aurum

ALTINBAŞ ÜNİVERSİTESİ MÜHENDİSLİK SİSTEMLERİ VE MİMARLIK DERGİSİ
ALTINBAŞ UNIVERSITY JOURNAL OF ENGINEERING SYSTEMS AND ARCHITECTURE

Cilt 1, Sayı 1 | Bahar 2017
Volume 1, No. 1 | Spring 2017, 39-66

OBJECT-ORIENTED ANALYSIS AND DESIGN OF A STUDENT TRACKING INFORMATION SYSTEM FOR VOCATIONAL SCHOOLS IN TURKEY

Mert Özkaya[*], Özgür Korkmaz[**]

Abstract

Vocational and technical high schools provide training in more than 130 occupations. Students are tracked by technical teachers at work with the guidance of school management and reports are prepared by teachers at each visit. It is better to save data digitally then on paper and benefit from a database management system. This paper aims at tracking of practical training process in vocational and technical high schools in Turkey and propose a web-based solution in order to digitalize their current processes and overcome the issues. The application creates an online student tracking system where the manager manages the system by inserting teachers, students, classes, companies, controlling added feedbacks and the teachers view their students and add feedbacks to the system for each student. To implement this as a web application we used LARAVEL as the Technology. To build any web application using LARAVEL we need a programming language such as PHP. LARAVEL uses MySQL to interact with the database as it can easily deploy and maintain a LARAVEL application. Student Tracking System has been evaluated in a school. The teachers and the manager expressed their experiences and feelings during the process by pointing out that the system simplifies the process by providing many features.

Keywords: Vocational school, High school, Education, Online education, Laravel

Türkiye'de Meslek Liselerinde Öğrenci Takip Sistemi ve Nesne Tabanlı Analizi

Özet

Mesleki ve Teknik Anadolu Liseleri 130'dan fazla mesleki alanda eğitim vermektedir. Öğrenciler, okul idaresi kılavuzluğunda öğretmenler tarafından kontrol edilir ve her işyeri ziyaretinde öğretmenler tarafından, öğrenciler hakkında raporlar tutulur. Bu kayıtların, bir veri tabanı yönetim sistemi kullanılarak dijital olarak saklanması kâğıt üzerinde saklanmasından daha faydalıdır. Bu yazı ile Türkiye'deki Mesleki ve Teknik Anadolu Liselerinde yürütülen staj sistemi takip edilerek; var olan sistemi dijitalleştirecek ve sorunları çözecek web tabanlı bir çözüm önerilmektedir. Uygulama, ilgili idarecinin sisteme öğrenci, firma, öğretmen ve sınıf ekleyerek sistemi yönetebildiği; girilen kontrol dosyalarını inceleyebildiği, ilgili öğretmenin ise öğrenci bilgilerine erişebildiği ve ilgili öğrenci için kontrol dosyaları ekleyebildiği çevrimiçi bir staj takip ortamıdır. Bu ortamı bir web uygulaması olarak oluşturmak için LARAVEL teknolojisini kullandık. Herhangi bir web uygulamasını LARAVEL kullanarak oluşturmak için PHP gibi bir programlama diline ihtiyacımız vardır. LARAVEL, kolay bir şekilde bir LARAVEL uygulaması oluşturup sürdürebildiği için veri tabanı ile etkileşim için MySQL kullanmaktadır. Staj Takip Sisteminin bir okulda değerlendirilmesi gerçekleştirildi. Öğretmen ve idareciler bu süreçte sistemin birçok özellik sağlayarak var olan sistemi çok kolaylaştırdığına dikkat çekerek deneyimlerini ve duygularını paylaştılar.

Anahtar kelimeler: Meslek lisesi, Lise, Eğitim, Online eğitim, Laravel

[*] *Department of Computer Engineering, Altınbaş University, mert.ozkaya@altinbas.edu.tr*
[**] *Department of Computer Engineering, Altınbaş University*

1. INTRODUCTION

It can be said that the vocational and technical education system in Turkey is divided into two parts: Theoretical (school training) and practical (in-company training). Vocational and technical high schools provide training in more than 130 occupations and leads to the qualification of specialized worker and technician. Theoretical education is provided in schools and practical training is provided in various companies. Vocational education centers provide mainly informal education. There are more than 19 different school types for vocational and technical high schools. There are also vocational education centers, Open Education and special private schools. The MoNE is responsible for all vocational and technical high schools. Vocational and technical education is officially co-educational and boys and girls attend schools designed for the other gender (Altın, R., Yalçın, O., & Kalkan, Ö.). After 8th class, students are placed to various schools according to transition from primary to secondary education system in Turkey. In ninth class, all students in high schools get the same education. After ninth class, there are two kind of school types for students in technical and vocational high schools: Technical High school and industrial high school.

Branch courses in technical and vocational high schools offers courses towards various professions. In addition, each branch contains some sub-branches. Technical high schools are not included in this generalization. Curriculums of technical high schools are similar with the general high schools as of the general education courses and branch courses of natural sciences branch (Altın, R., Yalçın, O., & Kalkan, Ö). There are about 225 occupational branches in vocational and technical education institutions. Industrial and technical branches: Plastics technology, apparel machinery maintenance and repair, Apparel, olive technology, computer-aided industrial modeling, decorative arts, automotive technologies, furniture and decoration, metal technology, machine technology, information technologies, electric technologies, electronic technologies, industrial casting, nourishment technology, construction technology, plastic arts and design, etc. Commerce and tourism branches: Radio, cinema and television, public relations and promotion, accounting and finance, insurance trade and risk management, computers, marketing, catering services, accommodation services, travel agency, travel, recreational services, tourism, journalism, office management and secretary, etc. Social services branches: Child development and education, Skin care and hairdressing, organization services, etc.

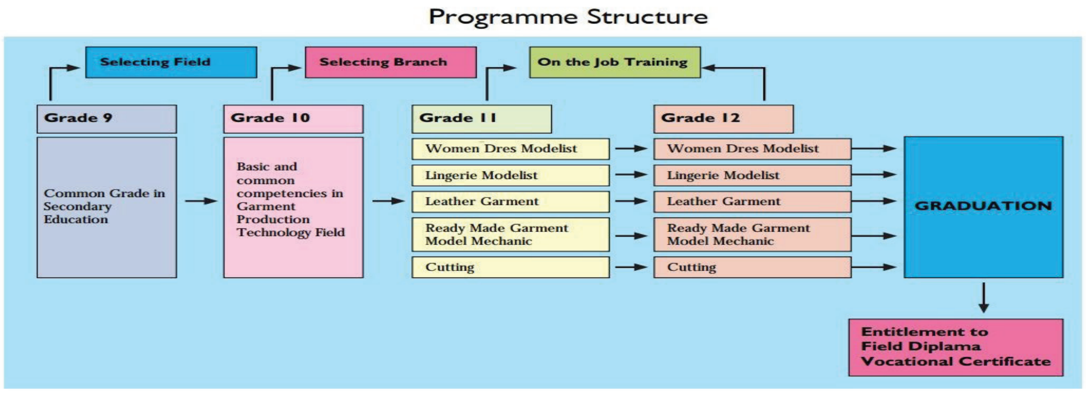

Figure 1. Program Structure

Practical training is 3 days a week during the 12th class for industrial vocational high schools and 5 days a week during the summer break between 11th and 12th class for technical vocational high schools (Figure 1).

The process starts with a meeting between technical manager and the technical teachers. Then, students assigned to teachers according to their distance to school and some other points like number of students to be tracked. If it is possible students assigned to teachers according to their branch to provide better student tracking process. For example, Students of information technologies assigned to teachers of information technologies or students of metal technologies to teachers of metal technologies. Sometimes it is not possible to apply this kind of distribution because of overcrowded students. There should be a fair distribution to prevent any polemic. After getting information and feedback papers from manager; teachers visit assigned students at work according to the predetermined rules at the meeting. The intervals between the visits can be a week, a month, two weeks or even two months. At the visit, teachers ask questions to company manager, person in charge of the student, the student. Then, prepares a report for students and return them to manager to be stored.

1.1 Motivation

Teachers visit assigned students at work after getting the list of the assigned students with some information on paper. These papers include the student's name, class, phone number, company address etc. The intervals between the visits can be a week, a month, two weeks or even two months. At the visit, teachers ask questions to company manager, person in charge of the student and the student; prepares a signed report for each student and return them to manager to be stored.

An online student tracking system will be created to simplify the process and to provide a paper-free solution. The system will have all information about all the members of the system and will reduce paperwork and all the data gathered and stored on database might be used to create career opportunities for students in future. For example, the system will have many information about the students: their pictures, google map links of the companies they work for, their parents' phone numbers, their phone numbers and e-mails. Therefore, it will be easier for teachers to get information about the students during their visit at work. Teachers will be able to see the pictures of the students beforehand, be able to phone to them or their parents if an emergency emerges and all the information can be gathered from their mobile phones or tablets. Teachers will not need to carry around papers.

Teachers have to prepare reports about students at each visit. For instance, let us think that a teacher has five assigned students for a week that means at least five reports a week, twenty a month and two-hundred in a year. So, recording this reports in an online system will reduce the paperwork and creates various opportunities for students: Teachers can see development of students during practical training very easily by inspecting the reports saved online and all the reports gathered can be used for career opportunities.

Technology can play a crucial role in whole education system. Schools need to manage more information day by day. Student registration operations, student transfer operations, marks, document operations, weekly lesson plans and some other operations all need DBMS because of the enormous data at hand. E-Okul is a web school management information system, which was opened for use in 2007 by

MoNE of Turkey. All the info of students from entrance to graduation are recorded in the system. As we look deeply into the system, we can easily notice that there is not any record of practical training period, except year-end marks of students.

1.2 Goal of this paper

The goal of this paper is to develop a web site application making up for drawbacks mentioned in motivation section by creating a web-based data management system that will allow the vocational high schools to manage student, teacher and company data to facilitate the process.

The Student Tracking system will allow teachers the ability to view and complete the process and evaluation online. All that is needed is a computer or a mobile device and web browser. This system will enhance the quality of the teacher/student supervisory experience and facilitate better communication. In addition to these benefits, student-tracking system also offers the following benefits: Provides a paperless solution for tracking and recording students' data, allows students to research potential field opportunities, information about companies can be accessed and updated very easily, allows technical instructor to follow students' developments, provides teachers to easily access student information, allows teachers access to the feedbacks in a paperless system, increases communication between the School and companies.

The Student Tracking system aims to solve the problems encountered in the existing systems while becoming the first web-based student tracking system to be used in vocational high schools in Turkey. Most of the systems reviewed either lack feedback feature or the feature is not suitable enough for the Vocational High Schools. Some of them lack the feedback part. Although some systems have the feedback feature, the feature is undetailed and not similar enough because of the differences in the educational systems. Vocational education in Turkey must have a unique student tracking software and our system is going to be designed to meet all the requirements needed. The second issue is the lack of information needed for all the users in the system. For example, parents' phone numbers are not included in any of the system. However, it is very important for the teachers when they cannot reach the students on the field. Another important information deficit is google-map info of the company for teachers to find companies easily. There is also other important information needed because of the uniqueness of the vocational education in Turkey. The third issue is that most of the system out there are created with old technologies. That is why there is some performance, security and visual quality issues. Our system is going to be PHP, LARAVEL based, and LARAVEL's own security precautions, visual standards and performance features are going to be implemented to the system. The system was scanned with a web vulnerability scanner and a web application scanner and no vulnerabilities encountered during the process.

1.3 Structure of the paper

This paper will explain statements of the problems in the existing related systems and introduce a new system to find a solution the revealed problems. There are three related works discussed. First one is Bahçelievler Trade Vocational High School, second one is Student Tracker Database by Open Source

Development Group of Georgia Southern University's College of Information Technology and the last one is Student Placement tacking by The Boise State School of Social Work. Then, this paper will dwell into Object-Oriented Analysis and Object-Oriented Design of this system. In the Object-Oriented Analysis, Non-Functional Requirements: Audit-Ability, Availability, Backup and Restore, Capacity, Security, Usability, Reliability, Performance, Maintainability, Privacy and Functional Requirements: System Users, Use Case Diagrams and Class Diagrams will be delivered with details. In the Object-Oriented Design, Software Architecture Design that explains system components and their relationships will be explained. Then Dynamic Models of the system, which includes Activity Diagrams and Sequence Diagram, will be inspected.

2 STATEMENT OF PROBLEMS WITH THE EXISTING SYSTEMS

2.1 Related Work

In this section, we review works that relate to our solution and discuss in detail how our solution advances. We review other systems that have been developed for student tracking, as well as proposed notions of similarity. We compare these solutions to ours from several perspectives and show how our solution provides better results than most current solutions.

2.1.1 Related-work 1 (Bahcelievler Trade Vocational High School Online System)

The system was developed by Bahcelievler Trade Vocational High School Information Systems department to be used by staff and students (Bilişim Teknoloji Araştırma Mektep Lügat, 2016). The system divided into two systems: Online exam system and student tracking system. The online exam system creates a paperless solution for all the exams in the school. All teachers create their exams using the system, students take the exams online, and the results can be viewed by using the system. Student tracking system gives assigned students' information to teachers and provides all the papers needed by the teacher.

The system has three type of users: Admin also the technical manager adds teachers, students, lessons and classes to the system with unique user names and passwords. Added teachers can create online exams for students and they can view the student list that they have to track and print the papers needed for student tracking. Students can take online exams. The results are recorded.

The tracking system is a web-based monitoring system designed to keep track of students placed in various companies. The system is used in order to track student placements more effectively while providing teachers an easier solution.

The main issue with the system is lack of feedback feature; students' feedbacks about his experiences or companies' feedbacks about the students; also teachers' idea about both the students and companies and other routine monthly feedbacks. Even though; every paper needed by the teacher for the process can be printed from the system, the system does not have a part where the information on the returned papers can be recorded. Although the system provides teachers the information of the assigned students, the information provided lacks important information about students like students' pictures, which is crucial for teachers to have an easier student tracking process.

Another issue is security. The system was scanned with a web vulnerability scanner and a web application scanner. A web vulnerability scanner is used for searching websites and checking for vulnerabilities to PHP attacks. Vulnerable scripts are shown so that they can fixed. A web application scanner is used for checking applications for injections and other vulnerabilities. A few vulnerabilities encountered during the process but let us not forget security means a lot when information about people stored in databases.

2.1.2 Related-work 2 (Student Tracker Database)

Open Source Development Group of Georgia Southern University's College of Information Technology is responsible for this project. Students from various computer departments formed the group. The purpose of Student Tracker is to help people in the student employment process to document their activities; help them to know their duties and help them to perform those duties. Student Tracker provides a better and online workflow management (Intern Tracker Database, 2017).

The system has three type of users: Schools, companies and students. They register to the system with unique user names and passwords. They have particular duties, which involve creating documents of various kinds such as job postings, resumes, and performance assessments. It reduces paperwork by storing data on computer. Although this system has much better security precautions, a few vulnerabilities encountered. It is a different approach to student tracking and lacks feedback feature again.

2.1.3 Related-work 3 (Student Placement Tracking)

The Boise State School of Social Work uses Student Placement Tracking to track all students in fieldwork. The university and companies communicate students in fieldwork using IPT. It was developed by Orem, a Utah-based Alcea Software. The web-based system was developed with the help of Brigham Young University School of Social Work (Intern Placement tracking, 2017).

IPT manages the information needed by field placement. IPT maintains information about companies, teachers and students and track student assignments. Gathering information and communicating with companies, teachers and Students involved in the program by allowing them to access and update field information using a web-browser is a very simple process.

The system provides communication with both students and teachers and allows manager to create notes regarding the process. Students can look for potential placement opportunities to find a better match for their qualifications. IPT helps to reduce paper wastage and provides the student access to their fieldwork information. IPT allows companies to describe their fieldwork in order to match students' qualifications. Teachers can access information regarding their students.

IPT supports a different kind of feedback system. It allows manager to create notes regarding the process. The processes are different because of the differences in tracking the practical education. For example, students are visited at work at intervals and then their feedbacks are recorded at each visit in technical and vocational high schools in Turkey.

Another issue is security. The system was scanned with a web vulnerability scanner and a web application scanner like other related works before. A few vulnerabilities encountered during the process but security means a lot when information about people stored in databases.

3 OBJECT-ORIENTED ANALYSIS OF THE SYSTEM

Software engineers apply rules to identify classes and objects. This identification defines the hardest part of object-oriented analysis (Booch, Maksimchuk, Engel, & Young, 1994). Object-oriented methods help us to create sets of objects that works together to produce software models that defines their problem domains better than other systems produced by traditional techniques. The systems are easier to adapt to changing requirements, easier to maintain, more robust, and promote code reuse. Object-oriented development enables us to create modules of functionality. Once objects are defined, they will perform their desired functions (Norman, 1996).

In the Object-Oriented Analysis, Non-Functional Requirements and Functional Requirements, which includes system users, use case diagrams and class diagrams will be delivered.

3.1 System functional requirements

3.1.1 Description of the System Users

A system user is a person who interacts with a system using an interface to gain something.

a) **Technical Manager**

Manager is also a teacher or a technical manager who is assigned by the school director to manage the student tracking process. She has the responsibility of adding, deleting and modifying the teachers, students or companies to the system. She has also the responsibility of viewing feedback and warning teachers if something missing. Manager should be able to add, modify and delete teachers, students and companies; can define username and password to the added teachers; can get the information of any member; can view all the feedbacks; can sign the feedback as controlled or not controlled.

b) **Teachers**

Teacher is assigned by the technical manager to follow the students closely on the job training. She has the responsibility of adding, deleting and modifying the feedbacks to the system. She can also view the students related to him with their detailed information; can view the related students; can view details of a specific student; can fill up; delete/modify feedback (If it is not checked by the manager).

3.1.2 Use case diagram

The use case describes the system's behavior under various conditions as it responds to a request from the primary actor. The primary actor interacts with the system and the system responds. Different scenarios can happen depending on the requests made and terms surrounding the requests (Cockburn, 2001). Use cases can be written in many forms: Text form, flow charts, sequence charts or programming languages. People do not need any special training to use them (Cockburn, 2001). A team can use the use case form to document the actual requirements. Another team might later use the former document to finalize design. They might do this for a system of a large company, or a small piece of an application. Both work can be done with the same knowledge level (Cockburn, 2001).

c) **Use Case Packages of the System**

Figure 2 shows the use case packages of the system and Figure 2 shows use case packages with details. Use case of the system is divided into six packages that represent the different functionalities to illustrate the functionality architecture of the system. Authorization Package represents how the login system works: a user can login as a manager or a teacher and also can logout. Account Package shows how the manager can create, delete, edit an account or assign roles to that specific account. After the user logins to the system as a manager, the user can view the all accounts via the related page. If the user wants to add a new account and clicks the "+Öğretmen&İdareci" button then the related use case "Create Account" runs. "Create Account" can be used to create an account, which also includes "Assign Role" use case to assign a teacher or manager role to the account. There is also The "Modify Account" use case, which can be triggered by "İşlemler" button. If "İşlemler" is clicked then the Edit Account and Delete Account use cases run that can be used to edit an account or completely delete it. Company Package shows how the manager can create delete, edit a company. After the user logins to the system as a manager, the user can view the all companies via the related page. If the user wants to add a new company then the related use case "Add Company" runs. "Add Company" can be used to create a company. There is also The "Modify Company" use case, which can be triggered by "İşlemler" button. If "İşlemler" is clicked then the Edit Company and Delete Company use cases run that can be used to edit a company or completely delete it. Class Package shows how the manager can create, delete and edit a class. After the user logins to the system as a manager, the user can view the all classes via the related page. If the user wants to add a new class then the related use case "Add Class" runs. "Add Class" can be used to create a class. There is also The "Modify Class" use case, which can be triggered by "İşlemler" button. If "İşlemler" is clicked then the Edit Class or Delete Class use cases run that can be used to edit a class or completely delete it. Student Package shows how the manager can create, delete, edit a student and view feedbacks assigned to the student and sign or unsign them. After the user logins to the system as a manager, the user can view the all students via the related page. If the user wants to add a new student then the related use case "Add Student" runs. "Add Student" can be used to create a student. There is also The "Modify Student" use case, which can be triggered by "İşlemler" button. If "İşlemler" is clicked then the Edit Student or Delete Student use cases run that can be used to edit a student or completely delete it. There are also View Student Details and View Feedbacks of The Student use cases here. View Feedbacks of the Student use case enables user to view feedbacks belong to a specific student. There is also The “Modify Feedback” use case extends from View Feedback of the Student use case, which can be triggered by “İşlemler” button. If “İşlemler” is clicked then the Sign/Unsign Feedback as Checked or View Details of the Feedback use cases run that can be used to sign or unsign a feedback or view it more detailed. There is also Allow/Deny Feedback Modification for Teacher extends from Sign/Unsign Feedback as Checked use case, which allows or denies modification to feedbacks by teachers. Student List Package shows how the teacher can view the assigned students and view feedbacks assigned to them and add new feedbacks to them.

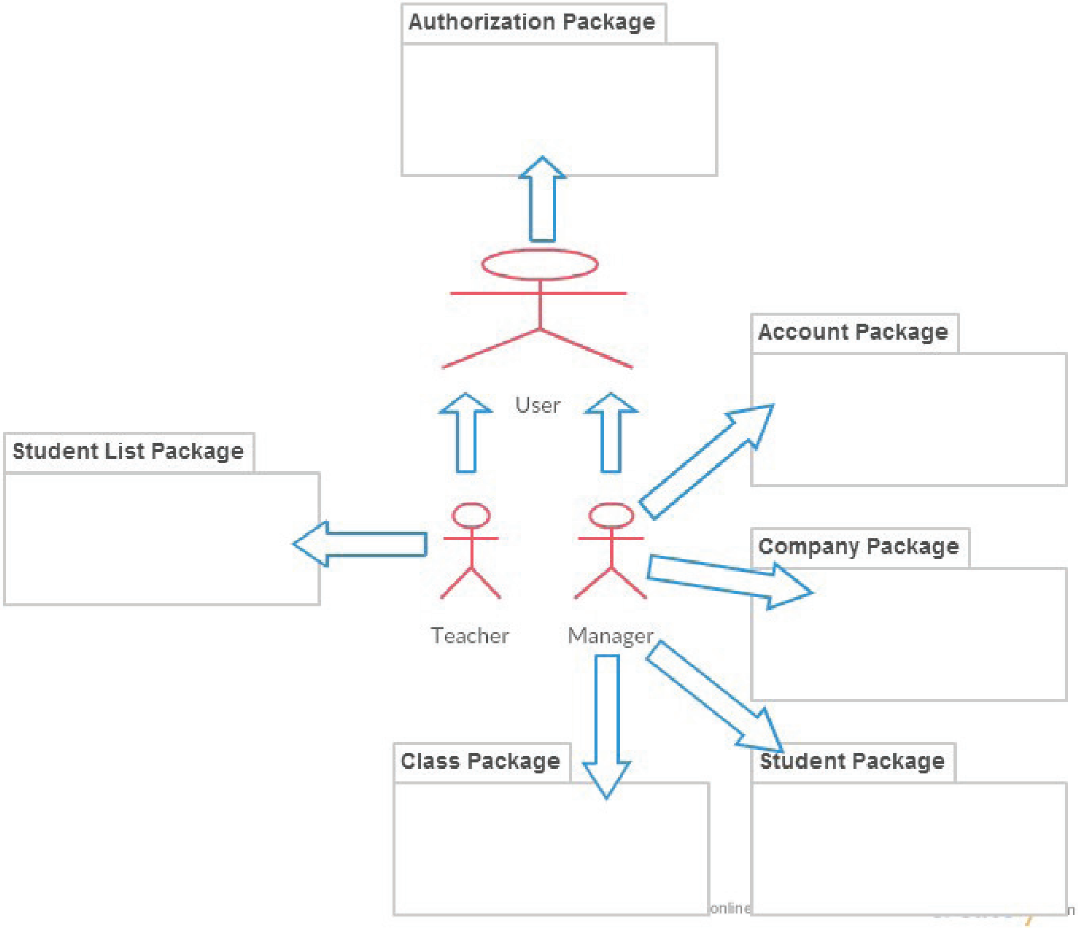

Figure 2. Use case packages of the system.

3.1.3 System Class Diagram

Class diagram is created simply by reading the class names and their operations off the interaction diagrams. Because class diagram gathers class operations and attributes in one place, it is easier to size up the relative complexity of classes in the system. The number of operations in a class correlates with the amount of responsibility handled by the class. If some classes have considerably greater number of operations than the other classes, they should be examined for undiscovered classes and misplaced responsibilities (Marsic, 2012).

MVC is the separation of model, view and controller. It is a way to use when designing classes to avoid mixing code from the other categories. For example, a data grid view should present data once shown but it should not have code on where to retrieve the data. In the same way, while it may have a function to sum up a column, the controller should do the calculation. As can be understood, separation of responsibilities allows flexibility and maintainability. Therefore, I followed rules of MVC model designing the classes in the system.

Figure 3 shows the class diagram packages of the system. The user interacts with the Login Screen and can be allowed or denied.

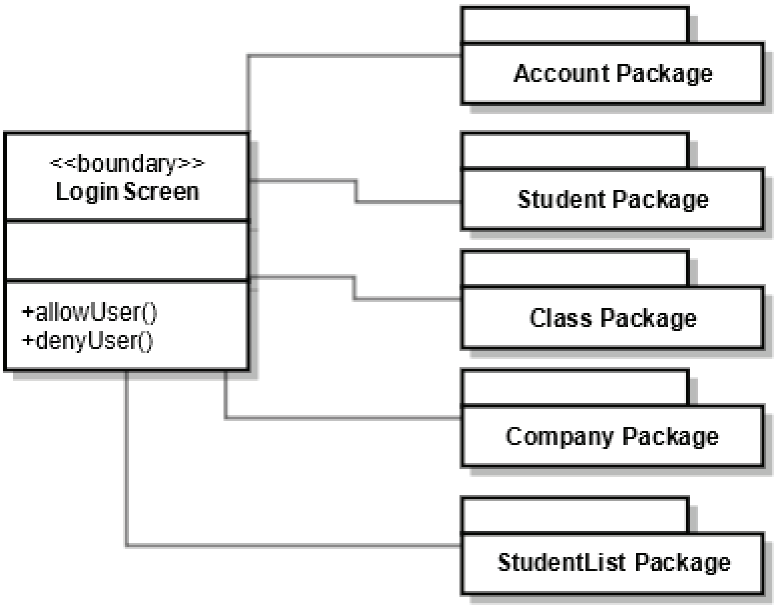

Figure 3. Class diagram packages of all system

Figure 4 shows the ManagerHomePage class diagram packages of the system.

If the user role is manager then home page for manager is displayed and the manager can interact with these packages. Figure 5 shows the TeacherHomePage class diagram packages of the system. If the user role is teacher then home page for teacher is displayed and the teacher can interact with these packages.

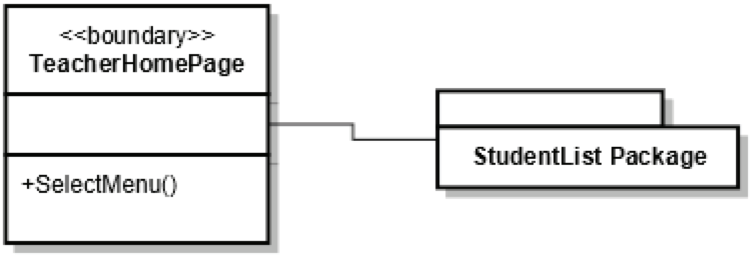

Figure 4. ManagerHomePage class diagram packages of the system.

*d) **ManagerHomePage Class Diagram***

Entities are objects representing system data. Boundaries are objects that interface with system actors. Pages, screens and menus are examples of boundaries that interface with users. Controllers are objects that link boundaries and entities. They connect boundary elements and entity elements, implementing the logic required to manage the elements and their interactions. Figure 6 show entities, boundaries and controllers of the system when manager logins.

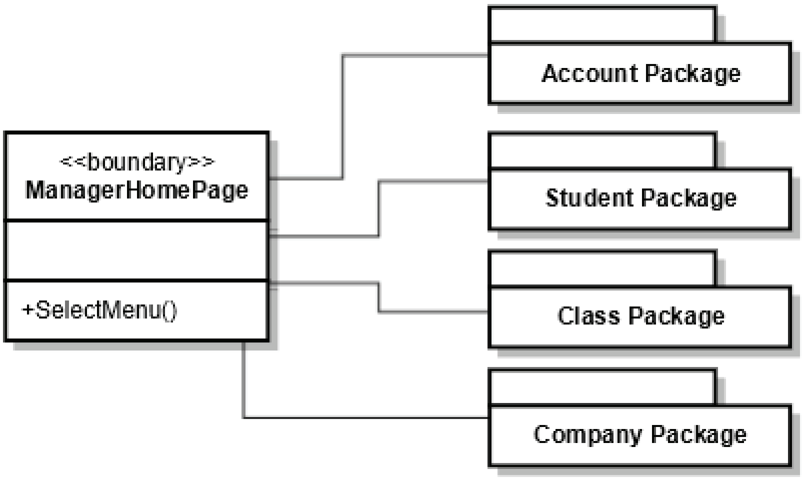

Figure 5. TeacherHomePage class diagram packages of the system

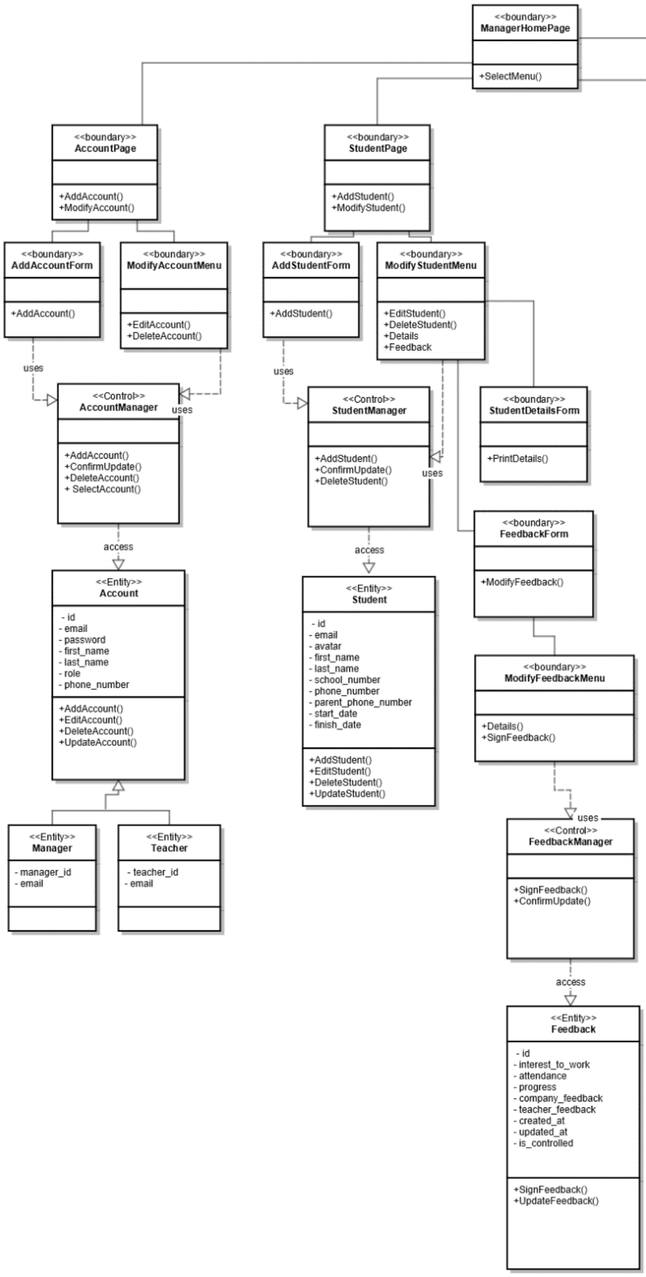

Figure 6. ManagerHomePage class diagram of the system

e) ***StudentList Package Class Diagram***

Figure 7 show entities, boundaries and controllers of the system when teacher logins and how they interact.

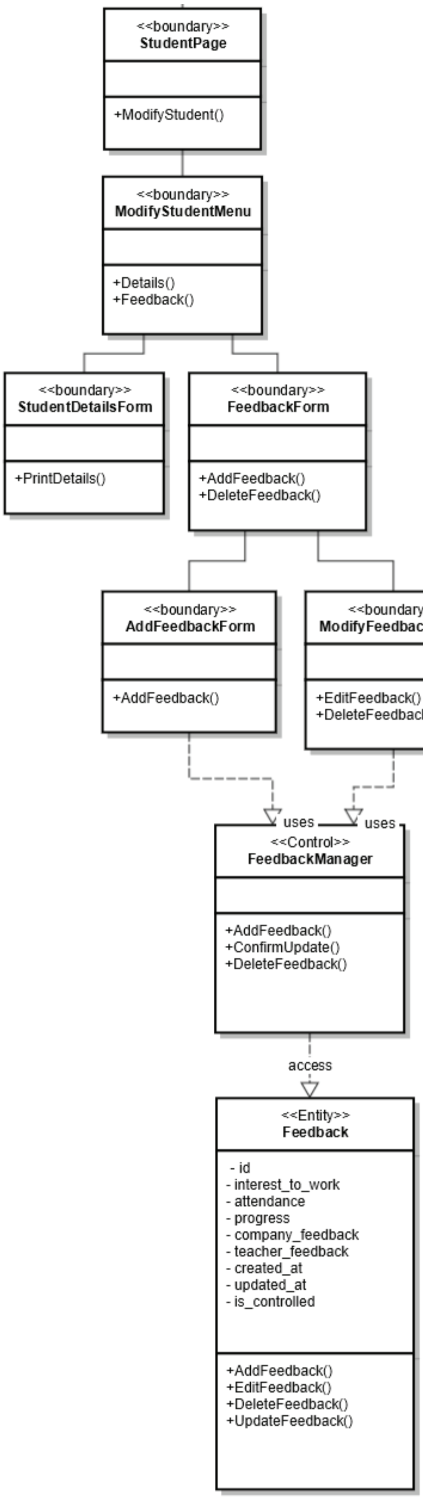

Figure 7. StudentList class diagram of the system

3.2 System non-functional requirements

Functional and non-functional characteristics must be taken into consideration in the development of a quality software system. However, most of the attention in software engineering has been centered on techniques for defining and providing the functions of a software system due to the soft nature of non-functionality. A Non-Functional Requirement measures how well the system must do what it does and it is measured over an interval or range (Chung & Leite). After the implementation of the system is completed, every non-functional requirement is analyzed or compared with the actual results of the system.

The Volere Requirements Specification Template is used for the requirements. It has proved to be a valuable resource for organizations worldwide by saving significant time and money for their requirements activities. It does this by providing a rock-solid template and guide to writing appropriate requirements specifications (Volere Requirements Resources, 2017).

3.2.1 Audit-Ability

Description 1: Individual performance report on work undertaken by the members of staff.

Fit Criteria 1: Preliminary reports taken to ensure that the members of staff are working to what is required every week.

Rationale 1: This is useful as it ensures that every member of staff is on track to the specific task given to them.

3.2.2 Availability

Description 1: The system operates 24 hours a day. Every working sector should be operating.

Fit Criteria 1: The availability will be monitored every hour and the uptime of every working sector should be operational.

Rationale 1: The system should be operating 24 hours a day excluding maintenance down time as this will achieve maximum potential output possible.

Description 2: Scheduled down time for maintenance once a year. The system is not available during this time.

Fit Criteria 2: This will be measured once a year and the system should not be operational at this time.

Rationale 2: This will be to ensure that the system is running efficiently up to the standards required.

3.2.3 Backup and Restore

Description 1: When the system fails, initiates restoration of the database on demand to the most recent backup that is operational from the archive.

Fit Criteria 1: Every 3 months the backup will be checked for and data loss.

Rationale 1: Through system failure this is to ensure that the system is restored must be up to date to the latest restoration point.

3.2.4 Capacity

Description 1: The server is able to accommodate up to 100 users.

Fit criteria 1: Every day the system's performance should be checked to ensure that it could satisfy the amount of users logged in at one time without causing the system to slow down.

Rationale 1: This is useful in the system if every member of staff is logged into the system all at once, the system should still perform to a very high and efficient level.

Description 2: The system should be able to handle over 10 MB amount of data stored a day.

Fit criteria 2: Every day the data stored in the system is examined to check that it can handle over 10 MB of data stored.

Rationale 2: This is useful as it keeps tracks of how much data flows within the system.

3.2.5 Security

Description 1: The system should allow for a restricted level of access:

- **Manager**: Access to all sub-systems except Student-List package.
- **Teacher**: Access to Student-List package only.

Fit Criteria 1: System will be tested every day to ensure that each member of staff has the correct login details.

Rationale 1: This is essential as each member of staff have their own access right.

- No restriction to password length or characters.
- Can contain numbers and symbols.

Fit Criteria 2: For each account, created passwords can be of any length and any character.

Rationale 2: This gives users the flexibility as he or she have the ability to choose their own personal passwords.

3.2.6 Usability

Description 1: Look and feel standards:

- Consistent layout and flow throughout the system
- Matching and steady colors in the system

Fit Criteria 1: The system will be monitored once updated to ensure the layout and structure is consistent.

Rationale 1: This will make it very easy for the user to work in the environment provided.

Description 2: Localization; Turkish keyboard and Language.

Fit Criteria 2: System will be used every day therefore this will be tested daily to ensure that the language and the keyboard layouts are of the Turkish standard.

Rationale 2: This is useful because all members of staff should be able to use and work with the Turkish language and keyboard.

3.2.7 Reliability

Description 1: Mean time between Failures expected no more than twice a month.

Fit Criteria 1: The system will be tested every day to insure there are no errors and

Users can use the system without any failures.

Rationale 1: This is essential, as it will allow users to use the system smoothly without any system breakdowns.

Description 2: Mean time to recovery when system fails should not exceed 2-3 hours.

Fit Criteria 2: System will be tested every minute to avoid any failures in the process of recovery.

Rationale 2: This is essential because there should be a smooth recovery operation on the system to ensure that it will be up and running again as soon as possible.

3.2.8 Performance

Description 1: Response times:

- Application loading not more than 2 seconds at most
- Screen open and refresh time not more than 2 seconds at most.

Fit Criteria 1: System performance will be examined every day to make sure there are no delays at any given time for any process made.

Rationale 1: This is important because in order to achieving the highest level of output, the application loading time and screen refresh rate should not take longer than it should.

3.2.9 Maintainability

Description 1: On system downtime, repairs should not exceed 3 hours' maintenance when restoring the system.

Fit Criteria 1: System will be tested every minute to avoid any failures in the process of restoration.

Rationale 1: This is essential, as there should be a smooth recovery operation on the system to ensure that it will be up and again as soon as possible.

3.2.10 Privacy

Description 1: Customer Loyalty:

- Customer details are to be kept private
- Details of customer are to be utilized only when necessary

Fit Criteria 1: This principle will be maintained and revised every day to insure customers' details are kept safe and not leaked without authorization.

Rationale 1: This requirement is essential to gain customer loyalty and trust at a professional standard. Their details have to be kept private and only used when necessary.

4 OBJECT-ORIENTED DESIGN OF THE SYSTEM

Object-Oriented Design is a process of deciding how a software system will work, how objects will interact with each other to solve specific problems. While a good Object-Oriented Design can make things easier for us, a bad one can make things troublesome. Object-oriented design differs from classic software design, because responsibilities assigned to different components (Budd, 2002).

4.3 Software architecture design

A component diagram shows relationship between the components of a system. A box with a little plug symbol in the top-right corner is the symbol for the component. A component provides one or more interfaces for other components to use. A semi-circle at the end of a line is used to show a component is using another component's interface. Various relationships among components: Component may execute each other's or methods in another component. It is possible for a component to generate another component. Communication is possible between two components (Laganière & Lethbridge, 2005). As depicted, Intern Tracking System consists of management components and users, which interacts with each other to safely maintain an Intern Tracking System. Components provides "lollipop" symbols that provide relationship of an interface classifier. Interface symbols with only a half circle at their end are called sockets. They represent an interface that the component requires.

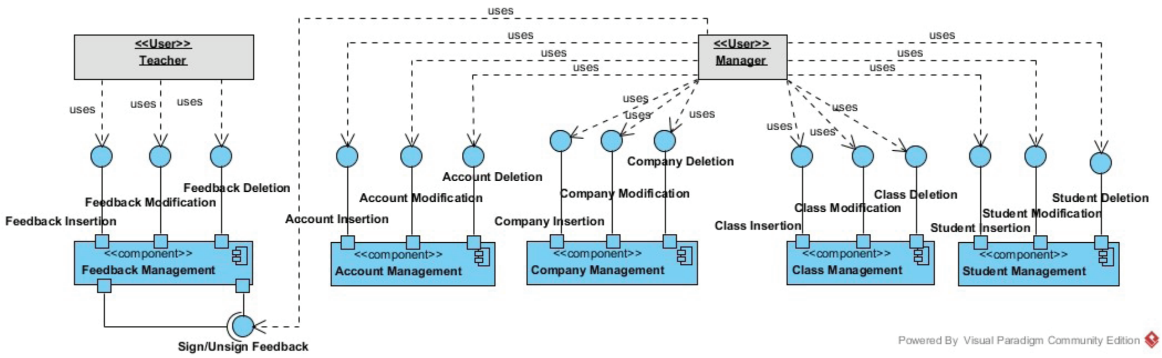

Figure 8. Component diagram of the system

The system in Figure 8, has five components: Feedback Management, Account Management, Company Management, Class Management and Student Management. Feedback management component has three bi-directional ports, which are Feedback Insertion, Feedback Modification and Feedback Deletion and a socket called Sign/Unsign Feedback which,is used by the interface used by the Manager User. They are required by interfaces Teacher User uses. Account Management component has three bi-directional ports, which are Account Insertion, Account Modification and Account Deletion. They are required by interfaces Manager User uses. Company Management component has three bi-directional ports, which are Company Insertion, Company Modification and Company Deletion. They are required by interfaces Manager User uses. Class Management component has three bi-directional ports, which are Class Insertion, Class Modification and Class Deletion. They are required by interfaces Manager User uses. Student Management component has three bi-directional ports, which are Student Insertion, Student Modification and Student Deletion. They are required by interfaces Manager User uses.

4.4 DYNAMIC MODELS

A dynamic model represents the behavior of an object over time. It is used where the object's behavior is best described as a set of states that occur in a defined sequence (Booch, Maksimchuk, Engel, & Young, 1994). It consists of activity diagrams and sequence diagrams. The Dynamic model of the system mainly based on the components that delivered with the component diagram of the system. Behaviors and interaction of the components are explained by using activity and sequence diagrams. For example; Account Activity Diagram of the System (Figure 10) shows how Account Component interacts with other components and Users using interfaces and on the other hand Add Student Sequence Diagram (Figure 19) shows how the Manager User interacts with the Student Management Component using interfaces.

4.4.11 Activity Diagram

Activity Diagrams used to model sequence of actions as part of the process flow. It focuses on the work performed in the implementation of an operation and the activities in a use case instance or in an object (Bhattacharjee & Shyamasundar, 2009).

Concurrent activities like using forks, joins and rendezvous can be shown easily. A fork has one incoming transition and multiple outgoing transitions. A join has multiple incoming transitions and one outgoing transition. A rendezvous has multiple incoming and multiple outgoing transitions. An activity diagram also has two types of nodes for branching within a single thread.

A decision node has one incoming transition and multiple outgoing transitions, each with a Boolean guard in square brackets. A merge node has two incoming transitions and one outgoing transition. It is used to bring together paths had been split by decision nodes (Laganière & Lethbridge, 2005).

f) Login Package

Figure 12 shows the process of logging into a website, from entering your username and password, to successfully logging in to the system.

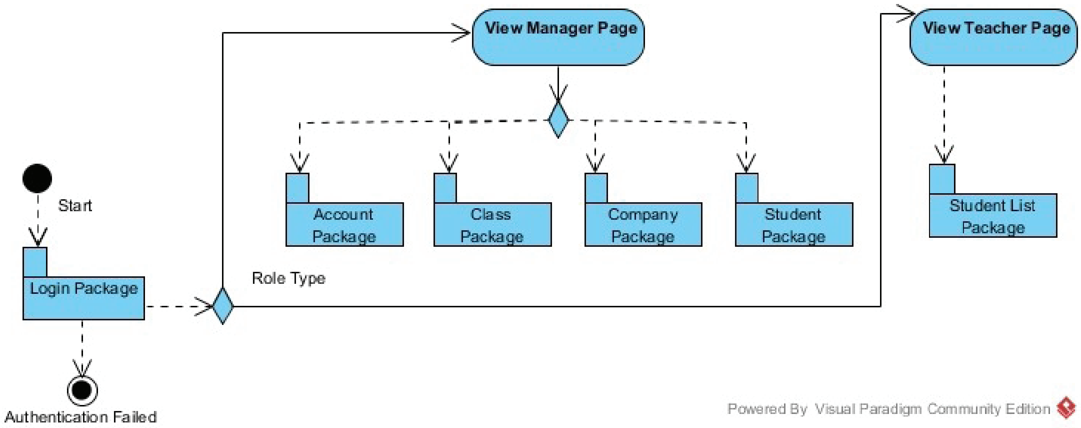

Figure 9. Activity diagram packages of the system

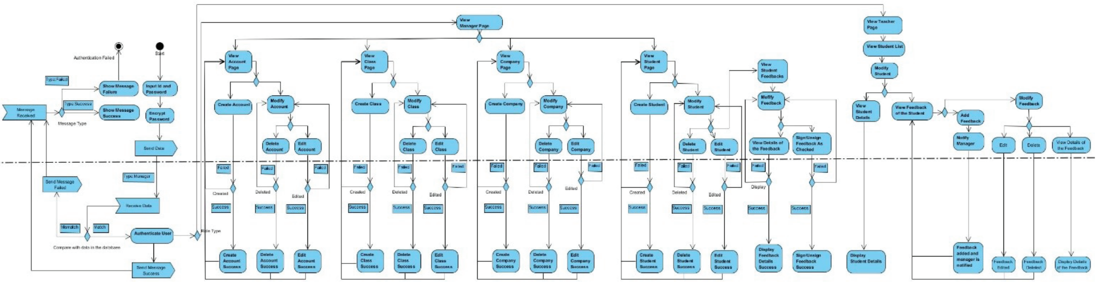

Figure 10. Activity diagram of the system

*g) **Account Package***

Figures 11-12 show the process of adding an account, deleting it or editing it.

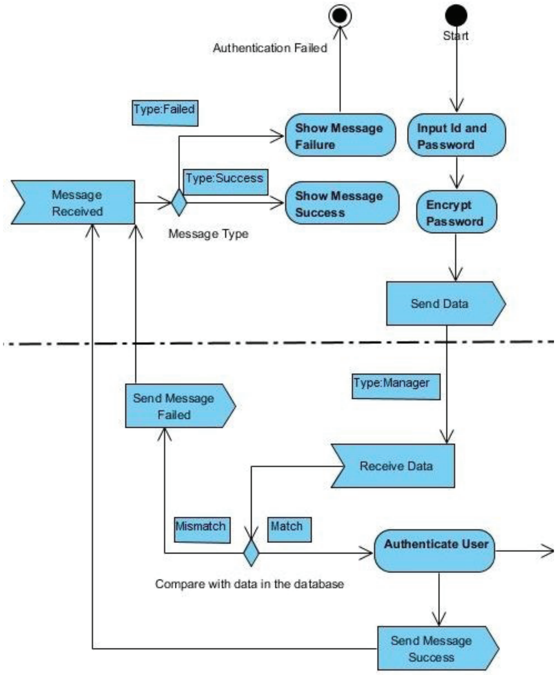

Figure 11. Login activity diagram of the system

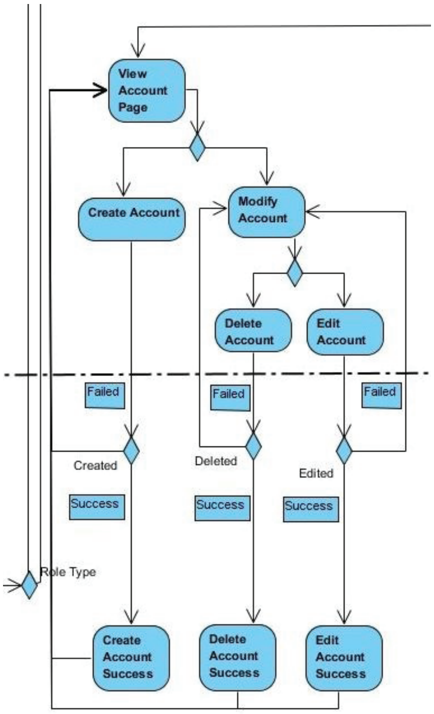

Figure 12. Account activity diagram of the system

h) ***Class Package***

Figure 13 shows the process of adding a class, deleting it or editing it.

i) ***Company Package***

Figure 14 shows the process of adding a company, deleting it or editing it.

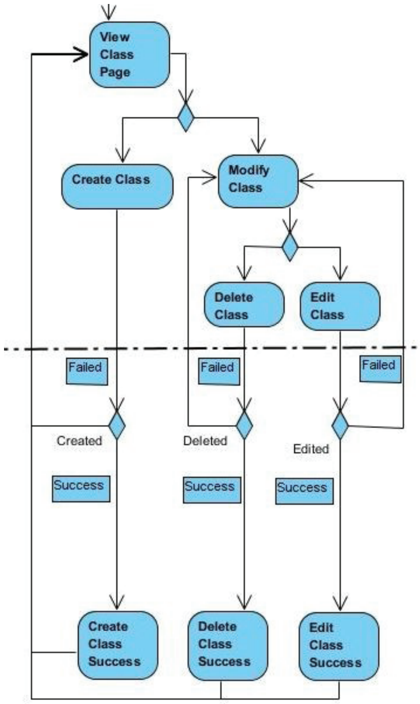

Figure 13. Class activity diagram of the system

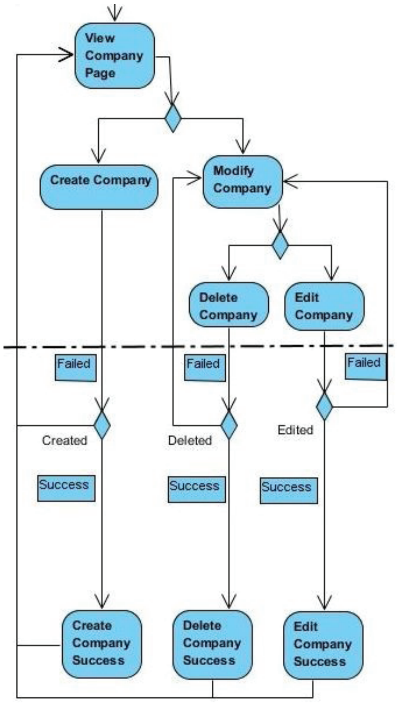

Figure 14. Company activity diagram of the system

j) ***Student Package***

Figure 16 shows the process of adding a student, deleting it or editing it also see feedbacks of the students and sign them as controlled.

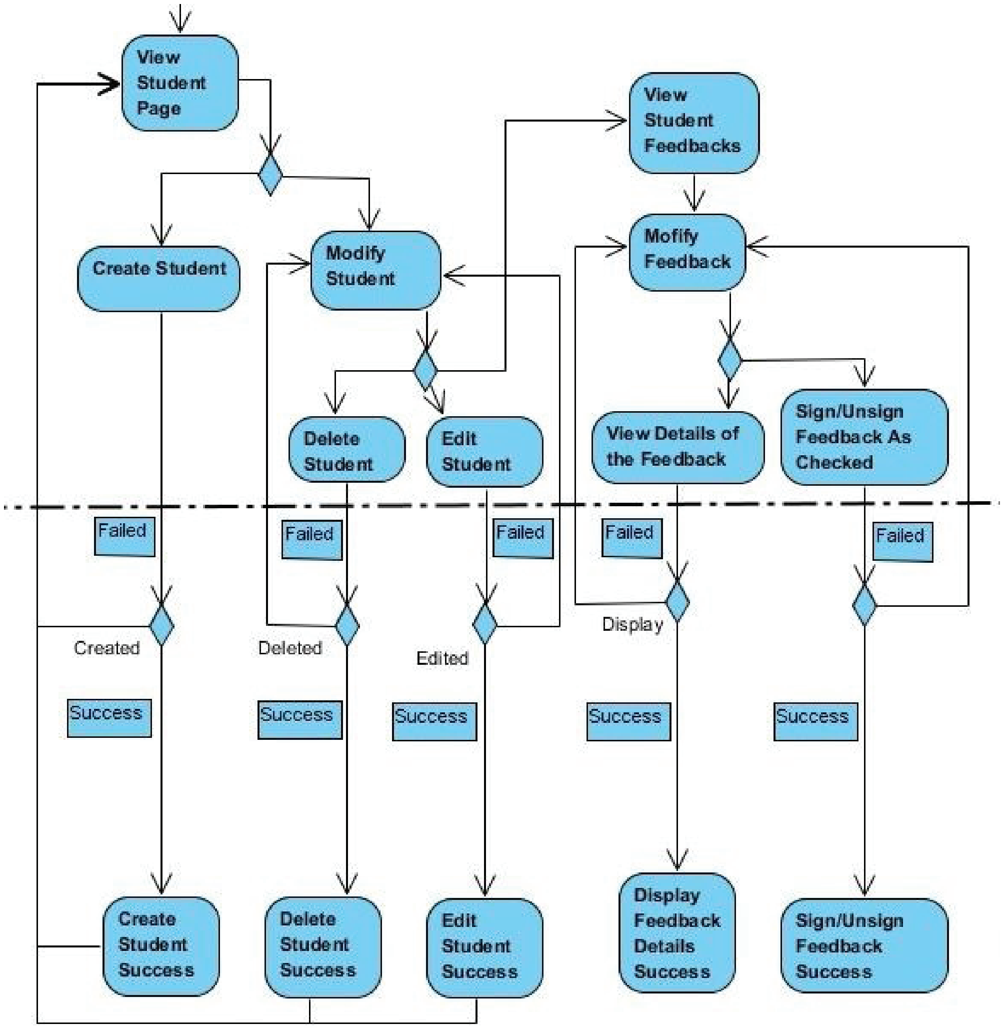

Figure 15. Student activity diagram of the system

k) Student List Package

Figure 16 shows the process of adding a feedback to a student, deleting it or editing it.

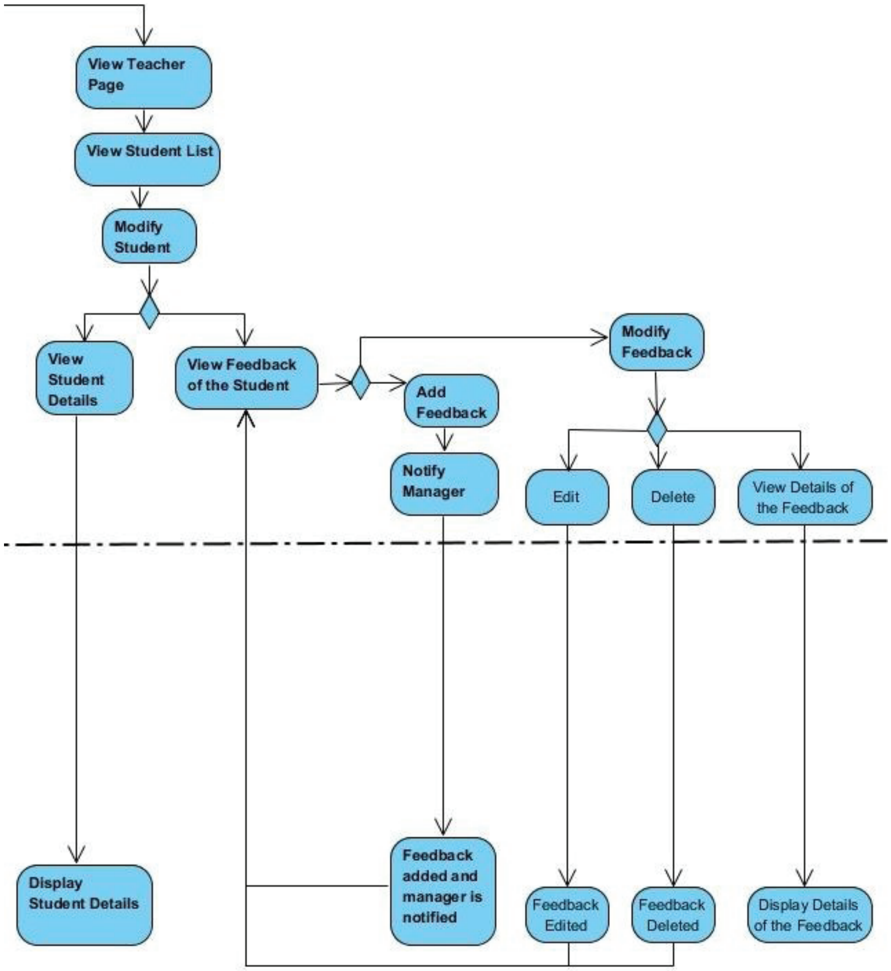

Figure 16. Student list activity diagram of the system

4.4.12 Sequence Diagram

A sequence diagram is used to display exchanging of sequential messages between interaction objects in the system. Information about the flow of control during the interaction, such as if-then-else conditions and iteration might be displayed in the diagram (Garousi, Briand, & Labiche, 2005). The sequence diagrams for "Add Student", "Login", "Sign Feedback as Controlled", "Display Feedbacks of a Student", "Add Feedback" are shown respectively.

*l) **Add Feedback***

Figure 17 shows the process of adding a feedback when teacher user is on the feedbacks of a student page.

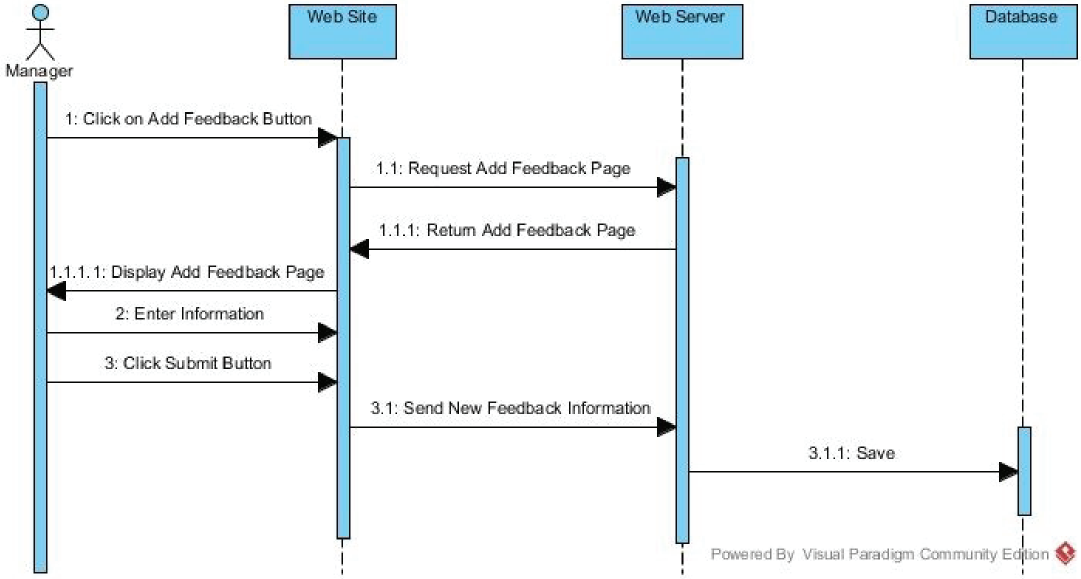
Manager
Web Site
Web Server
Database
1: Click on Add Feedback Button
1.1: Request Add Feedback Page
1.1.1: Return Add Feedback Page
1.1.1.1: Display Add Feedback Page
2: Enter Information
3: Click Submit Button
3.1: Send New Feedback Information
3.1.1: Save
Powered By Visual Paradigm Community Edition

Figure 17. Add feedback sequence diagram of the system

m) Add Student

Figure 18 shows the process of adding a student when manager user is on the Student Page.

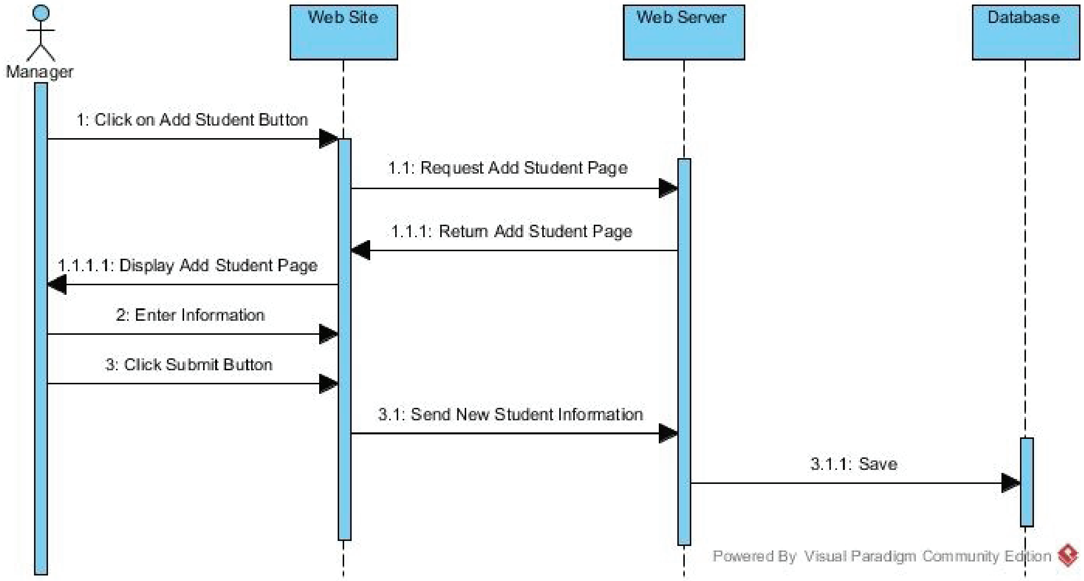
Manager
Web Site
Web Server
Database
1: Click on Add Student Button
1.1: Request Add Student Page
1.1.1: Return Add Student Page
1.1.1.1: Display Add Student Page
2: Enter Information
3: Click Submit Button
3.1: Send New Student Information
3.1.1: Save
Powered By Visual Paradigm Community Edition

Figure 18

n) Login

The process of logging into a website, from entering your username and password, to successfully logging in to the system.

o) Sign Feedback as Controlled

The process of a manager signing a feedback as controlled after login to the system.

p) Display Feedbacks of a Student

The process of a manager views the feedbacks of a chosen student. This sequence diagram is the same for the teacher user too.

5 THE IMPLEMENTATION AND TESTING OF THE SYSTEM

This system is a web application that carries out the student tracking process of a vocational and technical high school in Turkey. It is responsive, mobile compatible, user friendly and it has Session based authentication for granting access to two different user roles. PHP is used as the programming language and Mysql is used as the DBMS. Laravel framework is used as the web framework while HTML5, Bootstrap and JavaScript are used for the designing the interfaces and their behaviors. Sublime text editor and Phpstorm are used as development platforms. After development, Nginx is used for deploying and running the web application.

The system was evaluated by Information Technologies Department of Profilo Vocational and Technical Anatolian High School for the summer training period. After the deployment of the web site, functions are tested by feeding them input and analysing the output. Tables are prepared for every input-output operation. The tables included rows, which are scenario, test step, expected outcome and actual outcome. In the scenario row, the need for verification for the activation of input explained. In the test step row, action to execute to scenario given. In the expected outcome row, the expected outcome is explained and with the actual outcome row, the actual result after the scenario tested is given. Non-functional testing was also executed. Non-functional testing examines how a system operates, rather than specific behavior. Audit-ability, availability, backup-restore, capacity, security, usability, reliability, performance and privacy are all tested. The system was available for 24 hours a day for a month without any problems and never failed during the process so a backup restored just to be sure if it is working and it worked perfectly. System's performance was checked to ensure that it could satisfy the amount of users logged in at one time, without causing the system to slow down during the test month and it performed efficiently. Every day the data stored in the system was examined to check that it could handle data stored. Server handled the situation without any trouble. All major web-testing methods were used for security. Internal url was pasted directly into address bar without login and Internal pages did not open. Url options were changed directly after a successful login and access denied the user to view others pages. Invalid inputs in input fields like login username, password, and input text boxes were tried and the system reactions were controlled. SSL is tested for security measures and a message was displayed when user switch from non-secure pages to secure pages. All transactions, error messages, security breach attempts were logged in log files on web server. System was monitored to ensure the layout and structure is consistent. This made it very easy for the user to work in the environment provided. Software coding standards are important in producing maintainable code. All code was done according to these standards during both the development and maintenance phases.

The system can be accessed using http://www.ozgurkorkmaz.xyz/auth/login and for the passwords and the user names please contact Asst. Prof. Dr. Mert ÖZKAYA or Özgür KORKMAZ from Department of Computer Engineering, Altınbaş University.

6. CONCLUSION

The Internet has become a major resource in modern business, thus tracking systems has gained significance not only from the school's managers but also from the technical teachers' point of view. For the school manager, Student Tracking System generates new opportunities to simplify the tracking process and for the teachers, it makes it easier to track students and their data. In-company training is a big part of the vocational education and have big impact on students' career. Therefore, recording the process with detailed information is very important for students too. Hence, we have designed the project to provide the user with easy navigation, retrieval of data and necessary feedback as much as possible.

In this project, the user is provided with a student tracking web site that can be used to track students online. To implement this as a web application we used LARAVEL as the Technology. LARAVEL has several advantages such as enhanced performance, scalability, built-in security and simplicity. To build any web application using LARAVEL we need a programming language such as PHP; so PHP was the language used to build this application. LARAVEL uses MySQL to interact with the database as it can easily deploy and maintain a LARAVEL application, provides fast data access, easy installation and simplicity.

A good design must be accompanied with user-friendly application logic. It should be convenient for the manager to view the contents of the system and to be able to remove or add students, teachers or managers, companies, classes to the system and for teachers to be able to view students and add feedbacks to them. The application described in this project provides a number of features that are designed to make the users more comfortable.

This project helps in understanding the creation of an interactive web page and the technologies used to implement it. The building of the project has given me a precise knowledge about how LARAVEL is used to develop a website, how it connects to the database to access the data and how the data and web pages are modified to provide the user with an application.

Student Tracking System was evaluated in a vocational and technical high school for a month. Managers and teachers were given a lecture about the web site. Manager added the required data to the system. Teachers visited the students at work and added gathered feedbacks to the web site.

The teachers and the manager expressed their experiences and feelings during the process. Teachers pointed out that the system simplifies the process by providing features like the pictures of the students, phone number of students and their parents and all the features can be accessed from their mobile phones or tablets. They also pointed out that recording feedbacks to the database simplifies monitoring students' developments, reduce paper work and may create career opportunities for students. Managers pointed out that it made to process easier to manage for them.

REFERENCES

Altın, R., Yalçın, O., & Kalkan, Ö. (n.d.). *An Overview of Turkish Vet System* . Ankara. Retrieved from http://projeler.meb.gov.tr/pkmtr/document/e_kitap/vet_reform_turkey.pdf

Bhattacharjee, A., & Shyamasundar, R. (2009, January). *Activity Diagrams : A Formal Framework to Model Business Processes and Code Generation.* Mumbai: ETH Zurich, Chair of Software Engineering.

Bilişim Teknoloji Araştırma Mektep Lügat. (2016, 1 10). Bilişim Teknoloji Araştırma Mektep Lügat Web Site: http://b-atml.com/ adresinden alındı

Booch, G., Maksimchuk, R. A., Engel, M. W., & Young, B. J. (1994). *Object-Oriented Analysis and Design with Applications.*

Budd, T. (2002). *An Introduction to Object-Oriented Programming.* Addison Wesley Longman.

Chung, L., & Leite, J. C. (tarih yok). *On Non-Functional Requirements in Software.*

Cockburn, A. (2001). *Writing Effective Use Cases.* Addison-Wesley, c.

Coronel, C., & Morris, S. (2012). *Database Systems. Design, Implementation and Management.* Boston: Cengage Learning.

Garousi, V., Briand, L., & Labiche, Y. (2005, September). *Control Flow Analysis of UML 2.0 Sequence Diagrams.* Ottawa,: Software Quality Engineering Laboratory.

Intern Placement tracking. (2017, 1 28). Boise State University web Site: https://hs.boisestate.edu/socialwork/fieldwork/ipt/ adresinden alındı

Intern Tracker Database. (2017, 1 28). Intern Tracker Site: http://intern-tracker.sourceforge.net/ adresinden alındı

Laganière, R., & Lethbridge, T. C. (2005). *Object-Oriented Software Engineering Practical software development using UML and Java.* New York: McGraw-Hill Education.

Marsic, I. (2012). *Software Engineering.* Rutgers University: New Jersey.

Norman, R. J. (1996). *Object Oriented Systems Analysis and Design.* Prentice Hall College Div; 1 edition.

Ramakrishnan, R., & Gehrke, J. (2003). *Database Management Systems.* New York,: McGraw-Hill Higher Education.

Volere Requirements Resources. (2017, 1 26). Volere Requirements Resources Web Site: http://www.volere.co.uk/template.htm adresinden alındı

Watson, R. T. (2006). *Information Systems.* Georgia: Global Text Project.