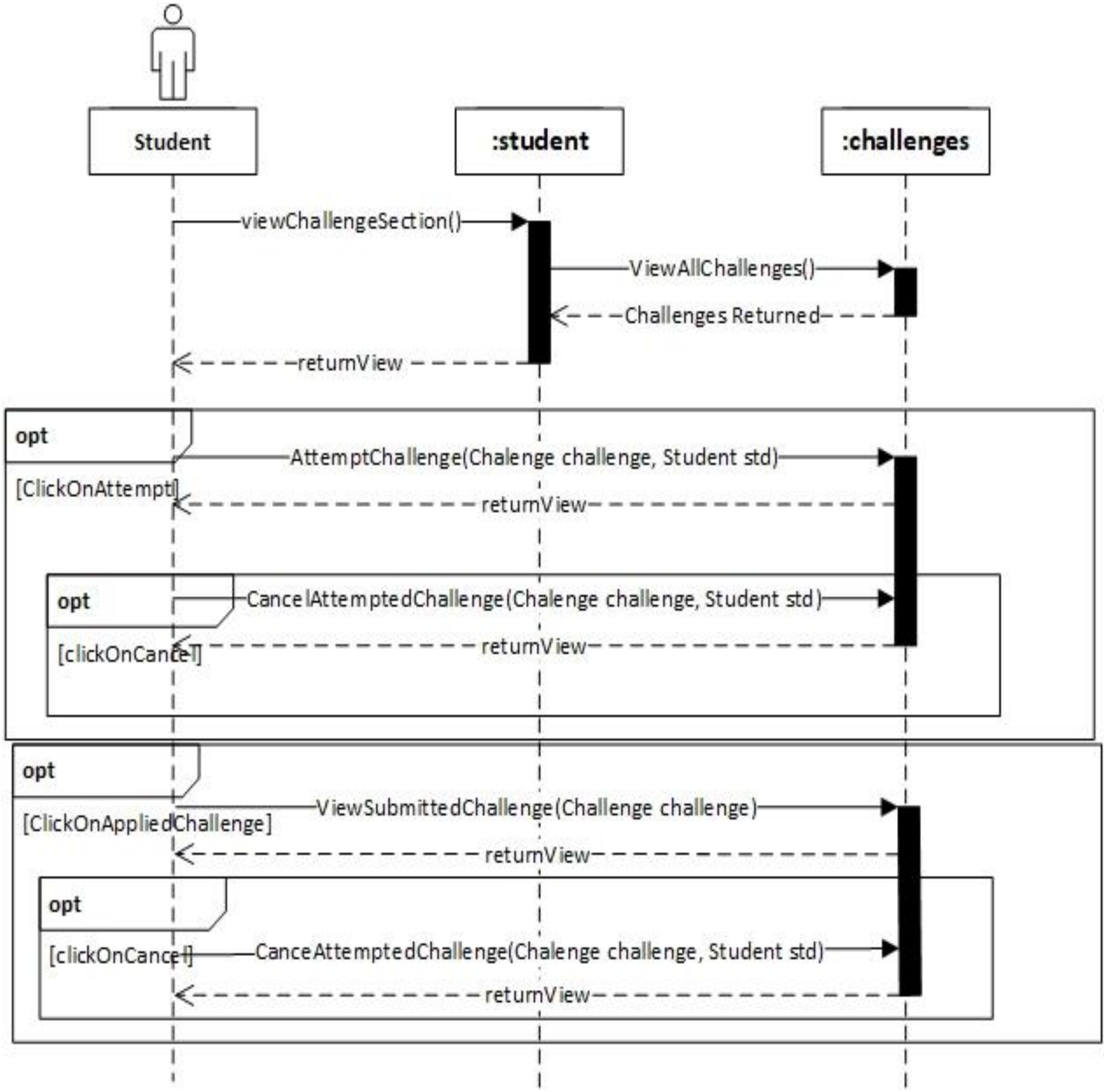

Student
:student
:challenges
viewChallengeSection()
ViewAllChallenges()
Challenges Returned
returnView
opt
[ClickOnAttempt]
AttemptChallenge(Chalenge challenge, Student std)
returnView
opt
[clickOnCancel]
CancelAttemptedChallenge(Chalenge challenge, Student std)
returnView
opt
[ClickOnAppliedChallenge]
ViewSubmittedChallenge(Challenge challenge)
returnView
opt
[clickOnCancel]
CanceAttemptedChallenge(Chalenge challenge, Student std)
returnView

Figure 28: Student Challenge Management Sequence Diagram

4.2.6 Student Event Management

In this diagram we explain the interaction of student with event class.

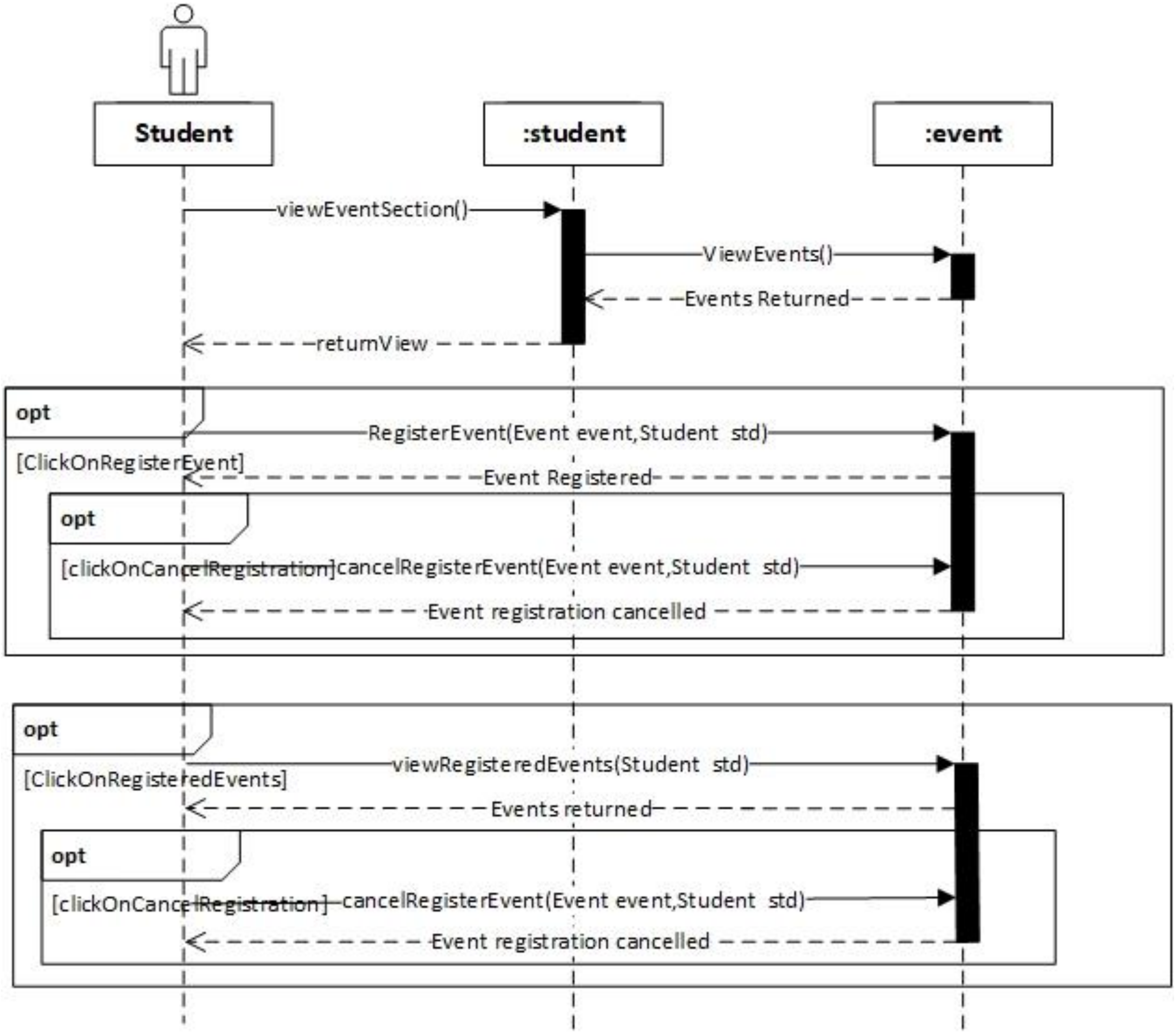

Student
:student
:event
viewEventSection()
ViewEvents()
Events Returned
returnView
opt
[ClickOnRegisterEvent]
RegisterEvent(Event event,Student std)
Event Registered
opt
[clickOnCancelRegistration]
cancelRegisterEvent(Event event,Student std)
Event registration cancelled
opt
[ClickOnRegisteredEvents]
viewRegisteredEvents(Student std)
Events returned
opt
[clickOnCancelRegistration]
cancelRegisterEvent(Event event,Student std)
Event registration cancelled

Figure 29: Student Event Management Sequence Diagram

4.2.7 Student Final Year Project Management

In this diagram we explain the interaction of student with final year project class.

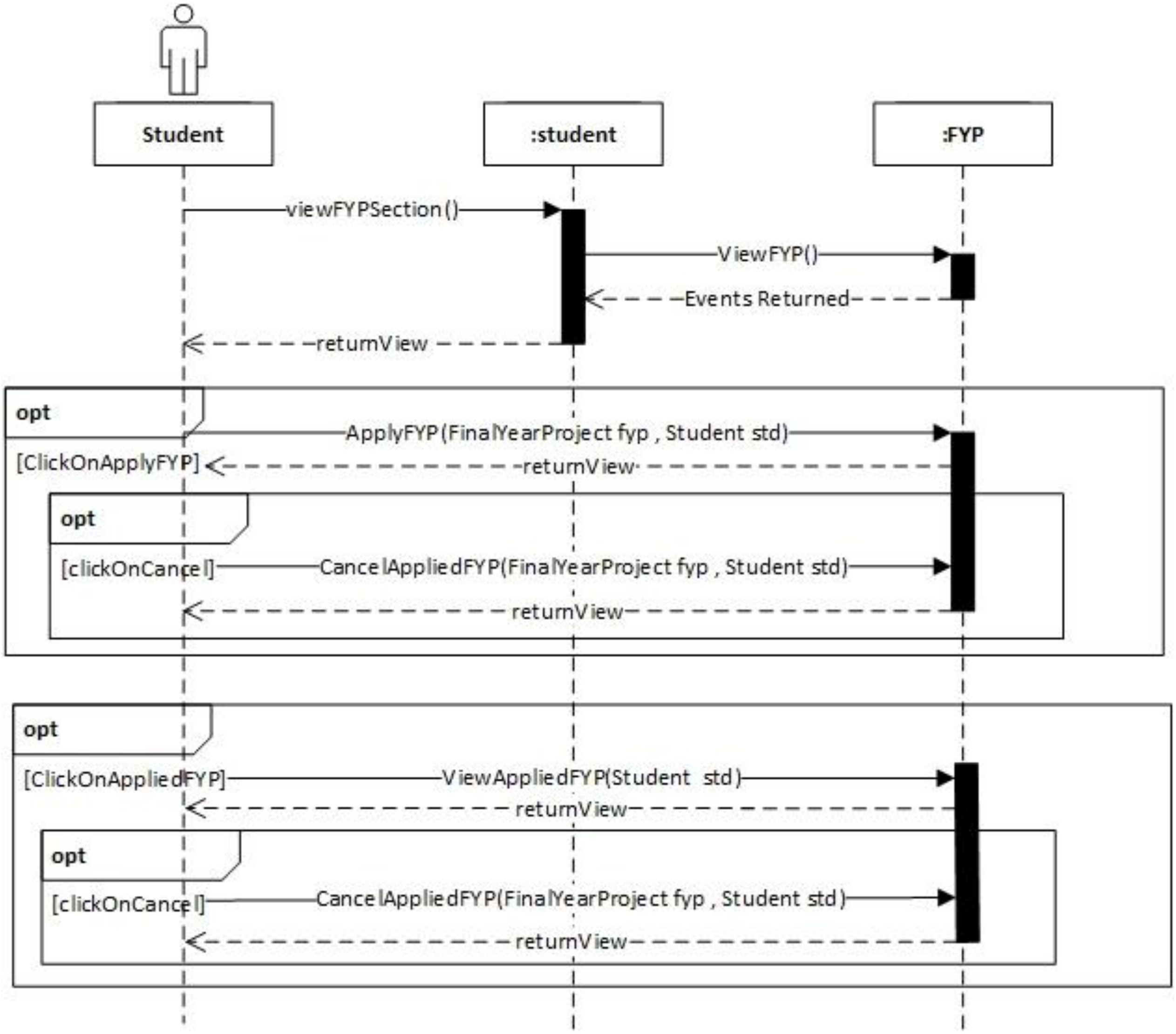


Figure 30: Student Final Year Project Management Sequence Diagram

4.2.8 Student Job and Internship Management

In this diagram we explain the interaction of student with job class

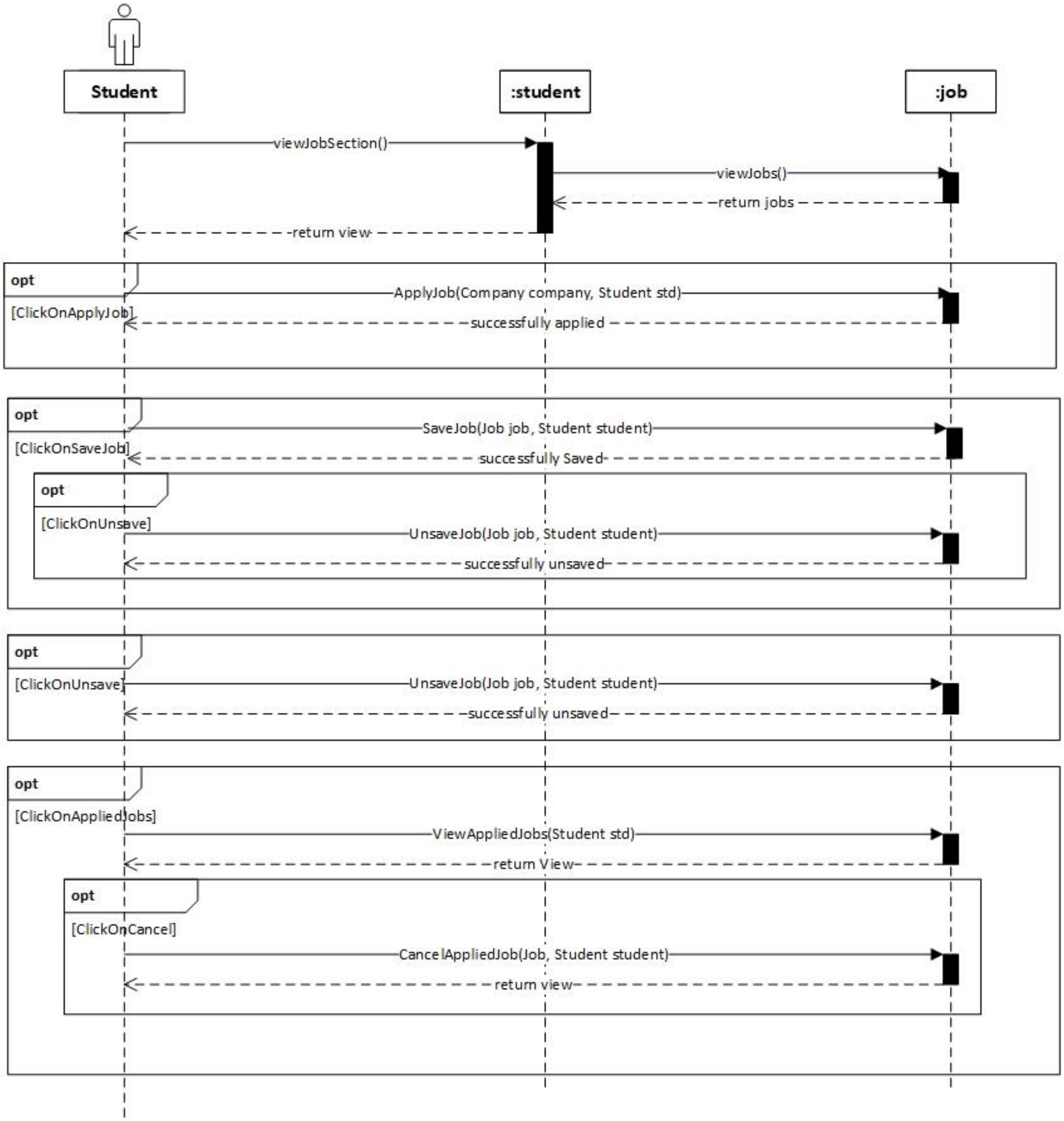


Figure 31: Student Job and Internship Management Sequence Diagram

4.2.9 University Event Management

In this diagram we explain the interaction of university with event class.

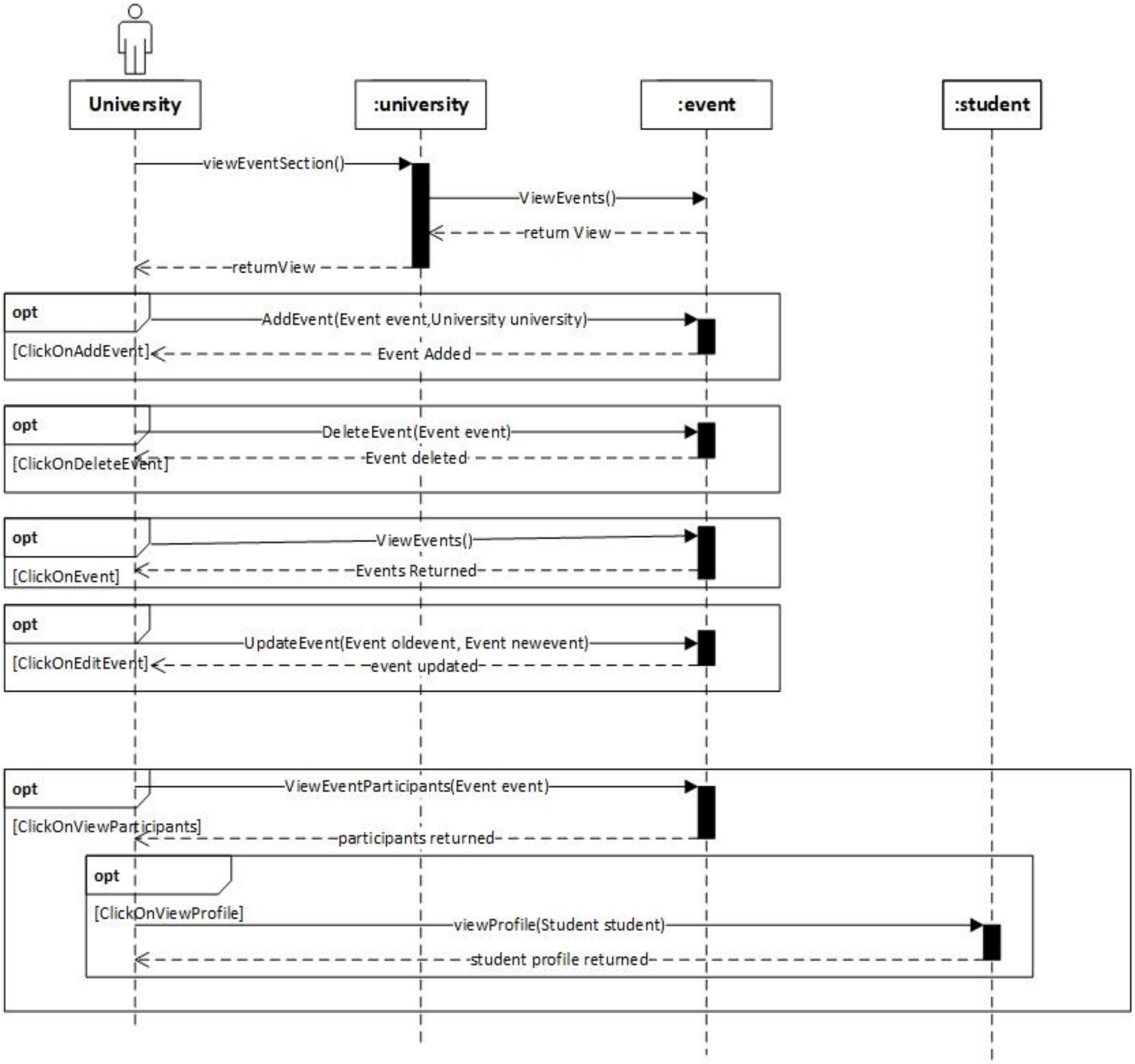


Figure 32: University Event Management Sequence Diagram

4.2.10 Chat

In this diagram we explain the interaction of users with chat class.

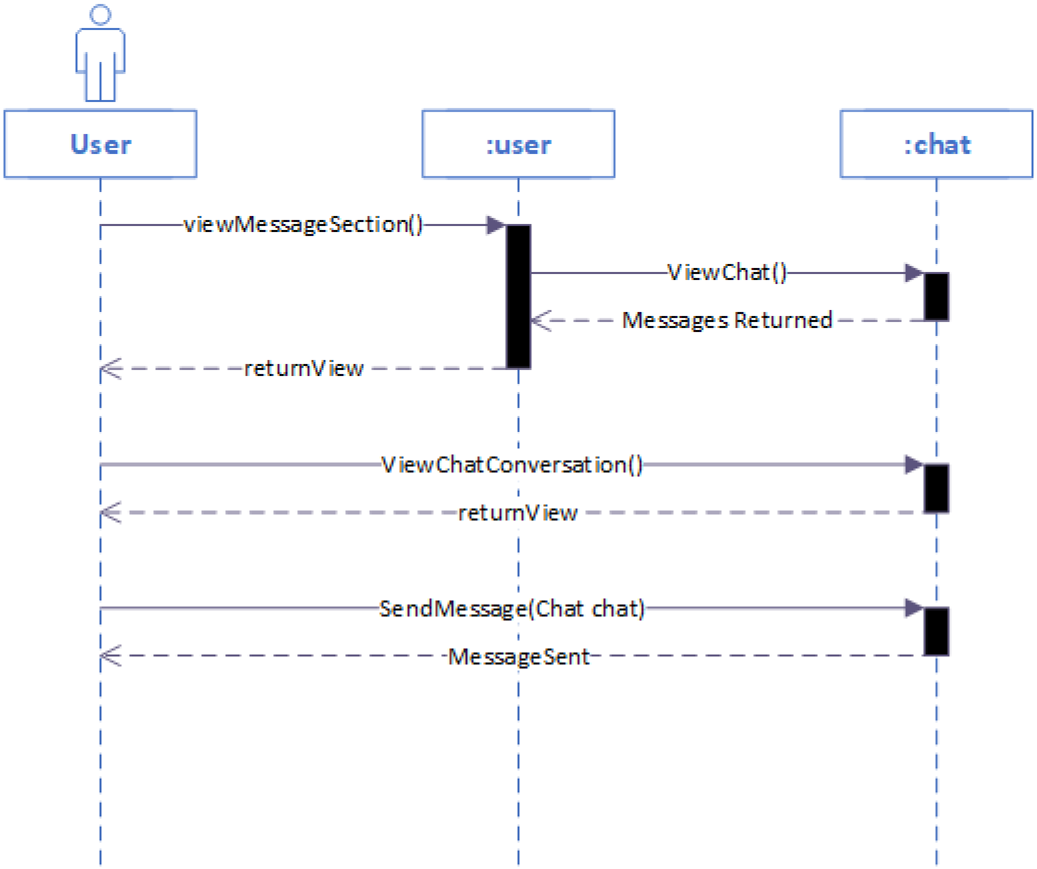


Figure 33: Chat Sequence Diagram

5. Data design

5.1 Data dictionary

JSON schemas of hath milao are represented in a diagram.

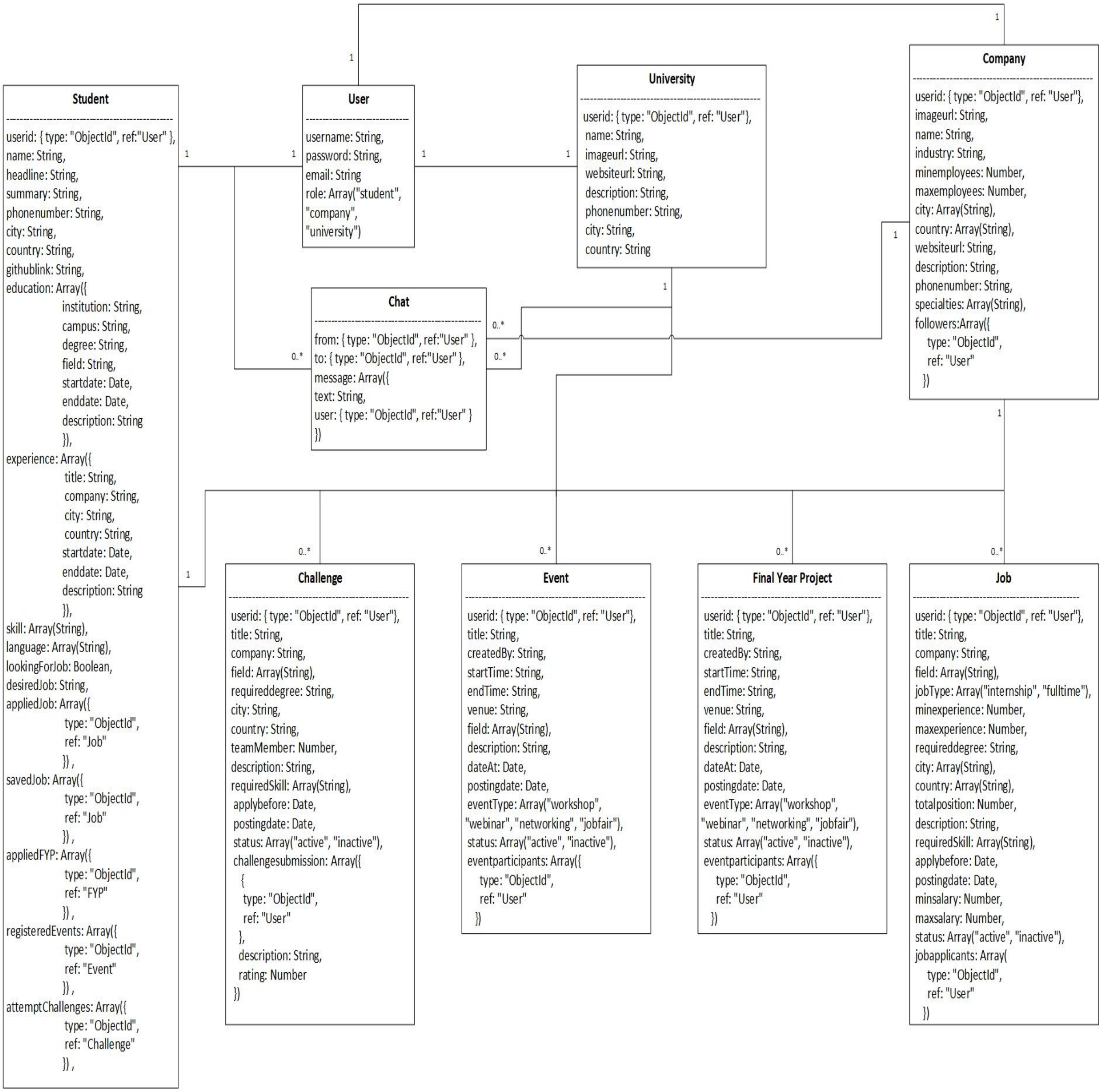


Figure 34: JSON Schema Diagram of Hath Milao

6. Algorithm & Implementation design

6.1 Content Based Recommendation Engine

Recommender systems are active information filtering systems which personalize the information coming to a user based on his interests, relevance of the information etc. We are using content based recommender system algorithms that recommend job to students and alumni's on the basis of their skill, education, etc. We also make a recommendation engine for a company that recommend student and alumni profile. Concept used in this algorithms.

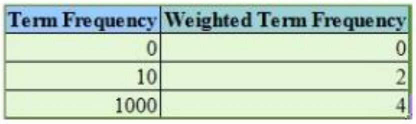
6.1.1 Term Frequency (TF) and Inverse Document Frequency (IDF)

TF is simply the frequency of a word in a document. IDF is the inverse of the document frequency among the whole corpus of documents. TF-IDF is used mainly because of two reasons: Suppose we search for **"the rise of analytics"** on Google. It is certain that **"the"** will occur more frequently than **"analytics"** but the relative importance of analytics is higher than the search query point of view. In such cases, TF-IDF weighting negates the effect of high frequency words in determining the importance of an item (document).

But while calculating TF-IDF, log is used to dampen the effect of high frequency words. For example: TF = 3 vs TF = 4 is vastly different from TF = 10 vs TF = 1000. In other words the relevance of a word in a document cannot be measured as a simple raw count and hence the equation below:

__**Equation**__

$$w_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{tf}_{t,d}, & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases}$$



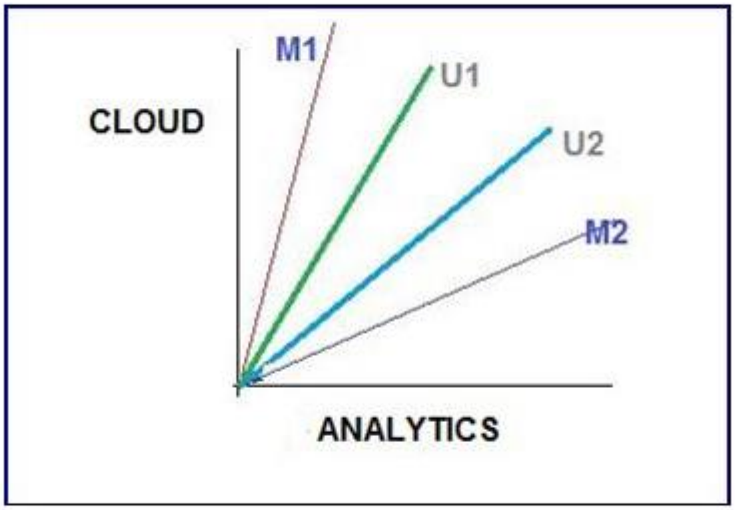
Term Frequency	Weighted Term Frequency
0	0
10	2
1000	4

It can be seen that the effect of high frequency words is dampened and these values are more comparable to each other as opposed to the original raw term frequency.

After calculating TF-IDF scores, how do we determine which items are closer to each other, rather closer to the user profile? This is accomplished using the Vector Space Model which computes the proximity based on the angle between the vectors.

6.1.2 Vector Space Model

In this model, each item is stored as a vector of its attributes (which are also vectors) in an n-dimensional space and the angles between the vectors are calculated to determine the similarity between the vectors. Next, the user profile vectors are also created based on his actions on previous attributes of items and the similarity between an item and a user is also determined in a similar way.



Shown above is a 2-D representation of a two attributes, Cloud & Analytics. M1 & M2 are documents. U1 & U2 are users. The document M2 is more about Analytics than cloud whereas M1 is more about cloud than Analytics.

User U1, likes articles on the topic ‘cloud’ more than the ones on ‘analytics’ and vice-versa for user U2. The method of calculating the user’s likes / dislikes / measures is calculated by taking the cosine of the angle between the user profile vector (Ui) and the document vector.

The ultimate reason behind using cosine is that the value of cosine will increase with decreasing value of the angle between which signifies more similarity. The vectors are length normalized after which they become vectors of length 1 and then the cosine calculation is simply the sum-product of vectors.

7. Software requirements traceability matrix

Following is the traceability matrix of hath milao.

Table 1 Requirements Traceability Matrix

Req. Number	Ref. Item	Design Component	Component Items
FR-1	Class Diagram	User	Login( )
FR-2	Class Diagram	User	Signup( )
FR-3	Class Diagram	User	VerifyEmail()
FR-4	Class Diagram	User	VerifyPassword()
FR-5	Class Diagram	User	CreateUserProfile ( )
FR-6	Class Diagram	User	updateProfile( )
FR-7	Class Diagram	User	checkLoginStatus( )
FR-8	Class Diagram	Student	followCompany( )

FR-9	Class Diagram	Student	UnfollowCompany( )
FR-10	Class Diagram	User	viewNotification( )
FR-11	Class Diagram	Job	SendNotification()
FR-12	Class Diagram	Education	AddEducation( )
FR-13	Class Diagram	Education	UpdateEducation( )
FR-14	Class Diagram	Education	UpdateEducation( )
FR-15	Class Diagram	Experience	AddExperience( )
FR-16	Class Diagram	Experience	UpdateExperience( )
FR-17	Class Diagram	Experience	UpdateExperience( )
FR-18	Class Diagram	User	UploadImage()
FR-19	Class Diagram	User	UpdateImage()
FR-20	Class Diagram	User	VerifyImageSize()
FR-21	Class Diagram	Job	AddJob( )
FR-22	Class Diagram	Job	viewAllJobs( )
FR-23	Class Diagram	Job	UpdateJob( )
FR-24	Class Diagram	Job	SearchJob( )
FR-25	Class Diagram	Job	DeleteJob( )
FR-26	Class Diagram	Job	ApplyJob( )
FR-27	Class Diagram	Job	CancelAppliedJob( )
FR-28	Class Diagram	Job	viewJobApplicants( )
FR-29	Class Diagram	Job	viewAppliedJobs( )
FR-30	Class Diagram	Job	SaveJob( )
FR-31	Class Diagram	Job	UnsaveJob( )
FR-32	Class Diagram	Job	viewSavedJobs( )
FR-33	Class Diagram	FinalYearProject	AddFYP( )
FR-34	Class Diagram	FinalYearProject	viewAllFYP( )
FR-35	Class Diagram	FinalYearProject	DeleteFYP( )
FR-36	Class Diagram	FinalYearProject	UpdateFYP( )
FR-37	Class Diagram	FinalYearProject	ApplyFYP( )
FR-38	Class Diagram	FinalYearProject	CancelApplyFYP( )
FR-39	Class Diagram	FinalYearProject	viewAppliedStudents( )
FR-40	Class Diagram	FinalYearProject	viewAppliedFYP( )
FR-41	Class Diagram	Event	AddEvent( )
FR-42	Class Diagram	Event	ViewEvents( )

FR-43	Class Diagram	Event	DeleteEvent( )
FR-44	Class Diagram	Event	UpdateEvent( )
FR-45	Class Diagram	Event	RegisterEvent( )
FR-46	Class Diagram	Event	CancelRegisterEvent( )
FR-47	Class Diagram	Event	ViewEventParticipants( )
FR-48	Class Diagram	Event	ViewRegisteredEvents( )
FR-49	Class Diagram	Event	ViewJobFairEvents( )
FR-50	Class Diagram	Challenge	AddChallenge( )
FR-51	Class Diagram	Challenge	ViewAllChallenge( )
FR-52	Class Diagram	Challenge	DeleteChallenge( )
FR-53	Class Diagram	Challenge	UpdateChallenge( )
FR-54	Class Diagram	Challenge	AttemptChallege( )
FR-55	Class Diagram	Challenge	CancelAttemptChallenge( )
FR-56	Class Diagram	Challenge	ShowSubmitChallenge()
FR-57	Class Diagram	Challenge	viewAttemptChallenge( )
FR-58	Class Diagram	Challenge	submitChallengeSolution()
FR-59	Class Diagram	Challenge	GiveRating( )
FR-60	Class Diagram	Challenge	ShowRating( )
FR-61	Class Diagram	Chat	SendMessage( )
FR-62	Class Diagram	Chat	RecieveMessage( )
FR-63	Class Diagram	Chat	ViewMessage( )
FR-64	Class Diagram	Chat	ViewChatConversation( )
FR-65	Class Diagram	Statistics	ViewStatisticsToStudent( )
FR-66	Class Diagram	Statistics	ViewStatisticsToCompany ( )
FR-67	Class Diagram	Statistics	ViewStatisticsToUniversity( )
FR-68	Class Diagram	Recommendations	ViewRecommendedJobs( )
FR-69	Class Diagram	Recommendations	ViewRecommendedStudents( )
FR-70	Class Diagram	University	ShowJoblessStudents()
FR-71	Class Diagram	User	Logout()
FR-72	Class Diagram	User	forgotPassword()

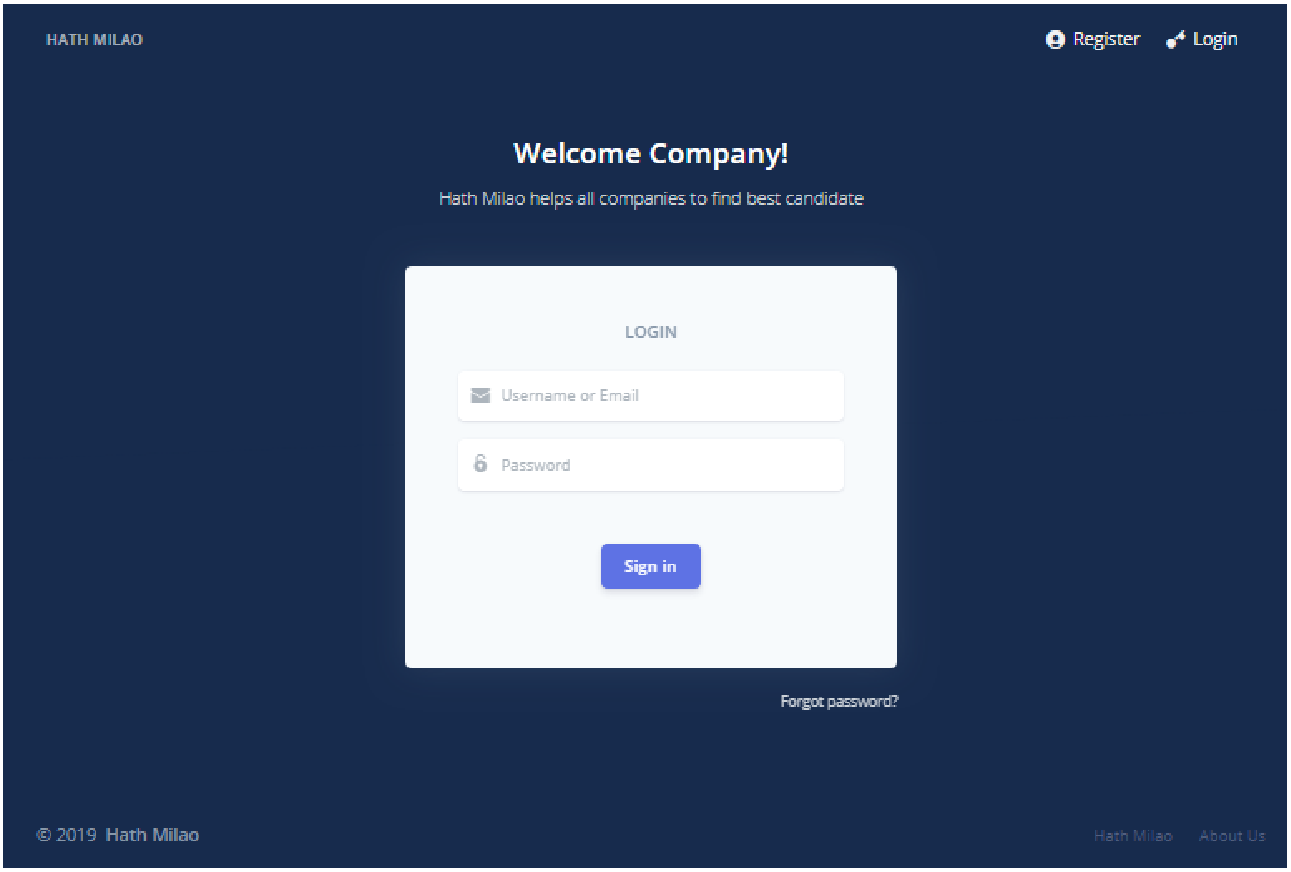
8. Human interface design

User enter URL of Hath Milao in the browser, Landing page of website is open. Different option on screen is shown such as company, student and university. User click on option according to their role after clicking they will redirect on particular selected option login page. If user is registered into system they directly enter their username and password. System will validate it and direct into portal home page. If user is not registered, first they have to register into system to use the features of system.

8.1 Screen Images for Student

8.1.1 Login Screen for Student

This is login screen which is used by user to login into system by entering the username and password.



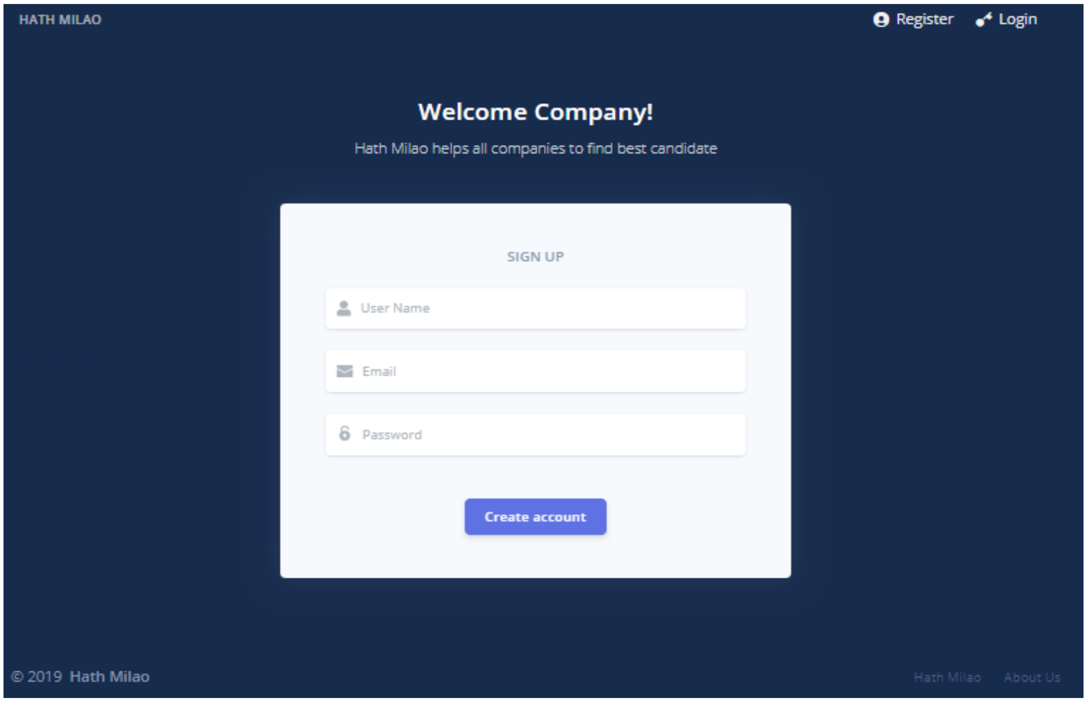
Welcome Company!

Hath Milao helps all companies to find best candidate

Figure 35: Login Screen for Company

8.1.2 Signup Screen for Company

This is signup screen used by company to register into system by entering the username, email and password.



Welcome Company!

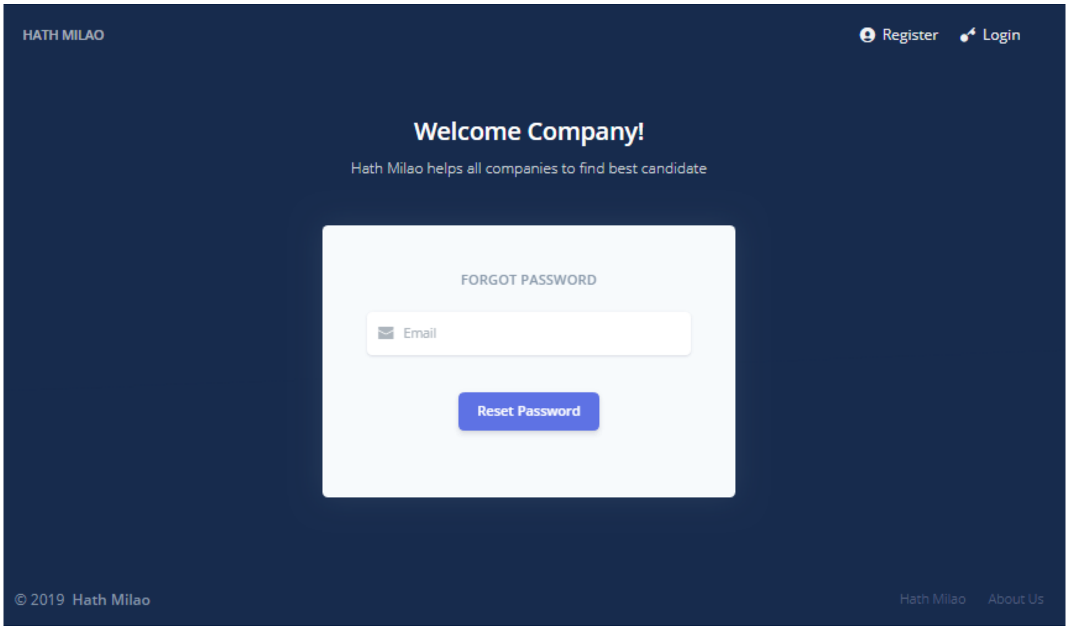
Hath Milao helps all companies to find best candidate

© 2019 Hath Milao

Figure 36: Signup Screen for Company

8.1.3 Forgot Password Screen for Company

This is forgot password screen used by company to forgot their password by simply submitting email which company used at time of registration.



Welcome Company!

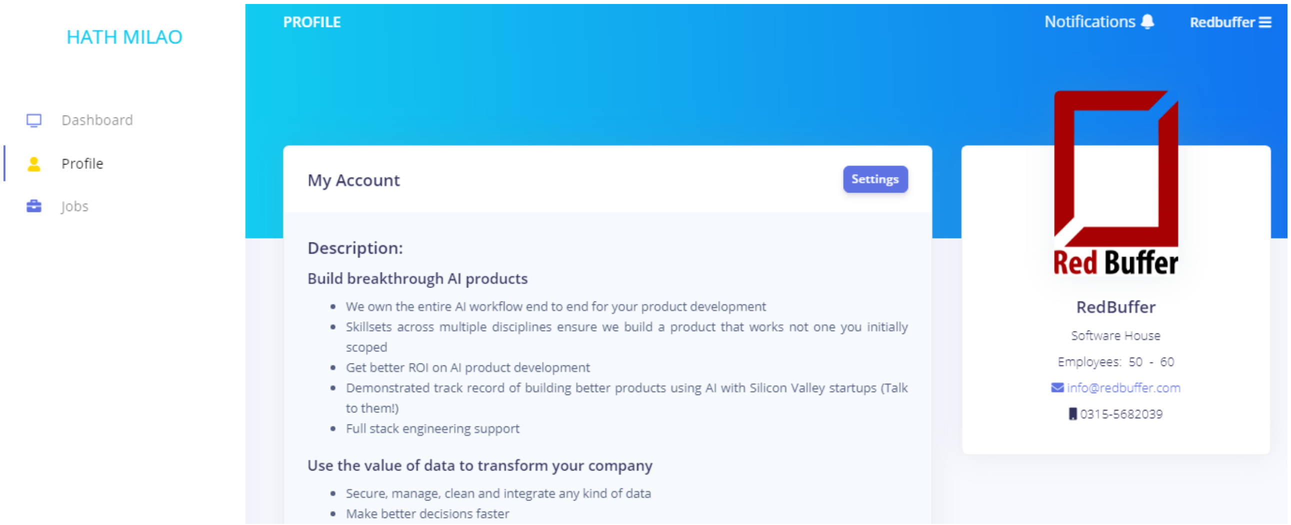
Hath Milao helps all companies to find best candidate

© 2019 Hath Milao

Figure 37: Forgot Password Screen for Company

8.1.4 Profile Screen for Company

This is profile screen for company in which all information related to profile is shown to company.



Description:

Build breakthrough AI products

- We own the entire AI workflow end to end for your product development
- Skillsets across multiple disciplines ensure we build a product that works not one you initially scoped
- Get better ROI on AI product development
- Demonstrated track record of building better products using AI with Silicon Valley startups (Talk to them!)
- Full stack engineering support

Use the value of data to transform your company

- Secure, manage, clean and integrate any kind of data
- Make better decisions faster

RedBuffer

Software House

Employees: 50 - 60

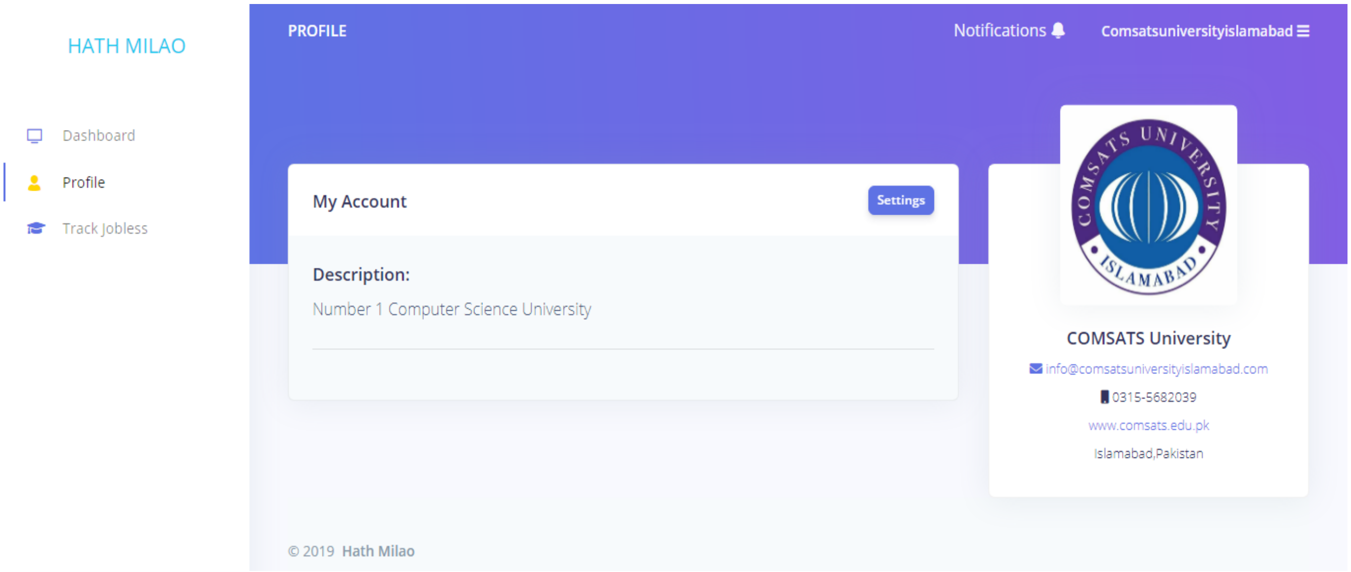
info@redbuffer.com

0315-5682039

Figure 38: Profile Screen for Company

8.1.5 Profile Screen for University

This is profile screen for university in which all information related to profile is shown to university.



Description:

Number 1 Computer Science University

COMSATS University

info@comsatsuniversityislamabad.com

0315-5682039

www.comsats.edu.pk

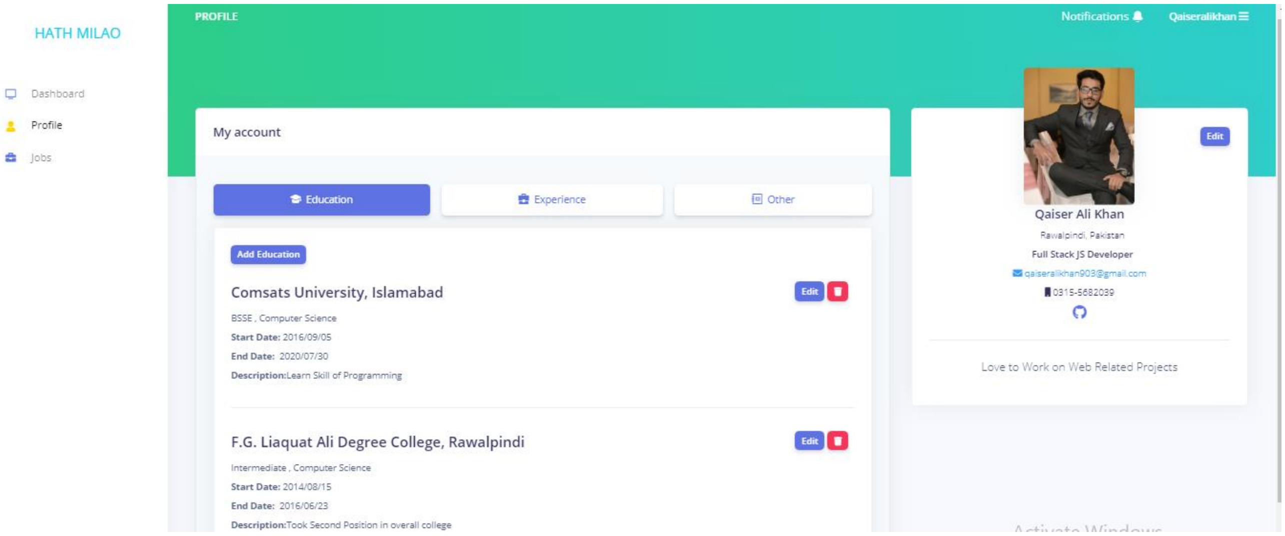
Islamabad,Pakistan

© 2019 Hath Milao

Figure 39: Profile Screen for University

8.1.6 Profile Screen for Student

This is profile screen for student in which all information related to profile is shown to student.



Comsats University, Islamabad

BSSE , Computer Science

Start Date: 2016/09/05

End Date: 2020/07/30

Description: Learn Skill of Programming

F.G. Liaquat Ali Degree College, Rawalpindi

Intermediate , Computer Science

Start Date: 2014/08/15

End Date: 2016/06/23

Description: Took Second Position in overall college

Qaiser Ali Khan

Rawalpindi, Pakistan

Full Stack JS Developer

qaiseralikhan903@gmail.com

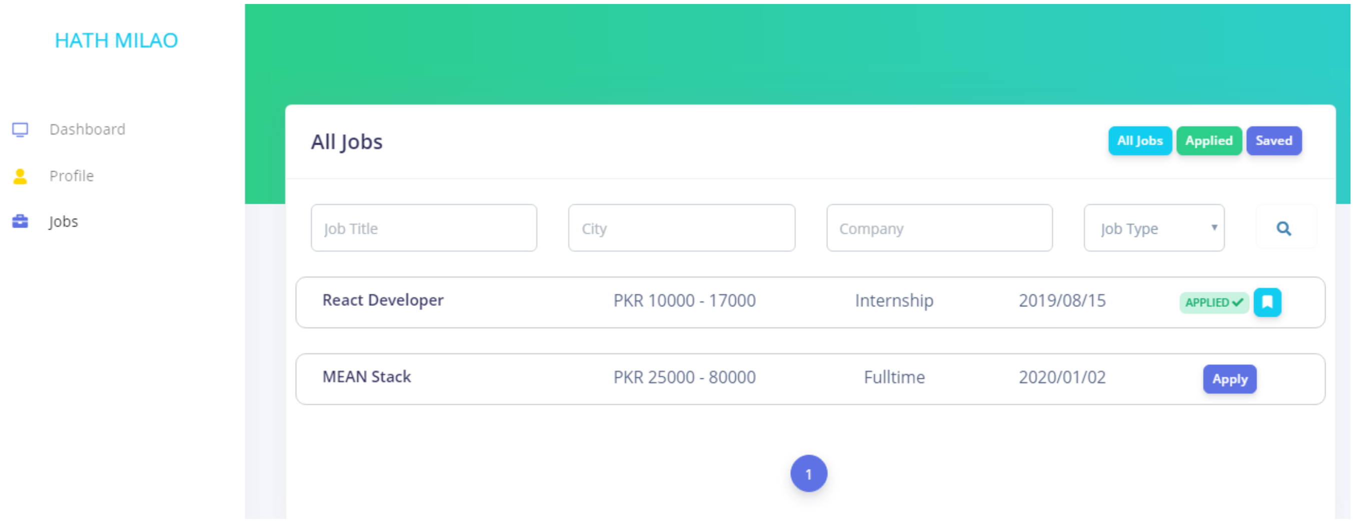
0315-5682039

Love to Work on Web Related Projects

Figure 40: Profile Screen for Student

8.1.7 Jobs Screen for Student

This is job screen for students in which all jobs shown to student. Student have different filters to search jobs and internships. Student have option to directly apply for job and have option to save a job.

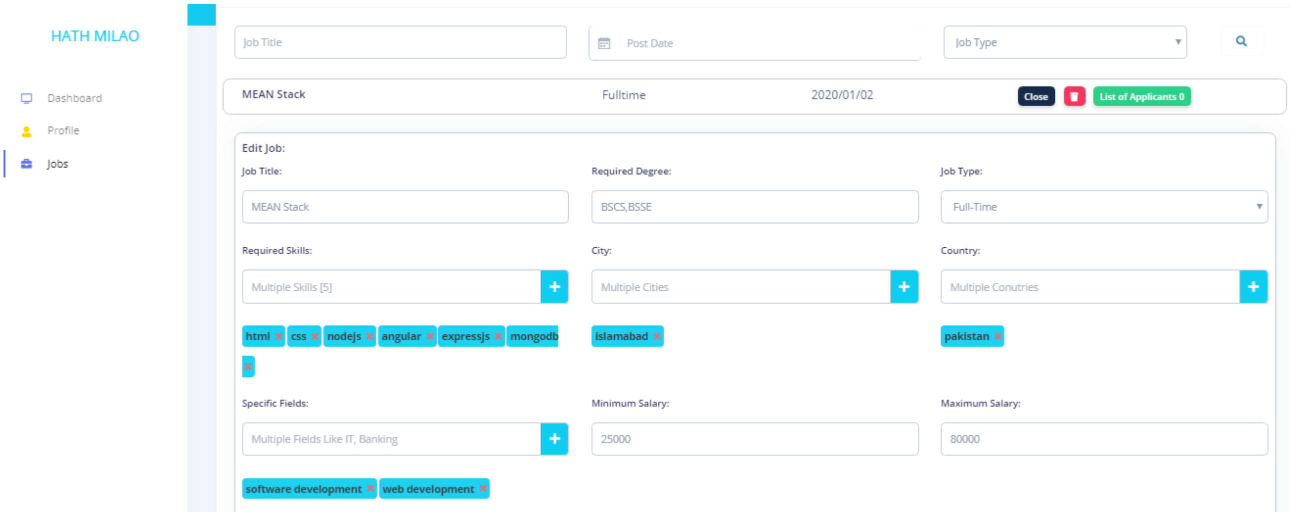


React Developer	PKR 10000 - 17000	Internship	2019/08/15	APPLIED ✓
MEAN Stack	PKR 25000 - 80000	Fulltime	2020/01/02	Apply

Figure 41: Jobs Screen for Student

8.1.8 Edit Job Screen for Company

This is edit job screen in which company easily edit or update job and internship.



MEAN Stack Fulltime 2020/01/02

Edit Job:

Job Title: MEAN Stack

Required Degree: BSCS,BSSE

Job Type: Full-Time

Required Skills: html ✕ css ✕ nodejs ✕ angular ✕ expressjs ✕ mongodb ✕

City: islamabad ✕

Country: pakistan ✕

Specific Fields: software development ✕ web development ✕

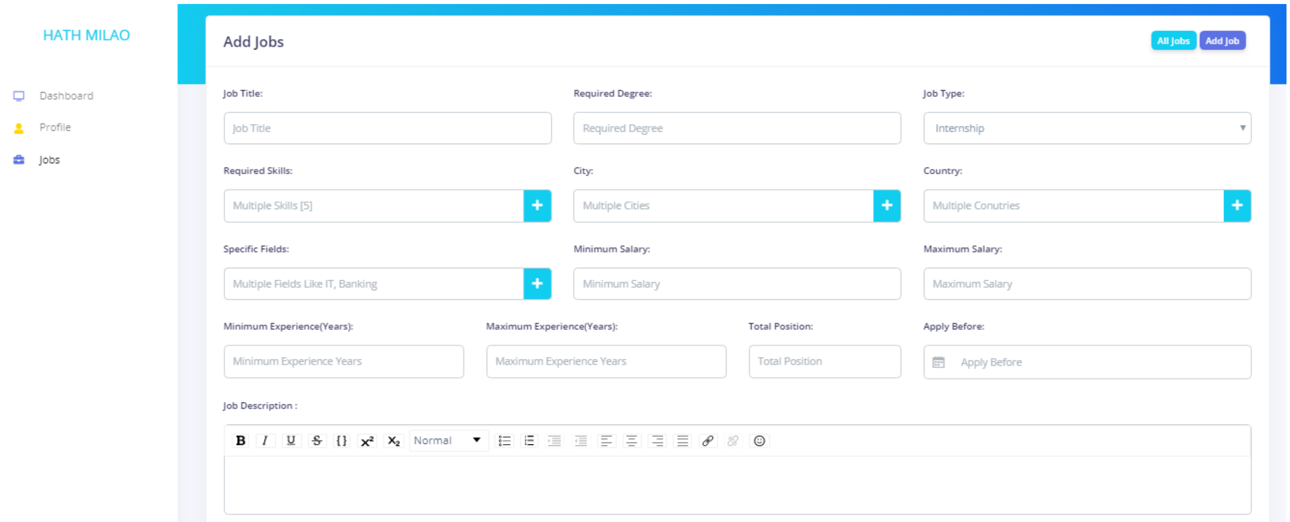
Minimum Salary: 25000

Maximum Salary: 80000

Figure 42: Edit Job Screen for Company

8.1.9 Add Job Screen for Company

This is add job screen for company to add new job and internship into system.



Add Jobs

Job Title:

Required Degree:

Job Type: Internship

Required Skills:

City:

Country:

Specific Fields:

Minimum Salary:

Maximum Salary:

Minimum Experience(Years):

Maximum Experience(Years):

Total Position:

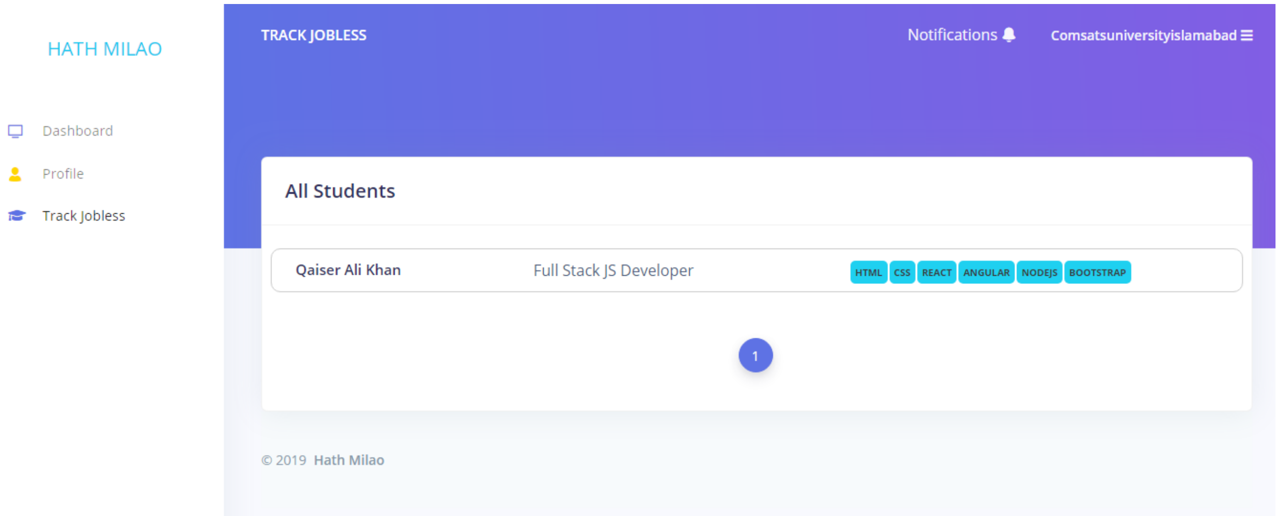
Apply Before:

Job Description :

Figure 43: Add Job Screen for Company

8.1.10 Track Jobless Student for University

This is screen in which all jobless student show to university with their profile information



All Students

Qaiser Ali Khan Full Stack JS Developer HTML CSS REACT ANGULAR NODEJS BOOTSTRAP

© 2019 Hath Milao

Figure 44: Track Jobless Student for University

9. Appendix I

- How to design using UML (OOP): For guidance please follow the instructions mentioned in the link: http://agilemodeling.com/artifacts/

- How and when to design ER diagrams: For guidance please follow the instructions mentioned in the link: http://people.inf.elte.hu/nikovits/DB2/Ullman_The_Complete_Book.pdf

- Data flow diagrams: For guidance please follow the instructions mentioned in the link and book:
  - http://www.agilemodeling.com/artifacts/dataFlowDiagram.htm
  - Software Engineering –A Practitioner’s approach by Roger Pressman
- Architecture diagram: For guidance please follow the instructions mentioned in the link and book:
  - Ian Sommerville – Software Engineering 9th Edition– Chapter 6

- Recommendation Engine Algorithms link that we follow to explain algorithms and implementation section.
  - https://www.analyticsvidhya.com/blog/2015/08/beginners-guide-learn-content-based-recommender-systems/
  - https://medium.com/@armandj.olivares/building-nlp-content-based-recommender-systems-b104a709c042