Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 1, pp: (241-254), Month: January-April 2016, Available at: www.noveltyjournals.com

A Study on Online Training and Placement System Open Source Tools

[1]A. Naga Sahithi, [2]P. Siva Sowmya, [3]Sravani, [4]KorraManasa

[1,2,3,4] SVEC, Tirupathi, India

***Abstract*: The main objective of "Online Application for the Training and Placement" portal is to automate the manual activity involved in Training and Placement programmer. Rich GUI of the system makes the placement activities in the Training and Placement Programmer Department of the college more interactive. The users of the system typically are TPO - Training & Placement Officer, Students, Recruiter, Visitor and Admin. TPO of the college can manage the details of all students on his console. He can send Email updates to the students about the placement activities. The System also provides the personal and academic details of a student to the recruiting company on demand. The students should be able to upload the Curriculum Vitae (CV) by logging into the system. Also, they are provided with the feature of editing some fields like Email ID, mobile number and address. Recruiter/Company representatives may also be able to access/search any information uploaded by the Students. The Administrator of portal should be able to add/modify/delete the student/company information which in turn gets stored or modified in a relational database. The system is an internet/intranet based online application that can be accessed throughout the organization and also outside with the proper login credentials are provided. In SDLC many open source tools are used to effectively develop this application. As part of it, MS Visio2010-at Requirement phase, Enterprise Architect and Agro-UML- at Design phase, Eclipse Helios(ER-Master), PREST and CIDE- at Implementation phase, Graph Walker- a Model-based testing (MBT) will be exploited in the Testing phase to automate the testing procedures of various modules.**

***Keywords:* OTAP, Defects, Feature Identification.**

I. INTRODUCTION

MICROSOFT VISIO 2010is a diagramming application and is a part of the Microsoft office family. The product was first presented by the SHAPEWARE Corporation in 1992. It was acquired by Microsoft in 2000.This tool read and write drawings in VSD or VDX file format. All the versions of Visio used VSD, the proprietary binary-file format. VDX is a well-documented XML schema-based format.

The advanced diagramming tools of MICROSOFT VISIO 2010 help to simplify complexity with dynamic, data-driven visual and new ways to share on the web in real-time. Depending on the user requirement we can create an organization charts, network diagrams, or a business processes, with the help of new tools and more intuitive interface in Visio 2010 make it easier to bring diagrams to life. It is used to draw diagrams with templates which are in-built in the tool and modern, pre-drawn shapes and can also easily link diagram to popular data source (such as Microsoft office excel).

Enterprise Architect is a graphical platform for designing and constructing software system, for business process modeling, and for more generalized modeling purpose. Enterprise architect is based on the latest UML 2.4 specification. Enterprise architect is an advanced tool that covers all aspects of the development cycle, providing full traceability from initial design phase through to development, maintenance, testing, and change control.

Eclipse is an IDE. It contains a workspace to work and an extensible plug-in system for adapting to the environment. Eclipse is written generally in java and its primary use is for developing java applications, but it may also be used to develop application in other programming languages through the use of plug-ins including: Ada, ABAP, C, C++, Cabal,

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 1, pp: (241-254), Month: January-April 2016, Available at: www.noveltyjournals.com

Fortran, Haskell, JavaScript, Lasso, Natural, Perl, Python, R, etc., It can also be used to develop packages software mathematical. Eclipse is an open source project of eclipse foundation[1]. Eclipse runs on multiple platform including windows, Linux, Mac OS. There may be minor differences between eclipse versions for different platforms and operating system, but the core features work the same way. This document describes the 2010 version of eclipse known as Helios version 3.6 releases for java developers.

ER-masters is a GUI editor for ER-diagrams. It runs on eclipse plug-in. it can be done graphically to draw ER-diagram, exporting to DDL from ER-diagram, etc. Moreover, importing from DB, management of the group, and the historical management, etc. is supported. ER-master is being developed with eclipse Helios now. Corresponding database need are: Microsoft access, database2, H2, HSQLDB, Mysql, Oracle, PostgreSQL, SQLite, Microsoft SQL Server.

ArgoUML[2] is a UML(Unified modeling Language)diagramming application written in java and released under the open source eclipse public License(EPL) 1.0. By virtue of being a java application, it is available on any platform supported by java.ArgoUML 0.30.2, tested on windows XP. ArgoUML main purpose is to draw Use case, Sequence, Collaboration, and other UML diagrams.

CIDE is originally called as colored integrated development environment is a software product line tool for software product line development. Developers do not physically extract the feature code, but just annotate code fragments inside the given code and use tool support for views and navigation for software product line development. For annotation, background colors are used. CIDE supports a number of different languages. Java is special, because there is a JDT version which uses the eclipse infrastructure. All other languages (including another java plug-in) are generated from the language's grammar. CIDE is an eclipse plug-in. To install you need Eclipse 3.5 (Galileo) or Eclipse 3.6 (Helios) running with Java version 1.6 (other versions may work but have not been tested).

Graphwalker Describing a normal workflow when designing for test automation using Graphwalker. There are 3 main steps involved. (1) Test idea, (2) The design, (3) Creating the test code.

Test idea is to write a regression test for the specific Desktop Client, more specifically, the feature logins. The feature is supposed to work like this i.e. if the client is started; the login dialog is expected to be displayed. The users enter valid credentials and the client is expected to start. If the users quits, or logs out, the login dialog is displayed once again. If the user checks there member me checkbox, and logs in; the client starts and next time the user starts the client, it will start without asking the client for credentials.

Prest Extracting common static code metrics such as C,C++, Java, JSP and PL/SQL languages. Presenting output via GUL components and in*.XML,*.CSV, *.XSL and *.arff file formats. Generating call graphs in class and method level. Defining new metrics or thresholds on extracted metrics. Applying machine learning methods for analysis and defect prediction. Prest works as an executable jar and does not actually required any particular step up process. However, you need to select a workspace in which project folders, parse results and call graph information will be saved. By using prest tool we can find defective were that may highlighted in green and red color. Green indicates non-defective indicates and red indicated defectives. Whenever the input is given then in output will convert into .CSV/.arff file and output will display when prediction is run by using 10flod cross validation/normalized data methods.

II. LITERATURE SURVEY

We refer a total of fifty papers among which following paper are taken for literature review.

Table.1. List of publications

S.no	Name of the journals
1.	IEEE Computer society
2.	ACM
3.	Science Direct
4.	IJRET

[1]http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/heliossr2
[2]http://argouml.tigris.org/

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 1, pp: (241-254), Month: January-April 2016, Available at: www.noveltyjournals.com

S.No	Title	Author and Year	Objective	Issue	Solution	Future work
1	FLAT3: Feature Location and Textual Tracing Tool.	Trevor Savage, Meghan Revelle, Denys Poshyvanyk, 2010.	Feature location is the process of finding the source code that implements a Functional requirement of a software system. It an important role in software maintenance Activities, but when it is performed manually, it can be challenging and time consuming, Especially for large, long-lived systems.	Feature location is the process of finding the source code that Implements a functional requirement of a software system. It plays an important role in software maintenance activities, but when it is performed manually, it can be challenging and time consuming, especially for large, long-lived systems	To reduce the amount of time and effort spent searching for a feature's implementation. Some of these approaches employ information retrieval to search body of text, such as source code, for sections that are relevant and dynamic analysis.	Future work on FLAT3 includes making it more robust to be able to index large source code bases, trace larger programs, and save and update annotations for evolving programs. A user study to evaluate the tool's usability is also planned.
2	A Semi-Automatic Approach for Extracting Software Product Lines.	Marco Tuilo Valente, Virgilio Borges, Leonardo Passos. 2012.	This paper describes a semi-automatic approach to annotate the code of optional features in SPLs. The proposed approach is based on an existing tool for product line development, called CIDE,	The extraction of nontrivial software product lines from a legacy application is a time-consuming task. First, Developers must identify the components responsible for the implementation of each program feature. Next, they must locate the lines of code that reference the components discovered in the previous step. Finally, they must extract those lines to independent modules or annotate them in some way.	The proposed solution is the first algorithm for feature extraction that relies on an annotative approach.	It is also not recommended for features coupled to external resources, such as databases. Furthermore, the current Implementation of the proposed annotation algorithm does not handle features associated with the following Java elements: exceptions, enums, and generic types. We have plans to support such elements in further versions.
3	Feature Location in Source Code: A Taxonomy and Survey	BogdanDit, Meghan Revelle, MalcomGethers, Denys Poshyvanyk. 2012.	No maintenance activity can be completed without first locating the code that is relevant to the task at hand, making feature location essential to software maintenance since it is performed in the context of incremental change.	This survey has also shed light on open issues in feature location, such as the need for comparisons and benchmarks.	IR for feature location, but their solution incorporates non-source code artifacts, such as bug reports, mailing lists, external documentation,	By structuring the research area of feature location, this taxonomy and survey contribute well-defined organization to the field and should aid in resolving some of the open issues.

4	A Systematic Survey of Program Comprehension Through Dynamic Analysis.	Bas Cornelissen, Andy Zaidman, Arie Van Deursen, Leon Moonen, Rainer Koschke. 2009.	The study of a program's execution, known as dynamic analysis, has become a common technique in this respect and has received substantial attention from the research community, particularly over the last decade.	The phenomenon in which software acts differently when under observation, might pose a problem in cases where timing issues play a role.	Comparisons refer to evaluation types in which an approach is compared to existing solutions side by side; the involvement of human subjects measures the impact of an approach from a cognitive point of view; and quantitative evaluations are aimed at the assessment of various quantifiable aspects of an approach	We discuss the implications of our findings and provide recommendations for future work. 1.generic attributes can we identify to characterize 2. The work on program comprehension through dynamic analysis. the attention for each of these attributes 3.the main activities typically evaluated distributed across the relevant literature
5	Variability-Aware Parsing in the Presence of Lexical Macros and Conditional Compilation.	Christian Kastner, Paolo G. Giarrusso, TillmannRendel, Klaus Ostermann. 2011.	Lexical preprocessors are used to manage different variants of the project and to define compile-time code transformations. Unfortunately, while being a simple way to implement Variability, conditional compilation and lexical macros hinder automatic analysis, even though such analysis is urgently needed to combat variability-induced complexity. To analyze code with its variability, we need to parse it without preprocessing it.	The parser issue by supporting only disciplined annotations in a controlled environment. With the parsing issue solved, variability-aware type checking for C is in reach.	We manually prepared 13 lines of Linux code to work around these bugs, but we are currently working on a solution. In that regard, the lexer's implementation is not entirely complete, but this is an implementation limitation, not a conceptual one.	The process is embarrassingly parallel, that is, trivial to parallelize, since every file can be parsed in isolation from the others we will need to consider dependencies between files only for type checks, or more accurately linker checks, in future work.
6	A Choice Relation Framework for Supporting Category-Partition test.	T.Y.Chen, P.L.Poon, T.H.Tse, 2003	A choice relation framework used for capture the constraints among various values of the parameters, the environment conditions identified from the specification.	A choice relation framework for capture the constraints among various values of the parameters and the environment conditions identified from the specification.	A algorithms have developed for generating test frames. These test frames can used as the basis for generating test cases.	To Improve the effectiveness and efficiency of complete test further frameworks are construction by means of consistency checks and automatic deductions of relations.
7	The AGEDIS Tool for Model Based Testing.	A. Hartman and K. Nagin, 2004.	Describes the tools and interface created by the AGEDIS, it includes an integrated environment for modeling, test generation.	Describe the tools and interfaces created by the AGEDIS, It includes an integrated environment for modelling.	Test generation, test execution, and other test related activities. These tools support a model based testing methodology features for a	New and open source tool are presented to improve the integrated environment for modelling.

					large degree of automation.	
8	Using UML Collaboration diagrams for static checking and test generation.	A.Abdurazik and J.Offutt, 2000.	UML diagrams represent a opportunity for testing, it present a criteria based on UML collaboration diagrams.	UML diagrams represent a opportunity for testing because they describe the function	Presenting a criteria based on UML collaboration diagrams. These criteria allows a formal integration tests based on high level design notations.	Three major directions for the future of this work currently carrying out an empirical evaluation of this work , we plan to implement tools to perform the static analysis, we continue to develop analysis and testing techniques that are based on the various design.
9	Selection of Software Testing Technique: A Multi Criteria Decision Making.	M.Victor, and N.Uapdhyay, 2011.	Selection of testing technique at every stage of SLDC depends on many factors such as resources, cost, schedule, etc.	Selection of testing technique at every stage of SDLC depends on many factors such as resources, cost, schedule etc.	There are a number of testing techniques at various phases of testing.	Other testing technique at every stage of SLDC are to be introduced to verifying the resources, cost, schedule.
10	Model Based Testing in Practice at Microsoft.	Keith Stobie, 2005	Providing one approach being extensively investigated and used is model based testing. With a finite state machine modeling tool (TMT) several product groups are exploring the use of the abstract state machine language (ASML).	One approach being investigated and used is Model Based Testing by using a Finite State Machine modelling tool (TMT).	as a solution from an approaches have shown an increased ability to find defects earlier it including in the specification and design stages also.	By using Finite State Machine modelling tool number of approach are to be present.
11	Methodology for Validating Software Metrics.	N.F.Scheidewind, 1992.	The classification approach for software defect prediction categorizes the module which is represented by a set of software metrics/code attributes into fault prone and non-fault prone based on data of previous developed projects.	The faults are identified by using metrics and code attributes from classifiers and data sets.	By using classification models we can solve the problems of software development.	Validating the software using defect prones and non-defect prones
12	Benchmarking classification models for software defect prediction: A proposed framework and novel finding.	S.Lessmann, B.Baesens, C.Mues, And S.Pietsch, 2008.	In error prone software module the number of prediction models has been proposed ranging from mathematical function to machine learning models. The performance defect predication methods vary based on type of datasets and data pre-processing.	The main issues is comparing classifiers and one are more public data sets	By using classification algorithms the results or solution can find.	By using 10 fold cross validation and till some classification models are implemented in future
13	Consequences of variability in classifier performance estimates.	T.Reader, T.Hoens and N.Chawla, 2010.	In classifier technique dichotomous is dominate approach having two class defect prone and not defect prone software component such as classes or subsystem. The most	The machine learning community involves to perform several algorithms over a series of usually	There are many dimensions along which methodologies vary wildly.	Steady-state estimates smooth out these inconsistencies and best estimator. The results may occur in statistical test

			specific technique under these caterigoe are rule induction algorithm C4.5, Neural network, Random Forests, Bayesian approaches and regression which for binary classification. Also finding suitable classifiers which are transferable to new settings are context specific and hence it is necessary generalization is via learning technique rather than to specific classifiers.	unrelated data sets.		
14	Reliability and validity in comparative studies of software prediction models.	I.Myrtveit, E. Stensrud, and M.Shepperd. 2005.	A frequently used research procedure comprising three main ingredients: a single data sample, an accuracy indicator, and cross validation. Typically, these empirical studies compare a machine learning model with a regression model. The results suggest that it the research procedure itself that is unreliable.	The reason for reliability and validating the metrics is lack of coverage is poorly understand	By using machine learning method the metrics are finding.	To find defects in metrics we use regression models in machine learning methods.
15	A general software defects-proneness prediction framework.	Q. Song, Z. jia, M. Shepperd, S.Ying, 2011.	In general software defects prone module the data normalization is used to create the prediction to find defects and by the machine learning framework models the defects are identified and then the frameworks for datasets and classifiers forever instances.	The software defect predication is used to generate the framework by using previous data sets	The problem solved by using the machine learning frameworks.	Developing the framework using more machine learning methods

III. CASE STUDY

Case Study: Online Training and placement System:

The main aim of this case study is to collect information regarding students like their personal details, their aggregate marks, their skill set and the technical skills that are required in the resume to be sent to a company. It also maintains details about the visit of company, eligibility criteria regarding the company. To develop such application, should come up with requirements and should be freezed after gathering them. Requirements information can be specified by SRS (Software requirement and specification).

1. Requirement Details:

SRS for Online Training and Placement System:

SRS (Software Requirement Specification) is a basis for agreement between customer and developer. It is the description software system to be developed and also checks whether the functional and non-functional requirements are met that are implemented in the product. SRS fully describe about two aspects. They are

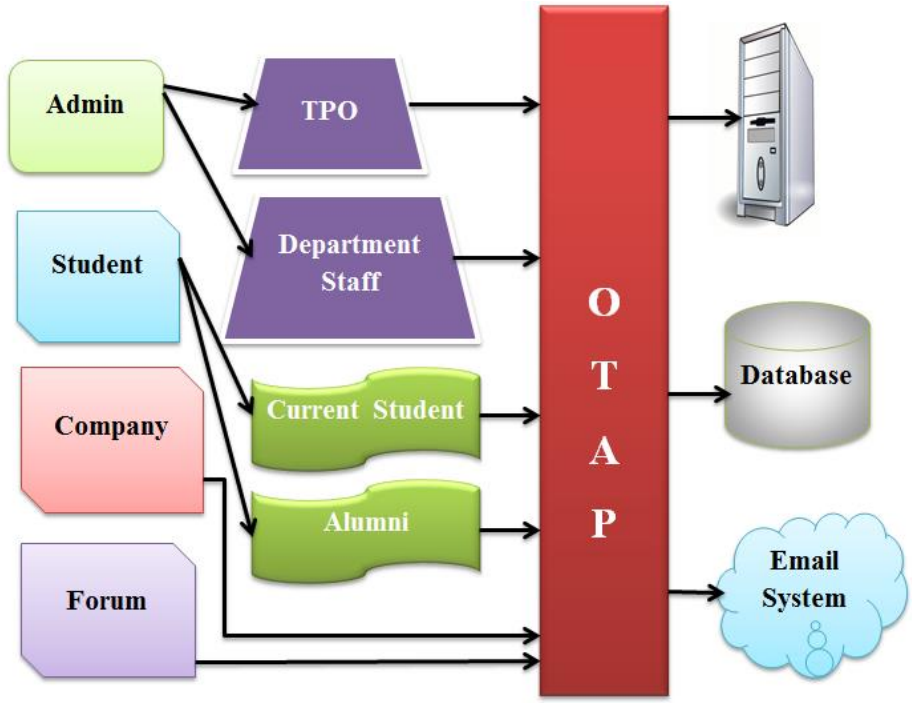
- What the software product need to do and
- What it is not expected to do

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 1, pp: (241-254), Month: January-April 2016, Available at: www.noveltyjournals.com

Table 2: SRS Template for Online Training and Placement System

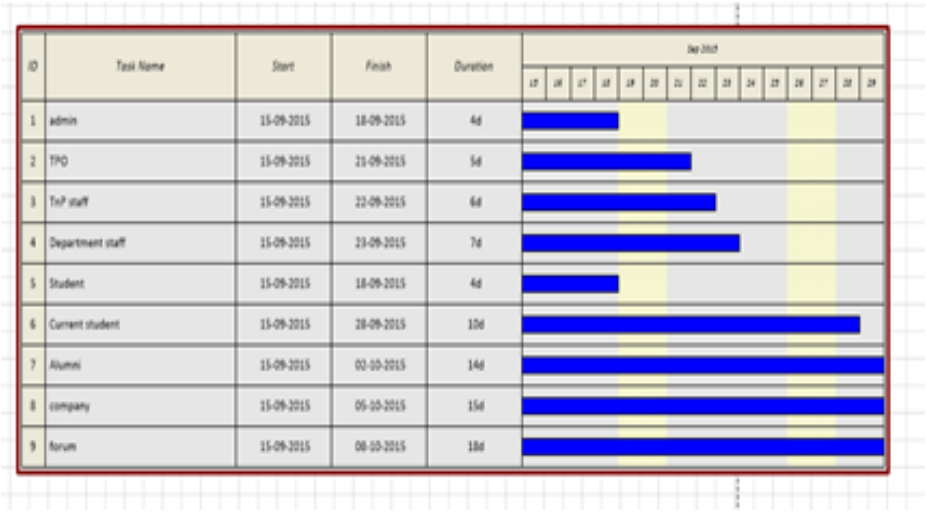
INTRODUCTION	This project aim is to develop an online application that can be accessed throughout the organization and outside as well to maintain and access students information
PURPOSE	The purpose of this project is to collect students data and resume and manage them based on the various streams and it also considers eligibility criteria to attend a company
SCOPE	Students should maintain the information and also have permissions to update their profile. Company must have permissions to access student's information. Notification regarding company need to be automatically sent to all those students who registered in this site
Goal	The main goal of this application is to guide students in various opportunities they are eligible for and also should provide the relationship with corporate companies for inviting them for a placement in the college
SYSTEM ENVIRONMENT	
FUNCTIONAL REQUIREMENTS	**Req-1:** Faculty Member should be able to login to the system **Req-2:** Student need to enter the roll number to access his information **Req-3:** Admin should login to the system to perform operations such as updating the details etc.
NON-FUNCTIONAL REQUIREMENTS	**Usability:** The developed project should be user friendly and interactive application to avoid confusion and also to save time **Performance:** The response of any page requested must be obtained in 5 second

After finalizing the requirements scheduling of the project is done by Microsoft Visio 2010 which is an open source tool. Gantt chart is designed by the above specified tool, displays which activity is to be performed based on the time scheduled. The following figure 1 shows the OTAP modules scheduling which tells about the start and end time to complete the module design plan. The following figure 2 shows the planning of OTAP modules.

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 1, pp: (241-254), Month: January-April 2016, Available at: www.noveltyjournals.com



ID	Task Name	Start	Finish	Duration
1	admin	15-09-2015	18-09-2015	4d
2	TPO	15-09-2015	21-09-2015	5d
3	TnP staff	15-09-2015	22-09-2015	6d
4	Department staff	15-09-2015	23-09-2015	7d
5	Student	15-09-2015	18-09-2015	4d
6	Current student	15-09-2015	28-09-2015	10d
7	Alumni	15-09-2015	02-10-2015	14d
8	company	15-09-2015	05-10-2015	15d
9	forum	15-09-2015	08-10-2015	18d

Fig 1.Gantt chart for OTP system

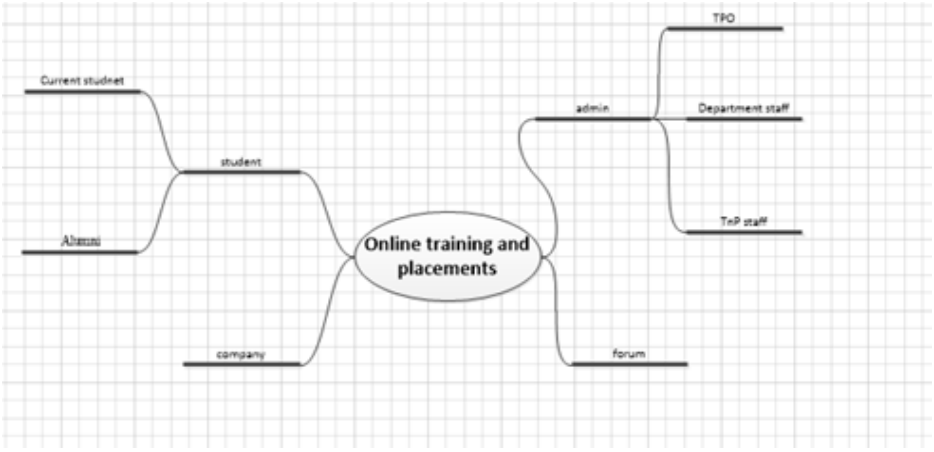


Fig 2. Brainstorm for OTP system

By designing brainstorming the team can capture and come to best plan based on the ideas and this helps to create the design effectively based on the concluded

2. Design Details:

Class diagram:

The class diagram is the building for Object oriented modeling which describes the structure of a system by its classes, attributes, operations and relation among the objects. At the Object Oriented Analysis we found following classes after few iterations. These classes are designed using an open source tool, namely Enterprise Architect-9.3

- ✓ Web Application
- ✓ Information Desk
- ✓ Form
- ✓ User
- ✓ Student
- ✓ Admin
- ✓ Current Student
- ✓ Alumni

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 1, pp: (241-254), Month: January-April 2016, Available at: www.noveltyjournals.com

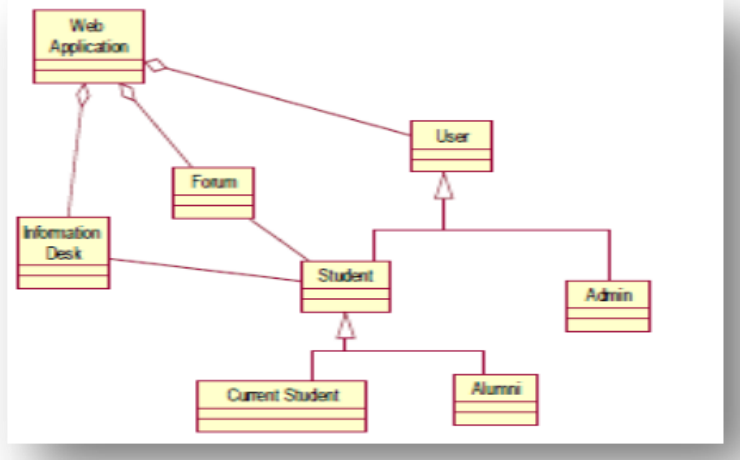


Fig 3. Class Diagram for Online Training and Placement System

Use case diagram:

Use case diagrams are used to elicitate requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified. Based on number of different scenarios, we have come across the following uses for the Online Training and Placement System are:

• Login

• Registration

• Update Student Detail

• Update Company Detail

• Search Student Details

• Update Staff Details

• Update Placed Student detail

• View Details

• Upload and Download Resume

The Use case Diagram for Online Training and Placement System is shown below and is implanted using open source tool called Agro-UML

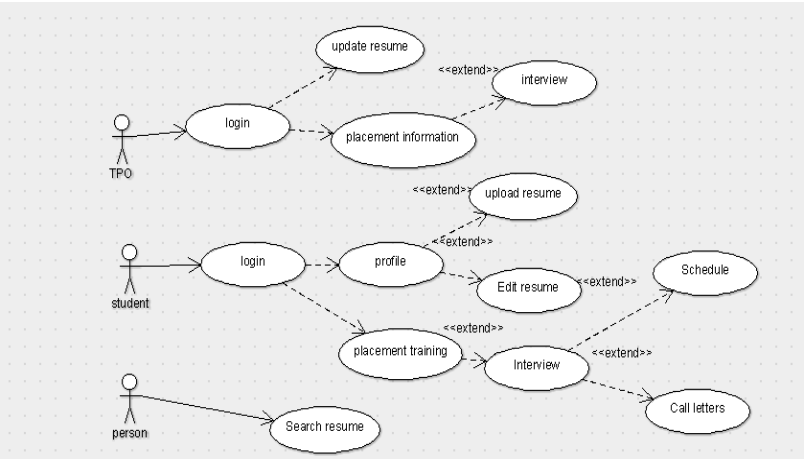


Fig 4. Use case for admin module in OTAP system

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 1, pp: (241-254), Month: January-April 2016, Available at: www.noveltyjournals.com

Sequence diagram:

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. For a Scenario Successful login the sequence of actions are shown with the sequence diagram and can be implemented based on Agro-UML tool which describes sequence of steps to be done for a successful login

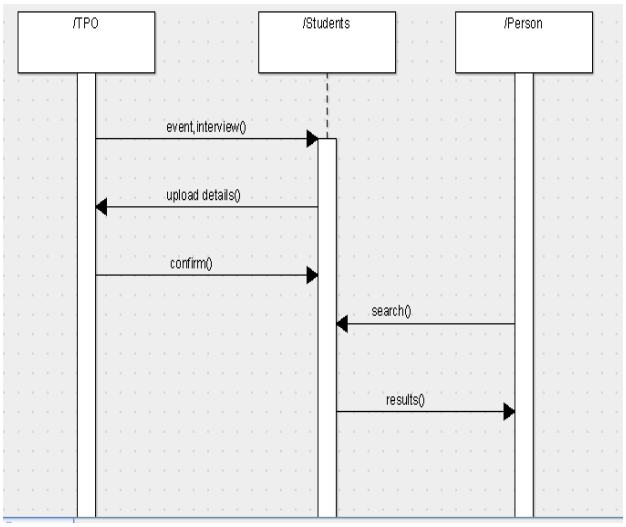


Fig 5. Sequence Diagram for Login module in OTAP

Collaboration Diagram:

Collaboration diagram emphasizes on the structural organization of the objects that send and receive messages. This collaboration diagram for online training and placement can be drawn based on open source tool called Agro-UML which tells each entities operations.

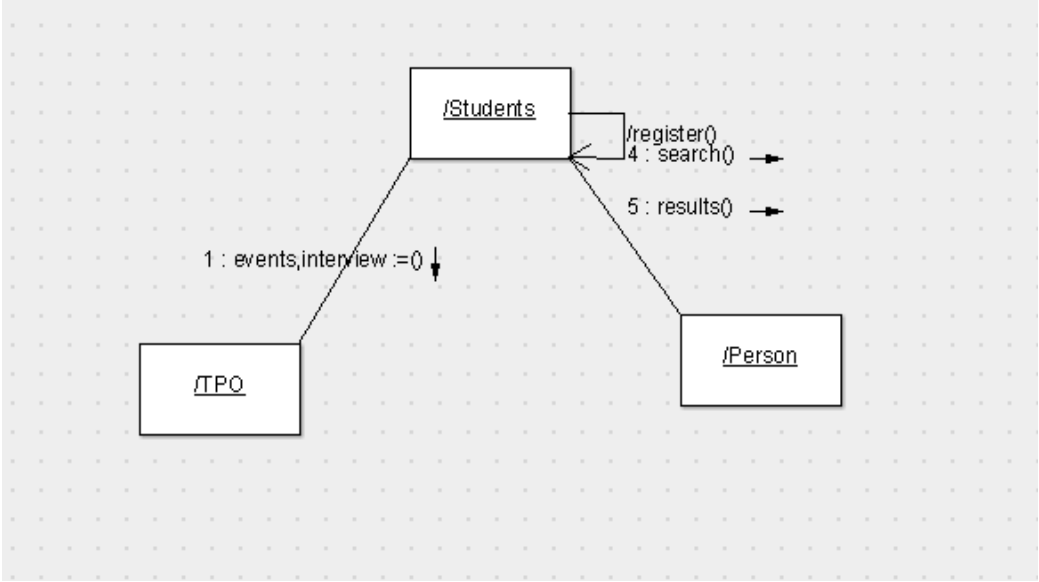


Fig 6. Collaboration Diagram for OTAP

Activity diagram:

Activity Diagram is used to describe dynamic aspects of the system. It is a basically a flow chart that represents flow from one flow to another. In this project, considered Login scenario and is implemented using Agro UML Tool where for updating resume the following activities shown fig 8, are followed.

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 1, pp: (241-254), Month: January-April 2016, Available at: www.noveltyjournals.com

3. Implementation Details

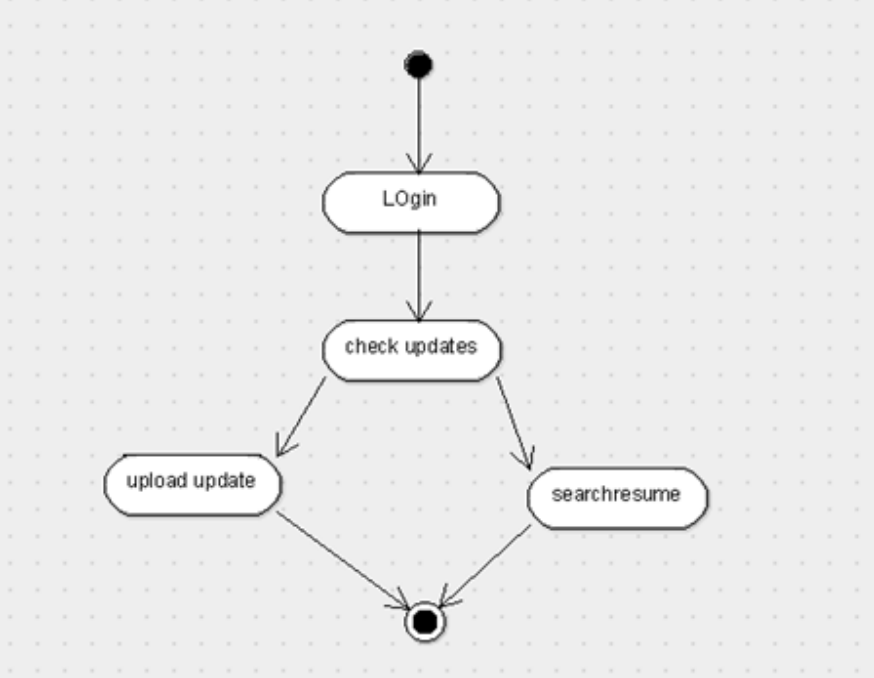


Fig 7. Activity Diagram for OTAP

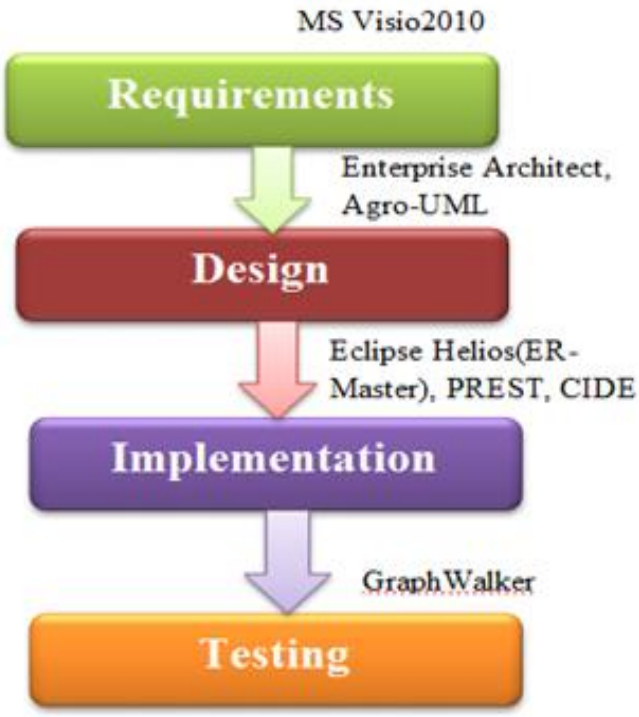


Fig 8.SDLC and its corresponding phase wise tools

ER-Diagram is model used to store data in tables and every row of each table represents one instance of an entity. In the following fig 4 represents the relationship between students, admin and company. Students are joined by admins and each student is allocated with unique Student-ID and student has permissions to modify their personal details and can upload their resume. Company looks for student details such academic performance and regarding details using Eclipse Helios

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 1, pp: (241-254), Month: January-April 2016, Available at: www.noveltyjournals.com

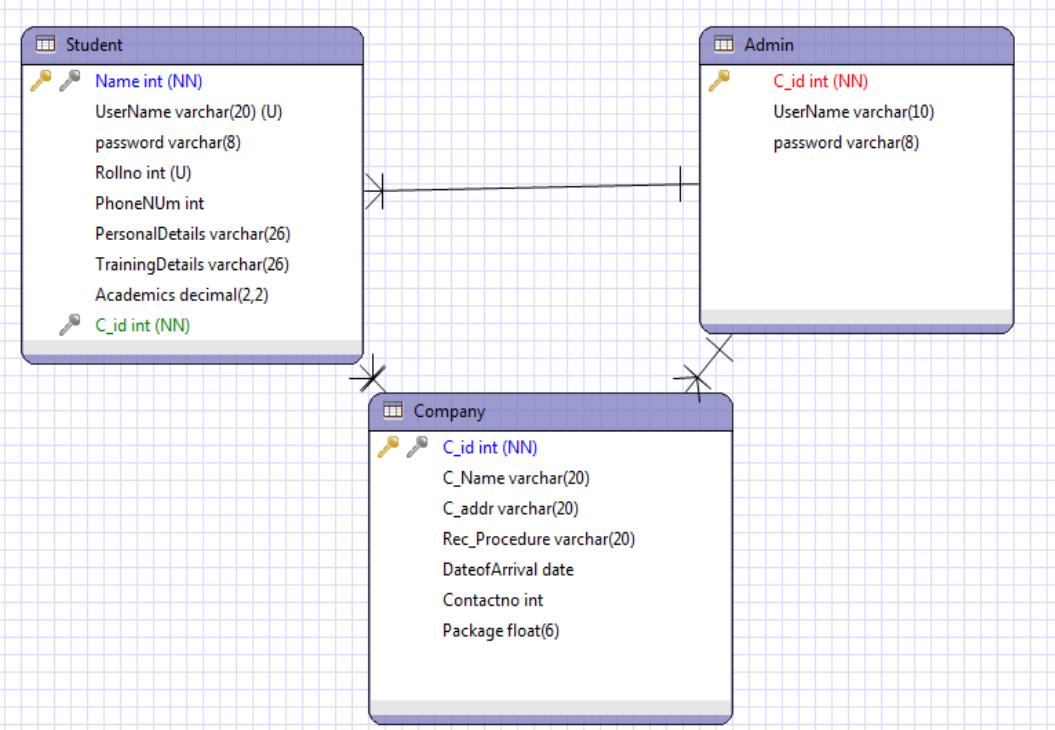


Fig 9: ER-Diagram for OTAP

	Admin	**Company**	**Student**
CODE	DROP TABLE IF EXISTS Admin; CREATE TABLE Admin (C_idint NOT NULL, UserNamevarchar(10), password varchar(8), PRIMARY KEY (C_id));	DROP TABLE IF EXISTS Company; CREATE TABLE Company (C_idint NOT NULL, C_Namevarchar(20), C_addrvarchar(20), Rec_Procedurevarchar(20), DateofArrival date, Contactnoint, Package float(6), PRIMARY KEY (C_id)); ALTER TABLE Company ADD FOREIGN KEY (C_id) REFERENCES Admin (C_id) ON UPDATE RESTRICT ON DELETE RESTRICT ;	DROP TABLE IF EXISTS Student; CREATE TABLE Student (Name int NOT NULL, UserNamevarchar(20), password varchar(8), Rollnoint, PhoneNUmint, PersonalDetailsvarchar(26), TrainingDetailsvarchar(26), Academics decimal(2,2), C_idint NOT NULL, PRIMARY KEY (Name), UNIQUE (UserName), UNIQUE (Rollno)); ALTER TABLE Student ADD FOREIGN KEY (C_id) REFERENCES Admin (C_id) ON UPDATE RESTRICT ON DELETE RESTRICT ; ALTER TABLE Student ADD FOREIGN KEY (Name) REFERENCES Company (C_id) ON UPDATE RESTRICT ON DELETE RESTRICT ;

PREST is used in coding phase and whenever the code is given as input to this tool then the defects will identified and displayed in the form of true and if it contains non-defeats at that time results may display in false. When the output will

be both true and false then the 10 fold cross algorithm will be used and the predication result will appear in the predication view.

CIDE provides a number of innovations. CIDE annotates code based on the underlying structure and can hence guarantee syntactic correctness of all generated products. On top, a product-line-aware type system even guarantees that all variants are type safe, i.e., they can be compiled. To cope with scattered code, CIDE provides views on the source code. To reduce code obfuscation, CIDE experiments with using visual representations, such as background colors to represent annotations.

4. Testing Details:

In Graphwalker tool, if we write the test cases manually it take lot of time to read and execute but in case using a Graphwalker tool based on MBT techniques it generate the test cases automatically and it reads the models automatically finally we can also reduce the time period to write the test cases and to execute them. The latest version of Graphwalker tool 3.4 and it runs on java platform. Graphwalker is very easy to setup and start using in your Test Automation Code. The models create an Abstraction Layer between the Test Design and the implementing Automation Code.

IV. CONCLUSION

Existing System is manual process for Training and Placement activities are heavy and also error prone. This kind of process is often difficult to maintain and manipulate, because the student information continuously changes and also information cannot be viewed instantly on demand as there is a loads of manual work involves in it. Also, the medium of communication and information events are not effective in the existing system as they still use the conventional method i.e., Notification boards for the same. As a general rule, Automation of the above process can significantly reduce the number of man-hours and TPO can become more efficient and productive in managing "Training & Placement" activities. Software Development Lifecycle (SDLC) development with open source tools plays a vital role in the software development. To utilize, other tools in various phases are used effectively and Model Based Testing (MBT) has been used extensively in the system to automate the testing procedures and also reducing the regression bugs/errors.

Currently "Online Application for the Training and Placement" is web based portal. To make a highly scalable web application using distribute computing. So that the system can place on the Cloud which would reduce the cost and maintenance of data.

REFERENCES

[1] T. Savage, M. Revelle, and D. Poshyvanyk, "FLAT3: Feature Location and Textual Tracing Tool," Proc. Int'l Conf. Software Eng. (ICSE), pp. 255-258, 2010.

[2] M.T. Valente, V. Borges, and L. Passos, "A Semi-Automatic Approach for Extracting Software Product Lines," IEEE Trans. Software Eng., vol. 38, no. 4, pp. 737-754, July/Aug. 2012.

[3] B. Dit, M. Revelle, M. Gethers, and D. Poshyvanyk, "Feature Location in Source Code: A Taxonomy and Survey," J. Software: Evolution and Process, vol. 25, no. 1, pp. 53-95, 2012.

[4] B. Cornelissen, A. Zaidman, A. van Deursen, L. Moonen, and R. Koschke, "A Systematic Survey of Program Comprehension through Dynamic Analysis," IEEE Trans. Software Eng., vol. 35, no. 5, pp. 684-702, Sept./Oct. 2009.

[5] C. K€astner, P.G. Giarrusso, T. Rendel, S. Erdweg, K. Ostermann, and T. Berger, "Variability-Aware Parsing in the Presence of Lexical Macros and Conditional Compilation," Proc. Int'l Conf. Object- Oriented Programming, Systems, Languages and Applications OOPSLA), pp. 805-824, 2011.

[6] N.F. Schneidewind, "Methodology for Validating Software Metrics," IEEE Trans. Software Eng., vol. 18, no. 5, pp. 410-422, May 1992.

[7] S. Lessmann, B. Baesens, C. Mues, and S. Pietsch, "Benchmarking classification modelsfor software defectprediction: A proposed framework and novel findings," IEEE Trans. Softw. Eng., vol. 34, no. 4, pp. 485-496,Jul./Aug. 2008.

[8] I. Myrtveit, E. Stensrud, and M. Shepperd, "Reliability and validity in comparative studies of softwareprediction models," IEEE Trans. Softw. Eng., vol. 31, no. 5, pp. 380–391, May 2005.

[9] T. Raeder, T. Hoens, and N. Chawla, "Consequences of variability in classifier performance estimates," in Proc.IEEE 10th Int. Conf. Data Mining, 2010, pp. 421–430.

[10] Q. Song, Z. Jia, M. Shepperd, S. Ying, and J. Liu, "A general software defect- proneness predictionframework," IEEE Trans. Software. Eng., vol. 11, no. 3, pp. 356–370, May 2011.

[11] M. Victor and N. Upadhyay, "Selection of software testing technique: A multi criteria decision making approach," Commun. Comput. Inf. Sci., vol. 204, pp. 453–462, 2011.

[12] T. Y. Chen, P. L. Poon, and T. H. Tse, "A choice relation framework for supporting category-partition test case generation," IEEE Trans. Softw. Eng., vol. 29, no. 7, pp. 577–593, Jul. 2003.

[13] A. Abdurazik and J. Offutt, "Using UML collaboration diagrams for static checking and test generation," Proc. 3rd Int. Conf. The Unified Modeling Lang.: Advancing the Standard, pp. 383–395, Oct. 2000.

[14] A. Hartmann and K. Nagin, "The AGEDIS tools for model based testing," in Proc. ACM SIGSOFT Int. Symp. Softw. Testing Anal., vol. 4, Jul. 2004, pp. 129–132.

[15] K. Stobie, "Model based testing in practice at Microsoft," Electron. Notes Theoretical Comput. Sci., vol. 111, pp. 5–12, 2005.