ENHANCING LEARNING ANALYTICS PLATFORM FOR SECONDARY SCHOOLS: DESIGN AND DEVELOPMENT OF NEW INDICATORS

BACHELOR THESIS REPORT

AUTHOR

HARIHARA SUBRAHMANIAM MURALIDHARAN

SASTRA UNIVERSITY (INTERNATIONAL STUDENT)

DIRECTOR

PROF. TOMAS ALUJA BANET

COORDINATOR

ALBERT OBIOLS VIVES

Bachelors Degree in Computer Science and Engineering

Facultat informática de Barcelona (FIB)

Universitat Politécnica de Catalunya (UPC) - BarcelonaTech

June, 2015

யாதும் ஊரே யாவரும் கேளிர்

தீதும் நன்றும் பிறர்தர வாரா

நோதலும் தணிதலும் அவற்றோ ரன்ன

சாதலும் புதுவது அன்றே, வாழ்தல்

-கணியன் பூங்குன்றனார்

To us all towns are one, all men our kin,

Life's good comes not from others' gifts, nor ill,

Man's pains and pain's relief are from within,

Death's no new thing, nor do our blossoms thrill

-Kanian Poongundranaar

Abstract

Enhancing the Learning Analytics platform for secondary schools with new indicators: Design and development of new indicators.

Learning analytics is a novice branch of data science has a lot of emerging applications. Just like any other data science project, it majorly involves statistics, computer science and data visualization. The main goal of Learning analytics is to provide conclusion and interesting observation on the learning behavior of students' by using the data from a learning platform. The inLab carries out research on the MOODLE based Agora platform of school students trying to understand their learning behavior. Thus this project aims at development of a dashboard containing the Access logs of all the students showing the login time stamps that may be used to derive interesting conclusions operated at various levels of drill downs implemented as a set of filters. The project also has another orthogonal that aims at the development of a motivation index that computes the motivation value for each student. For the calculation of motivation index, in this TFG carried out 4 indicators namely Forum Access Rate, Resilience Level, Effort level and a modification to the Agility Rate are devised. The effort level was proposed by myself. The above mentioned 4 indicators were designed and developed and integrated with the already existing platform in this part in addition to providing the functionality of the various graphs.

Resumen (Spanish)

Mejora de la plataforma de aprendizaje Analytics para las escuelas secundarias con nuevos indicadores: Diseño y desarrollo de nuevos indicadores.

El análisis del aprendizaje es una rama de la nueva ciencia de datos tiene una gran cantidad de aplicaciones emergentes. Al igual que cualquier otro proyecto de ciencia de datos, se trata mayormente de estadística, ciencias de la computación y visualización de datos. El objetivo principal del análisis del aprendizaje es proporcionar conclusiones y observaciones interesantes sobre el comportamiento de aprendizaje de los estudiantes "mediante el uso de los datos de una plataforma de aprendizaje. El inLab lleva a cabo investigaciones sobre la plataforma Ágora basada en MOODLE de estudiantes de las escuelas e institutos tratando de entender su comportamiento de aprendizaje. Así, este proyecto tiene como objetivo el desarrollo de un cuadro de mandos que contiene los registros de acceso de todos los estudiantes que muestran las marcas de tiempo de inicio de sesión que se pueden utilizar para derivar conclusiones interesantes mediante varios niveles abstracción de implementados como un conjunto de filtros. El proyecto también tiene otro ortogonal que tiene como objetivo el desarrollo de un índice de motivación que calcula el valor de la motivación para cada estudiante. Para el cálculo del índice de motivación, en este TFG realizado 4 indicadores tasa de acceso foro, nivel de resiliencia, nivel de esfuerzo y de una modificación a la agilidad Tasa estén concebidas. Los 4 indicadores fueron diseñados y desarrollados e integrados en la plataforma existente en esta parte, además de proporcionar la funcionalidad de los diversos gráficos.

Resum (Catalan)

Millora de la plataforma d'aprenentatge Analytics per a les escoles secundàries amb nous indicadors: Disseny i desenvolupament de nous indicadors.

L'anàlisi de l'aprenentatge és una branca de la nova ciència de dades té una gran quantitat d'aplicacions emergents. Igual que qualsevol altre projecte de ciència de dades, es tracta majorment d'estadística, ciències de la computació i visualització de dades. L'objectiu principal de l'anàlisi de l'aprenentatge és proporcionar conclusions i observacions interessants sobre el comportament d'aprenentatge dels estudiants "mitjançant l'ús de les dades d'una plataforma d'aprenentatge. El inLab porta a terme investigacions sobre la plataforma Àgora basada en MOODLE d'estudiants de les escoles i instituts tractant d'entendre el seu comportament d'aprenentatge. Així, aquest projecte té com a objectiu el desenvolupament d'un quadre de comandaments que conté els registres d'accés de tots els estudiants que mostren les marqués de temps d'inici de sessió que es poden utilitzar per derivar conclusions interessants mitjançant diversos nivells abstracció d'implementats com un conjunt de filtres. El projecte també té un altre ortogonal que té com a objectiu el desenvolupament d'un índex de motivació que calcula el valor de la motivació per a cada estudiant. Per al càlcul de l'índex de motivació, en aquest TFG realitzat 4 indicadors taxa d'accés fòrum, nivell de resiliència, nivell d'esforç i d'una modificació a l'agilitat Taxa estiguin concebudes. Els 4 indicadors van ser dissenyats i desenvolupats i integrats en la plataforma existent en aquesta part, a més de proporcionar la funcionalitat dels diversos gràfics.

Acknowledgements

எந்நன்றி கொன்றார்க்கும் உய்வுண்டாம் உய்வில்லை
செய்ந்நன்றி கொன்ற மகற்கு

Who every good have killed, may yet destruction flee;
Who 'benefit' has killed, that man shall ne'er 'scape free

-Thiruvalluvar

First and foremost I am very thankful to my guide, Prof. Dr. Tomas Aluja for his continuous support in the development of the project. I would be considered ungrateful if I don't thank my mentor Mr. Albert Obiols Vives for his constant support and periodic review that helped project scale greater heights. I should thank Prof. Dr. Maria Ribera for her valuable comments and support that helped in getting things done on time. This project has been done under the framework of the research project, Reference: TIN2010-46790-P. I take immense pleasure in expressing my heartfelt gratitude to my beloved Dean Dr. P. Swaminathan and our associate deans Dr. A. Umamakeswari, Dr. N.Sairam and Dr. K.S. Ravichndran of SASTRA University for letting me take part in this Research Internship at this excellent institute. I also thanks Dr. M. Sridharan for coordinating us with great sincerity. I should thank my home institute for having given us an international exposure. I also thank the inLab team for setting me the conducive work place. I also take immense pleasure in expressing my gratitude to my friends Pau, Jordi, Ivan and Pranathi for giving me excellent suggestions and help whenever needed in the course of this project. At this juncture I also extend my thanks to my GEP tutor Ms. Jasmina Berbegal for her timely help in providing excellent suggestions with the preliminary report. All this would not have been possible but for the support from my parents, I thank them for being a constant source of support and care. I thank the almighty for his love and blessings for helping me complete my project successfully.

CONTENTS

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

This chapter gives the introduction to the project under study, Enhancing the Learning Analytics Platform for Secondary Schools: Design and Development of New Indicators, its context in a very high level manner and the various stakeholders. The way in which the document is structured is also briefly described in this part.

1.1. Context

Before describing the context of the project it is very important to define what Learning Analytics is. Learning Analytics is a branch of Data Analytics that deals with the study of learning behavior of students from a learning platform. For this project the supporting Learning Platform is the MOODLE based Agora. The Agora is one of the most widely used Learning platforms in secondary schools of Catalonia. The data obtained from the Agora is used to mine out useful and interesting information that can be used to characterize the learning behavior of students under study. Learning analytics encapsulates the various fields: data mining, data analytics, data modelling, and educational data mining and sentimental analysis into one to provide different dimensions in analytics. The project developed by inLab shifts the whole view of education into a different one. The project yearns to learn from the digital traces of the students.

This project is done with the collaboration of UPCNet and the Educational department of the Generelitat de Catalunya. Only a pilot version of the project is done and on successful completion of the project, it may be scaled to all the schools of Catalonia.

Before understanding the Initial state of the platform the various important aspects of this project are described. Learning Analytics is a branch of data science and thus just like any other data science project this also involves the following activities. The adjoining figure gives the overall flow of the project providing a high level view of the project. It begins with the identification of the problem to data visualization. The other various methodologies that were used are described a little later in the document.

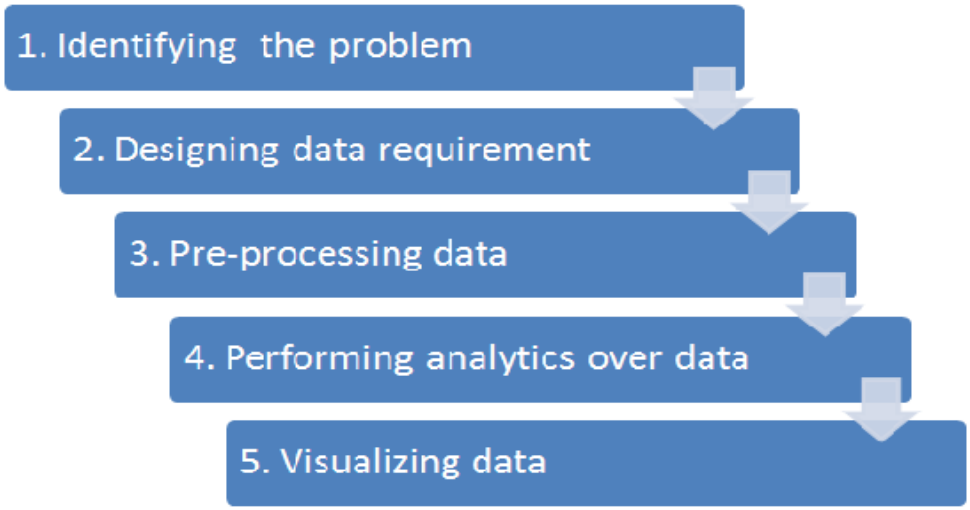

Fig. 1.1 Data Analytics Project Process flow. Adapted from "www.pingax.com" (May, 2015)

Given the various steps of a general data analytics project, the same steps were adhered to in the completion of the project. The first few weeks of the project involved keen study of the data and the various databases available for our analytics. The other details are described in the later part of the document.

This project on Learning Analytics by inLab also aims at measuring the motivation as a function of a system of indicators. My TFG has also contributed to the development of some of the indicators that were instrumental in the measurement of motivation. The measurement of motivation was not carried out by me in my TFG but was done by the inLab's Learning Analytics team.

In the next section the various stakeholders of the project are described and their various roles are discussed.

1.2. STAKEHOLDERS AND USERS OF THE SYSTEM

Stakeholders refer to a person or a group of person who are directly or indirectly affected by the product, here the Learning Analytics tool.

Thus the various stakeholders of the systems are summarized below.

- UPCnet and inLab FIB
- The inLab Participating team
  - Jordi Casanovas
  - Pau Vila
  - Harihara Subrahmaniam Muralidharan (Myself).
  - Pranathi Mylavarapu
  - Ivan Vukic.

- Prof. Thomas Aluja Banet, Prof. Maria Ribera, our project advisor and Prof. Albert Obiols, our project co-director at inLab FIB.
- The teachers of the secondary schools, they are the direct users of the system developed.
- The students of Catalunya, they are affected as a result of the product.
- The Education department of Catalunya. The project is carried out in accordance with them.
- Directors of the secondary schools.

1.3. Main Objectives Of the TFG

It is very apt at this juncture to mention the work carried out by me in the inLab FIB's Learning Analytics project.

a. **Data Handling and Data wrangling for the design of a 24 x 7 timeline.** In this project of inLab, we were asked to develop a chart to show the login activities of the students, the 24 x 7 timeline. I handled the backend activities in development of the dashboard. The actual work is described in detail in the implementation section.
b. **Design and Development of new Indicators.** As a part of my TFG I also developed a bunch of indicators that were used to measure motivation. The detailed list of indicators and their design and implementation is specified in the later sections.
c. **Integration.** The artefacts developed as a part of the project should be integrated with the existing platform and tested for consistency.

The results of these two activities are incorporated into the Learning Analytics platform already existing which will then be integrated with the Agora.

1.4. The Learning Analytics project at the inLab.

The Learning Analytics project in inLab started last year with the master thesis of Miriam Ramirez, student of the DMKM masters, who proposed the first design of the project. The initial architecture proposed by Miriam in her thesis provided the foundation for the various developments the project has seen. She initiated the ETL process which was further enhanced by the inLab Learning Analytics team. As a part of her thesis she developed 4 indicators namely,

1. Percentage of Accesses.
2. Number of Accesses.
3. Time to first access
4. Time spent on the activity.

These 4 indicators were mainly visual indicators.

The adjoining figure represents the initial architecture proposed by Ms. Miriam Ramirez. A few modifications were made by the inLab team.

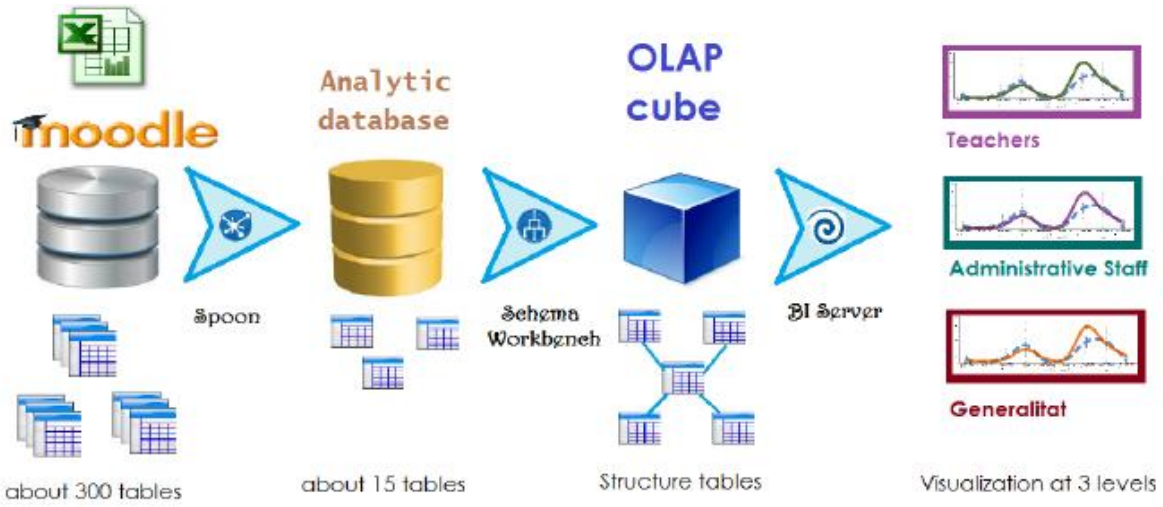

Fig. 1.2. Initial Architecture of the Learning Analytics platform

The first proposal was implemented by the inLab team (Jordi Casanovas and Pau Vila) from the start of the actual course 2014 and 2015. From February 2015 I joined the team to improve the visualization by adding a new tab to the learning analytics dashboard and implement of new indicators that measures motivation. My work in the project runs in parallel with the work done by Pranathi which altogether defines a new milestone for the Learning Analytics inLab project. It is also worth mentioning that the project was well complemented by Ivan's work on the theoretical justifications of motivation.

1.5. Report Structure

This section describes how the report is structured. This gives an overall process flow of the project giving the necessary theoretical foundations to the project.

Chapter 2 lays a theoretical foundation to the project giving a glimpse of the various technologies used in the project and their implications. Chapter 3 discusses the current state of the art and the various practical implications of the project. Project management is detailed in the chapter 4. The chapter 5 elucidates the design and implementation of the project. In the project, I had taken up the job of development of the various indicators, thus the algorithms used to compute the indicators are described in the chapter 5. Finally in the chapter 6 the conclusions along with the learning outcome and the scope for future work is presented. The document also contains an Appendix that gives the footnotes on the various codes developed as a part of the project to enable the students to take this research project in the future.

CHAPTER 2

State of the Art

This chapter discusses the state of the art of Learning Analytics as a branch of data analytics and also cites references of learning analytics in Literature and how the inLab's project and my TFG is different from the others. Having understood the context of the project and its stakeholders, it is high time to discuss the state of the art**.**

2.1. State of the Art

The project that is undertaken, as a part my undergraduate final bachelor thesis, is a developing branch of data science. This being a novice science the inLab are one of the fore runners of the field. The project requires a very good understanding of the very term Learning Analytics, which is described clearly by Tanya Elias. According to which, it is the application of business intelligence to academic data and studying the learning characteristics of students. The Learning analytics is a very personalized study and it has no general process. The literature defines three terms in general: Learning Analytics, Academic Analytics and Educational data mining. Each of the terms specified appears to be the same but we find that they are different only on very close introspection. The difference between the terms learning analytics and Educational data mining is available in the paper by M.A.Chatti (2012) and Ryan S.J.D. Baker paper Educational Data Mining and Learning Analytics (2013), according to which Educational data mining deals with the application of higher order statistics to the data and they are more result oriented whereas Learning analytics deals with the analysis of the students' ability to learn.

The project that the inLab aim at developing is one of its kind. It has drawn inspirations from the LEMO project and this actually triggered interest to work on this project. The project aims to develop a product for the secondary school students.

The project aims at performing Learning analytics on the secondary schools students' data. This deals with designing parameters that aim at capturing the learning characteristics of the learner. The major task in the project is to identifying indicators. The indicators should be feasible and they must be obtained from the data. Maren Scheffel (2014) defines a few indicators. The indicators described in the paper is not sufficient and may not for our need. Any general assumption needs to be avoided and there by try to make reasonable conclusions by observing the data.

The inLab's project narrows down to the measurement of motivation for which specific indicators are required. Motivation and its indicators are described in the next chapter, the inlab has come up with its own set indicators. In my TFG I have developed some of the indicators mentioned in the next chapter.

Initial Context. The current platform developed by the inLab Learning Analytics team is a web based portal. The portal was developed as pilot version. Some of the schools that are active in using IT in Barcelona is chosen for the pilot version of the project. The MOODLE logs of the students using the platform from these schools is collected and the project is carried out. The platform initially showed 4 plots, Percentage of accesses, Number of accesses, Time to first Accesses and total time spent. On completion of my TFG, one more plot to the dashboard has been added, which shows the Access Logs and some summary plots. In addition the platform is planned to be more enhanced with a set of indicators that aid in measurement of motivation and visualisations of the same.

The initial context and the platform details and the various information about the presently existing graphs were studied from Miriam Ramrez Munoz's report who worked in the inLab's Learning Analytics project in the previous year. A very clear knowledge on the existing platform was very important because this was very instrumental in understanding of the existing architecture and the data cube that is formed. This helped in forming the system that is currently existing.

The TFG aims at enhancing the already existing Learning Platform, developed by inLab FIB, a pilot version. To meet the objectives specified earlier in the Chapter 1, the following activities were performed. This required an extensive study of the MOODLE framework and its databases. Not only that the existing platform was studied with a deep understanding but this also required an understanding of web based analytics. As a result interactive visualization techniques was studied and the concepts specified by Scott Murray (2013) was studied and practised. This used the d3.js library. Later the same chart was made more interactive with Highcharts.js.

I specialized in the backend work. Thus some theoretical underpinning of client and server side programs functioned was required. They were studied and later implemented to obtain the graphs by making suitable queries on the data obtained by the ETL. The theory needed for this is mentioned in the next chapter. Thus JSP, Java Database Connectivity and servlets were studied and some programming exercises were also taken. The theory relating to all this is described in the next chapter.

In addition to developing this plot of access logs and a bunch of summary plots, a system of indicators was required to be developed. I developed some of the indicators specified in the next chapter and this required understanding of the databases and the ETL process. The indicators were primarily programmed using R and SQL, hence an understanding of R and SQL was also required. A deep knowledge on SQL was needed as an efficient query was needed to enable faster and efficient performance given the size and the complexity of the data.

Finally in addition to proper understanding of the database schemas, it also placed the necessity to understand the system properly as whatever developed as a part of my TFG needed to be

integrated with the already existing platform. The addition of my module should not hinder the functioning of the existing system.

It is also worth mentioning that the product developed doesn't violate aby governmental policies and it sticks to all the regulations and the laws imposed. It is noted that the Generalitat de Catalonia is also an important stakeholder in this project and they provide us the data.

CHAPTER 3

Theoretical Framework

This section gives a detailed description of the theoretical framework bolstering the project as a whole. This is the most important section of the project because only if the fundamental concepts and definitions governing the project are correct and clear, the project turns up flawless. Thus this section gives a plethora of definitions and concepts that were instrumental in the development of the project.

In this section the various concepts such as the ETL, Web architecture, servlets and JSP, SQL queries, Indicators and motivation is discussed.

3.1. Data mining

Han and Kamber (2006) in their book Data Mining-Concepts and Techniques state data mining as "Data mining may simply refer to extracting or mining knowledge from large amounts of data". The entire process of data mining can be summarized into a series of steps namely, Data cleaning, Data Integration, Data Selection, Data Transformation, Data mining(application of intelligent methods to mine useful information), Pattern Evaluation and knowledge presentation. This is clearly illustrated by the following figure,

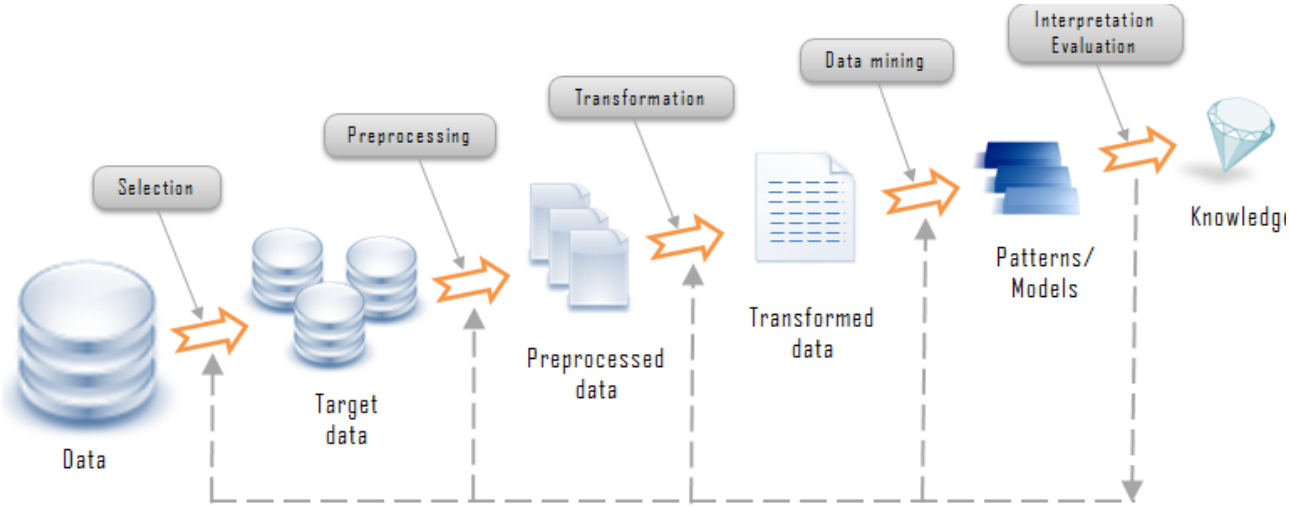

Fig. 3.1. Data mining as a process. Adapted from "zenut.com" (May, 2015)

Though the entire process is not followed in this TFG, the study undertaken is certainly a subset of data mining. The project deals with selection of certain data in the region of interest and

preprocessing them, which is described in the later part of this section. The data* used up in this project, the TFG, is a result of a process called the ETL, Extract Transform and Load.

3.1.1. ETL-Extract Transform Load.

In this section, a precise definition, the importance of ETL and an overview of the tool used by the inLab[1], FIB to perform the ETL is mentioned.

ETL is an acronym for Extract Transform and Loading. These are very important and significant stages in the construction of a data warehouse. Wikipedia (May, 2015) defines ETL as

- **Extract.** Extract data refers to mining of data from homogenous or heterogeneous data sources.
- **Transform.** Transform refers to transforming the data for storing it in proper format or structure for querying and analysis purpose.
- **Loading.** Loads it into the final target (database).

The most common data formats for the Extract phase could be XML, relational databases and flat files. And the Transform phase of the ETL converts the data into required formats which may be same as that of extract load phase. And in the last phase a suitable database is devised to store the data. The entire ETL process can be summarized to the following steps.

The tool used by the Learning Analytics team in inLab FIB is the Pentaho's Kettle tool. The functionalities and features of the Kettle tool are briefed as follows.

Some of the Pentaho's Data Integration (PDI) is described as follows.

The PDI can be used to perform,

- Extract Transform and Load.
- Migrating data between applications and databases.
- Exporting data from databases to flat files.
- Loading data massively into databases.
- Data cleaning.
- Integrating Applications.

The adjoining flowchart very clearly shows how the ETL works.

[1]The ETL was not done by me in my TFG. It was carried out by Jordi Casanovas, a team member in the Learning analytics Project by inLab.

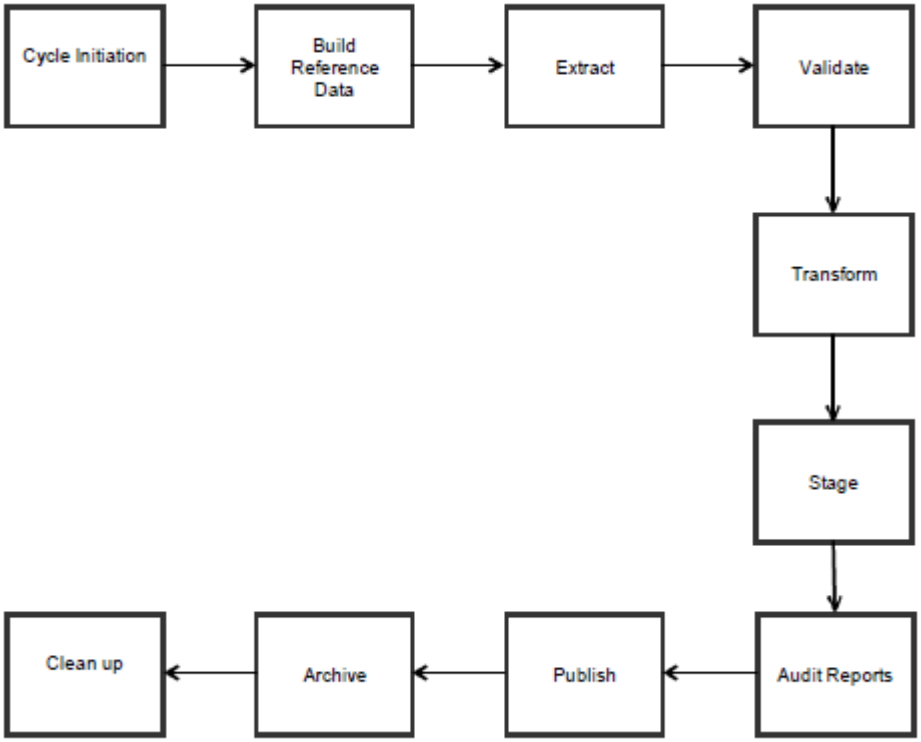

Fig. 3.2. ETL as a process.

The ETL is performed as a series of jobs and transformations.

A transformation is a bunch of jobs that aim at performing a particular function. A transformation involves the following steps.

1. Create a transformation.
2. Construct the skeleton of the transformation using steps and hops.
3. Configure the steps in order to specify their behavior. A steps is a minimal unit inside a Transformation. Each step is designed to accomplish a specific function, such as reading a parameter or normalizing a dataset. A hop is a graphical representation of data flowing between steps.

In the context of the project, the MOODLE data is stored in the form of relational databases. They are massive relational databases. Performing analytics on such huge databases is not very easy and time consuming. Thus the larger databases are broken down into simpler databases which is easier to work with. Hence the ETL tool Pentaho reduces the complex databases of the MOODLE into simpler relational tables, which is easier to perform analytics.

This process is a very important and a crucial process because this process ensures the correctness of data as this phase is important for further analytics. The data obtained from this phase is visualized and a few indicators are derived as part of my TFG. The next section describes another very important theoretical foundation of the project.

3.2. Web Architecture

The product developed as a part of the TFG involves integrating this with the web. The entire project was also developed in the web platform. Thus it is of utmost importance to understand how a web program functions. This section describes the generic web architecture and its relevance to my TFG. It also describes the programming tools instrumental in completion of the first half of the project.

3.2.1. MVC Architecture

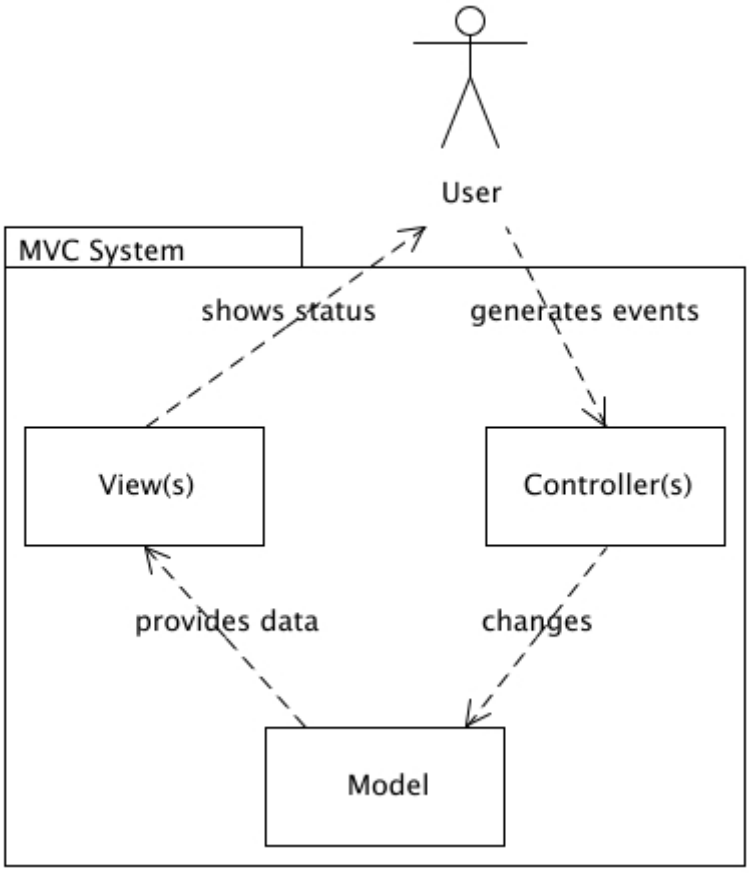

Fig. 3.3. MVC Architecture. Adapted from "Best-Practice-Software-Engineering.ifs.tuwien.ac.at" (May 2015)

This above adjoining figure illustrates the MVC architecture of any user interface. This also shows how the user interacts with the system. This general architecture is used in general by many web developers. The main components of the architecture are model, views and controllers.

- **Model.** A model stores the data that is retrieved by the controller and displayed in the view. Whenever there is a change to the data it is updated by the controller.

- **View.** The view requests information from the model that is used to generate an output representation to the user.
- **Controller.** A controller can send commands to the model to update the model's state. It can also send commands to its associated view to change the view's presentation of the model.

The Model, view and controller are only logical entities. This doesn't mean a view cannot generate events or a controller cannot show status. It is just convention to maintain the functionalities of the components. This lays the foundation many of the web architectures. Though this was initially used for graphical design, it is currently adopted by many web programming languages.

The TFG was mainly developed with java. Thus java servlets were written at the backend to perform majority of the computations and they were connected to the front developed in JavaScript. Servlets were very important because they connected to the database and were instrumental in retrieving the necessary results. The Java web architecture was mainly chosen because first, it is easier to develop and deploy, second, it perfectly adheres to the MVC architecture thereby making the development activity very easy.

In the next section, the various programming technologies such as the JSP and the Servlet technology and their important methods are discussed.

3.2.2. Programming tools

The simple programming model of the system can be schematically expressed as follows,

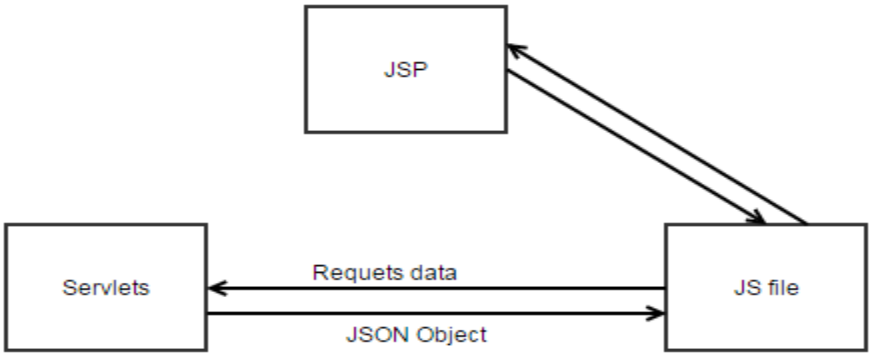

Fig. 3.4. Simple programming model of the system.

This figure represents the overall flow data and process in the TFG. The JSP acts as the view and the servlets act as the controller in the system. The servlet file is always executed in the server side on the other hand the java script is executed on the user's browser. Thus any

calculation and complex operation is always restricted to the servlet to avoid any complex computation at the browser and security purposes. Another major programming tool was the AJAX (Asynchronous JavaScript and XML). The need for AJAX is described in the later sections of the same chapter. The following sections describes what a servlet and a JSP is and their differences.

3.2.2.1. JSP- Java Server Pages

JavaServerPages (JSP) is a technology that helps software developers create dynamically generated web pages. This is a very useful programming tool when it comes to the web because, it allows the developer to program in java within the web script. To include a java code fragment the code needs to be enclosed within the <% and %> tags.

This comes in handy when there is a bit of computation that needs to be carried out in java. Thus for an instance, connecting to a database. This cannot be achieved by any web languages such as HTML or JavaScript. In general to perform backend activities like this a programming language like java is needed. To embed java code in a web code a JSP is used. But the following point is worth noting: putting all java code directly in the JSP is ok for simple applications, but overusing this feature leads to a spaghetti code that is not easy to read and understand. When there is too much of computation that needs to be performed, it should be done using Servlets, which is described in the next section.

A servlet is not totally different from a JSP because all JSP is cast into a Servlet before execution, as all the web scripting languages are interpreters and java needs to be compiled, which needs to be done before execution. Thus JSP is a simple way to embed a java code within a HTML or HTML like web scripts.

The following architecture describes the Lifecycle of a JSP page.

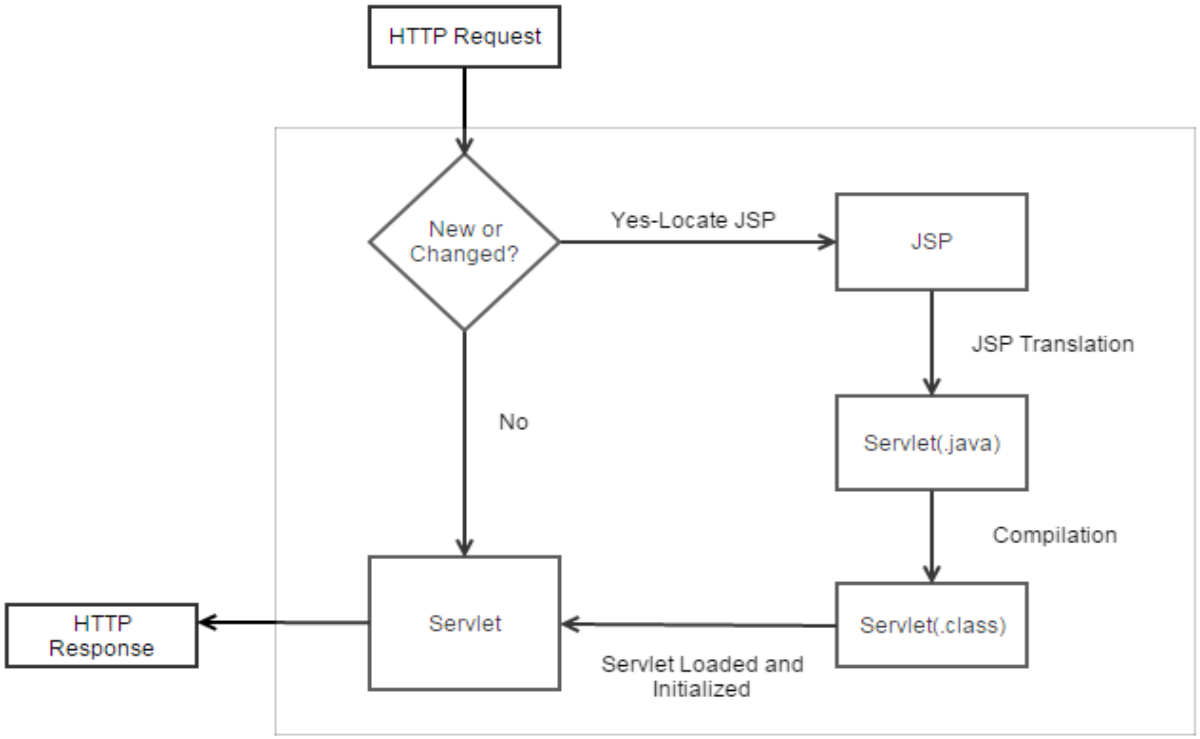

Fig.3.5. Lifecycle of JSP.

3.2.2.2. Servlets

A java servlet is a java programming language that extends the capabilities of a server. They are very common and most often used for

- Process or store data that was submitted from an HTML form.
- Provide dynamic content such as the results of a DB query.
- Manage state information that doesn't exist in stateless HTTP protocol.

To deploy and run a servlet a web container is used.

The web container is responsible for managing the life cycle of servlets, mapping a URL to a particular servlet and ensuring that the URL requester has the content access rights.

In java, javax.servlet defines the expected interactions of the web container and a servlet. Servlets are defined as a part of the web application in several entries in the J2EE standards. The servlet definition is made in the web.xml file. The first entry under the root servlet element defines a name for the servlet and specifies the compiled class that executes the servlet. The main aim in defining the servlet is to transfer the computational load into the server's side. Thus as a rule of thumb it can be said that all database transactions are restricted to the servlet program.

At this juncture it is worth mentioning how the java servlets handle requests. The requests are usually through the RESTful methods, namely HTTP (Hyper Text Transfer Protocol). There are 6 HTTP methods that are most commonly used. They are GET, PUT, POST, DELETE, HEAD and CONNECT. The GET and the POST are the most widely to request data from the server. In other words when the client wants some information from the server or when the client wants to use the server then either the get or the post method is used.

The fundamental difference between the GET and the POST method is described below. In the GET method, the parameters are appended to the URL and sent along with header information. This does not happen in case of POST. In post the parameters are sent separately. Since most of the web servers support only a limited amount of information to be attached to the headers, the size of the headers should not exceed 1024 bytes. The POST doesn't have this constraint.

Java offers two methods namely the doGet() and doPost() which allows interactions with the servlet very easy. These methods are contained in the HTTPServlet class. These two methods are worth mentioning as the entire application development requires proper functioning of these two methods properly.

The syntax of these two methods is described below:

```
void doGet (HTTPServletRequest Request, HTTPServletResponse Response)
    throws IOException, ServletException;
```

Thus whenever a request is of GET type this method is executed. Similarly the syntax of doPost method is given by,

```
void doPost (HTTPServletRequest Request, HTTPServletResponse Response)
    throws IOException, ServletException;
```

Having got a comprehensive idea about servlets and how servlets can be realized and executed in java, the next section describes how a GET request from a JavaScript can be translated into a doGet() method of the Servlet program. It is also worth mentioning that the Java Servlet is registered with the web.xml file giving the servlet name and it's URL.

The next section describes the fundamentals of AJAX and jQuery methods are instrumental in connecting the servlet's doGet() and doPost() method from the frontend i.e. the JSP/JS segment of the program.

3.2.2.3. AJAX-jQuery

AJAX expands to Asynchronous JavaScript and XML. This framework when implemented with jQuery provides a very helpful set of functions to make the webpage very interactive. It allows webpages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This is of certain interest in the project because the browser creates an event and the event is translated into a GET type request which awaits a response from the server. The response is usually a JSON document which needs to be processed and painted as graphs in the

webpage. It is not desirable to have the page load the data one by one. Thus AJAX is used to asynchronously get the data from the server and display its contents once the page is ready. The AJAX taps the functional capabilities of the jQuery.

In context to my project, the web based dashboard that was created used a lot of graphs, an Access logs plot and a bouquet of summary plots based on the data. The access logs plot required a lot of input data and on the other hand the summary plots required very few data. It is logical for the summary plots to be loaded even before the Access logs plot to be loaded. Thus the AJAX methods are used to load the data once the page is ready and translate the HTTP request to java servlets methods. The following methods are worth mentioning at this juncture.

```
$.ajax ({
        url: url,
        data: data,
        Success: function({}),
        datatype: datatype
        });
```

The above method is used to initiate a GET request to a servlet. The URL is set to the servlet URL with the required parameters. Another very important method is the

```
$(document).ready (function ({

        });
```

This method runs only once the entire DOM (Document Object Model) is fully loaded and ready. The entire backend was handled using Java, Servlets, JSP and AJAX and jQuery. Another very important aspect is how the data was handled. The data was handled using JSON documents. The next section gives a very brief idea of what JSON is and why this data model is chosen.

Now having described the roles of JSP, Servlets and AJAX the overall architecture of the system can be put in a nutshell as above.

3.2.2.4. JSON

JSON expands to Java Script Object Notation. The data retrieved from the database is usually a ResultSet which is converted into JSON documents by the server and passed to the JavaScript by the servlet. JSON is chosen because it easier to work with in the JavaScript parts and the front end.

JSON is nothing but a set of key value pairs and the required data is formatted to the following way.

```
"Glossary":
 {
    "Title": "example glossary",
    "GlossDiv":
```

```
        {
                "Title": "S",
                 "GlossList":
                  {
                        "GlossEntry":
                         {
                                "ID": "SGML",
                                "GlossDef": "para": "A meta-markup language ",
                                "GlossSeeAlso": ["GML", "XML"]
                         },
                        "GlossSee": "markup"
                 }
        }
    }
```

The data that needs to be painted is formatted into JSON and suitable libraries are used to paint the graphs. This section completes the theoretical foundations required on the programming tools used for the first half of my TFG.

3.3. Structured Query Language

In the previous section the programming tools and the architecture that was used in the completion of the first half of the TFG was described in detail. In this section the theoretical foundations of the database technology is described. The Database model that was used was the relational database and MySQL was used to query the database. This section describes the various types of queries and their relevance to the TFG.

The first half of the project involved plotting the access logs of students and a few summary plots. I did the backend work of the display of the charts and the second half of the project involved design of indicators which primarily consisted of extraction of suitable data from the database tables. The algorithm and the foundations of the indicators are described in the sections to come. This sections briefs about the type of SQL queries and its relevance in the TFG.

SQL can be broadly categorized into the DDL (Data Definition Language), DML (Data Manipulation Language) and Database Control Commands.

3.3.1. Data Definition Language (DDL)

The Data Definition Language (DDL) is used to define new table schemas or alter the already existing ones. They usually take the name of the entity that needs be added or deleted to the already existing/new schema as an input and performs the action specified. The most common queries that were used as a part of the development of the project were the

- **CREATE TABLE**- This command is used to create a table to the specified database. It takes the columns and their data types as their input and creates the tables accordingly.
- **ALTER TABLE**- This command is used to alter the already existing table schema by adding a new column. This takes the column that needs to be added to the schema and its datatype as an input and alters an already existing schema.

- **DROP TABLE/COLUMN**- This command is used to delete the table schema or a particular column from an already existing database. This cannot be reverted. A dropped table cannot be reversed.

3.3.2. Data Manipulation Language (DML)

The Data manipulation language of the SQL is used to perform operations with the data in the database tables. They were the most used in the development of indicators. They are usually used to manipulate and perform data analytics. The most commonly used DML queries are described below,

- **SELECT-** This query is used to select a subset of tuples from a table based on a condition specified by the WHERE clause.
- **UPDATE-** This query is used to update an existing table just by modifying the values of a certain column. This goes with the SET keyword.
- **INSERT-** This keyword is used to insert a new tuple into the table.

A combination of the above two sets of queries were used to create special indicator tables and populate them suitably. Another very important group of queries worth mentioning is the joins. Joins are used to join two tables based on a key.

JOINS

There are 4 kinds of joins namely the inner join, the left outer join, the right outer join and the full join. The 4 joins are explained below. Let us consider two tables A and B. The intuition behind these group of queries is that the tables are treated as mathematical Relations. These queries were useful in selecting the data required for the indicators.

- **INNER JOIN**. The inner join selects only those tuples common between tables A and B based on a certain condition usually that the Key of A matches the Key of B, i.e. inner join is $A \cap B$.
- **LEFT OUTER JOIN.** The Left outer join selects all the rows of the Left table (A) and those columns that match in the right table (B). The missing values are replaced with NULL. Left outer join mathematically is A – B.
- **RIGHT OUTER JOIN.** The Right outer join selects all the rows of the Right table (B) and those columns that match in the Left table (A). The missing values are replaced with NULL. Right outer join mathematically is B – A.
- **FULL JOIN.** The full join performs the Cartesian product of the tables. Mathematical interpretation of Full join is A x B (A cross B).

Having got an overview of the programming tools, in the next section the statistics and the more logical theoretical foundations of the project is described.

3.4. Importance of Indicators and measurement of motivation

Before describing the indicators of motivation that is developed as a part of my TFG, it is of utmost importance to describe why indicators are important at this juncture. Let us for an example consider a tool that is used to monitor the health conditions of a person. The health conditions can be monitored by a variety of parameters, for an instance Blood Pressure, body temperature, previous medical history etc. Thus if a person is healthy or not can be answered by assessing these parameters. But in this case, the inLab's Learning Analytics project aims at measuring the motivation of students'. Motivation cannot be directly measured but can be made a function of an array of indicators. Hence, it is of a necessity to develop those indicators that characterize the motivation of students. In my TFG I have proposed one indicator and developed some of the indicators proposed[2].

The first step in any analytics project is to identify the objective of the analysis, the next being identification of suitable indicators and features that aids in reaching the objective of the project, here motivation. Thus development of indicators is a crucial step in this process. Only after this step comes the other intelligent data mining methods. In this project some of the indicator and their design and implementation are discussed. In the next section the various indicators that were developed in my TFG are defined and the foundations of motivation are described in the sections to follow.

Indicator design is a very creative activity and it needs to be done very carefully, by making proper and meaningful assumptions. Indicators are statistical features that is obtained from the data which can be used to characterize the data. In the learning Analytics context it is worth mentioning the following points.

- Indicators rely on monitoring of the learning actions and the learning context.
- Indicators have to adapt according to the learner's goals, actions, performance and history as well as to the context in which the learning takes place. In other words Indicators should be correct and should capture the sense of the entire data.
- Indicators are responses to learner's actions or to change in the context of the Learning process, where the response is not necessarily immediate.

In essence Indicators identify and capture the traits of motivation from the data and can be used to represent motivation as a function of these indicators.

[2] Only one of the 8 indicators was defined by me. The others were defined by Ivan Vukic, inLab Learning Analytics team member. I designed and developed some of them.

3.5. Notes on Motivation

This section deals with the definition of motivation. This also gives an overall picture and the theoretical foundations that gave rise to the definitions of indicators mentioned and defined in the next section.

Defining, structuring, explaining and measuring motivation were the topics of interest for many researchers over a several decades. Starting with pioneers like Abraham Maslow, Victor Vroom, Frederick Herzberg, Clayton Alderfer, Stacy Adams as well as the others, who offered different theories of motivation and therefore different perspectives of the same problem. Contemporary researchers talk more and more often about the necessity to measure motivation with Ryan Baker, Ayelet Fishbach and Maferima Toure-Tillery as one of the leading authorities in the field.

At this juncture it is apt to introduce the definition of motivation, its behaviors and a few observations.

Definition. Berhenke et al (2011) summarize motivation definition in an elegant phrase: "Motivation is that, which activates and directs behavior towards certain goals." Moreover, Gage and Berliner (1984) describe motivation as the intensity of behavior, the direction of behavior, and the duration of behavior.

Structure. According to the literature (Chelladurai 2006, Scholl 2015), motivation can be decomposed into three major components, the ones regarding activity, persistence and intensity.

- **Activational** motivation refers to a part of a motivation linked to initiate a behavior. This is motivation to start.
- **Persistentional** motivation refers to a part of motivation linked to effort to move toward the goal even though the obstacles exist. This is motivation to persist.
- **Intensifying** motivation refers to a part of motivation linked to the concentration and vigor that goes into pursuing a goal. This is motivation to stake of one's own effort.

Conclusions on behavior. This section makes some characteristics remarks on how motivated and unmotivated people behave, which is tapped well to create and design indicators.

Always having other priorities. Procrastinating. (They don't want to start.) Prolonging. (Weak intensively of work, a lot of voluntary interruptions). Bad emotion associated while working. Boredom. Negative perceptual bias. The task is perceived more difficult than it is.

On the other hand this is how motivated people behave. It is a priority. I want to do this first. Now. I want to start now. Quick. I want to finish now. Good emotion associated while working. Excitement fulfillment. Positive perceptual bias. The task is perceived easier than it is.

Characteristics. Motivation, like intelligence, cannot be directly observed. Instead, motivation can only be inferred by noting a person's behavior. Extracted from the literature, the overall characteristics of motivation can be summarized as the following:

- **Complex phenomenon**. Complex structure, complex interconnections with the other phenomena.
- **Intangible**. We have to measure it like intangible. Cannot be observed directly. We don't actually observe a motive; rather, we infer that one exists based on the behavior we observe. Nevid (2013)
- **Dynamic**. It changes over time, and those changes can be extreme. Short life time.
- **Personal**. Psychological concept. Internal feeling. Strong individuality (different for different individuals).

The necessities in measuring motivation.

- **Motivation has to be externally measured**. Self-reported measures of motivation is an approach where people are asked, in obvious or less obvious ways, to rate their motivation level. However, as psychologists David C. Mc Clelland and John W. Atkinson argued, although one can be indeed motivated, he or she does not have to be conscious of their own motivation. In fact, one does not necessarily has to have conscious understanding of its own psychological state. Thus, this approach can potentially capture only the conscious part of motivation while neglecting possibly large part of it.
- **Motivation is measured indirectly**. As motivation is intangible psychological construct one has to use indicator or indicators to estimate its level. This means that indicators are measured directly and motivation is estimated using those indicators. Learning results are used as an indicator of learning motivation. However, Romainville (1994), Bessant (1997) and Chen (2004) found out that there is also a correlation between learning strategy and learning results. According to the theory of self-regulated learning and research of Wang et al (2008), both learning motivation and learning strategy have direct effects on learning results. Therefore, using only learning results as an indicator of learning motivation is wrong.
- **Motivation is measured in relative terms** that is compared to something else. It can be compared to its own previous or subsequent levels, to motivation in a different goal state, to motivation of different people etc.
- **Motivation has to be measured constantly**. Measuring motivation as a stable trait is not accepted in this work.

3.6. Measuring Motivation

Based on these definitions, characteristics and constraints, a system of indicators was developed to track motivation, where motivation is a function of seven indicators and can be formulated as following:

$$M = f(x_1, x_2, x_3, x_4, \ldots\ldots, x_n)$$

Where n is the number of indicators.

Moreover, the linear correlation between motivation and the indicators was tested. And the model took the form of:

$$M = \sum_{i=1}^{n} \beta i X i$$

Here M refers to the "motivation index" and the X_i refers to the ith indicator and the β_i refers to the coefficient of the indicator, defining its importance. This β_i is computed by performing PCA on the big tables formed by the indicator. (Supplementary variables are student, date and course). Motivation is calculated for each student, per day, within each course.

$S = \{s_1, s_2, \ldots, s_s\}$	s – number of students
$D = \{d_1, d_2, \ldots, d_d\}$	d – number of days
$C = \{c_1, c_2, \ldots, c_c\}$	c – number of courses

Now in my TFG, the indicators are extracted from the suitable databases and designed that aids in the measurement of motivation. The correctness of the indicators are evaluated only after all the indicators are obtained and the motivation is measured from the data. The next section discusses the various indicators developed in the inLab Learning Analytics Project and the implementation section describes the indicators developed along with their algorithms. A matrix as shown below is constructed for each course.

	DR	CR	PR	AL	RL	EL	PL	CL
$s_1\ d_1$								
$s_1\ d_2$								
…								
$s_1\ d_d$								
$s_2\ d_1$								
$s_2\ d_2$								
…								
$s_2\ d_d$								
…								
$s_s\ d_1$								
$s_s\ d_2$								
…								
$s_s\ d_d$								

Fig. 3.6. Model table for indicators of motivation.

3.7. Indicators Proposed and developed as a part of the inLab's Project on Learning analytics

In this section the system of indicators developed as a part of the inLabs's learning analytics project is described. The adjoining table explains the various indicators and their statistical interpretations long with their implications. The implementation details are described in the chapter on implementation and design.

S.NO	NAME OF THE INDICATOR	DEFINITION	ADDITIONAL INFORMATION	STATISTICAL DEFINITION
1	Delivery rate	This indicator reflects percentage of pending obligatory tasks a student has completed during the day. Reflects Performance.	Obligatory tasks include homework, assignments and hotpots.	$\frac{\#fulfilled\ tasks}{\#pending\ tasks}$
2	Curiosity Rate	This indicator reflects percentage of non-obligatory tasks (without deadlines) a student has complete during the day. Reflects Performance.	Downloaded lectures, links and resources.	$\frac{\#fulfuilled\ tasks}{\#pending\ tasks}$
3.	Peering Rate[3]	This indicators reflects the number of times a student has accessed the forum in a day. Reflects Performance.	Forum activity.	*Accessed->1* *Not accessed->0*
4.	Agility Level[3]	This indicator reflects the time a student takes to access the activity for the first time. F(time first access) Reflects Speed.	Agility=calculated for each activity of a given subject daily. Agility level=average of all activities of 2 weeks. If no data Agility Rate is NA.	*(Date of delivered-date of announced)=x* *x<1 -> y=1* *1<x<7 -> y=f(x)* *x>7 -> y=0*

[3] These indicators were designed and developed by me as a part of my TFG.

5.	Resilience Level[3]	This indicator reflects the percentage of today's activity from a particular subject, done in a sequence. Reflects Persistence.	All the activities of a given subject. All the interruptions in a 2 hour window.	$\frac{\#Longest\ Sequence}{\#Activities}$
6.	Engagement Level	This indicator reflects how active a particular student is on a given day, in comparison to his best performance in the last 14 days.	All the activities of a given subject.	$\frac{\#activities}{\#the\ last\ 14\ days\ peak}$
7.	Competitive Level	This indicator reflects how active a student is on a given day, compared to the most active student on that day.	All the activities of a given subject.	$\frac{\#activities}{\#most\ active\ student}$
8.	Effort Level and Cognitive index[4]	This indicator measures the Effort made by a student to submit an activity. This is measured on a 15 day window.	All the obligatory tasks of a given subject.	*Effort Level= (#Accesses+1)x(#Attempts +1)* *Cognitive index=(#Accesses)/(#Attempts)*

Table 3.1. Indicators

This concludes this chapter. With all the necessary theoretical foundations, the implementation and the design is described in the chapters to come.

[4] This indicator was defined by me.

CHAPTER 4

Project Management

This chapter deals with the various management aspects of the project. This section describes the scope, the goals of the project and the various requirements that the project enforces. Though a traditional project management methodology may not be applicable in this setup, an agile methodology may be well suited. An agile methodology is the one in which the requirements keep flowing in on the go of the project and the project slightly adjusts itself to account for the various requirements. Periodic meetings are conducted in order to discuss the progress of the project. This does not mean the quality may be sacrificed but it is very important to ensure quality in each phase of the project.

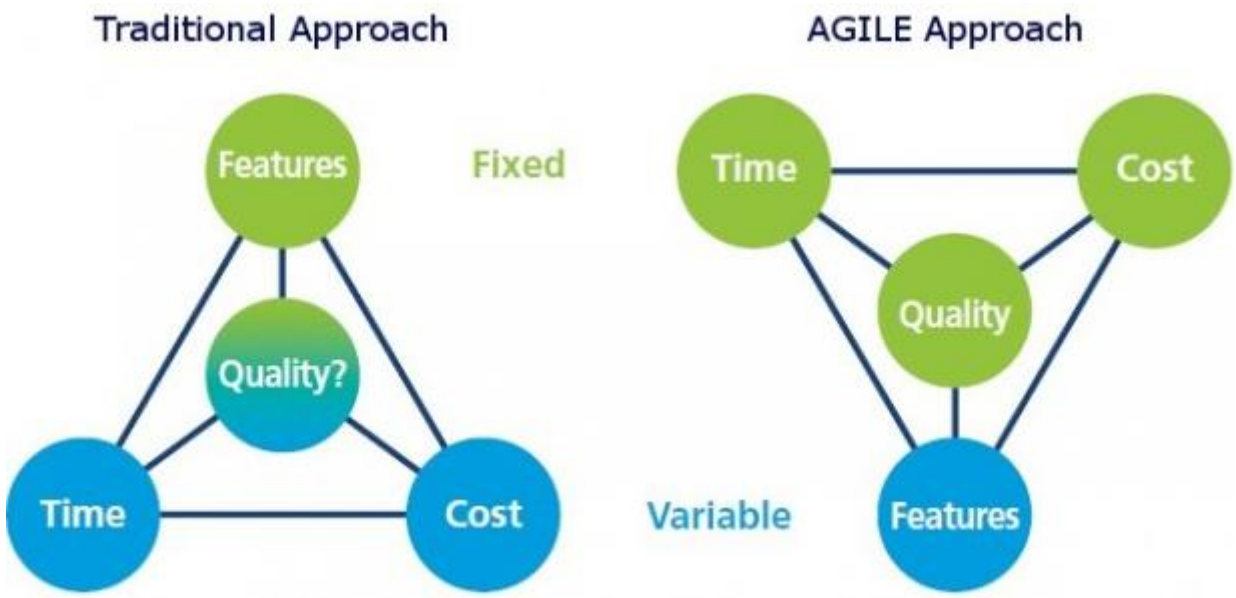

Fig.4.1. Traditional Vs Agile approach. Adapted from “www.entrepreneur-ideas.org” (May 2015)

The above figure show how the cost and the time factor varies for traditional and the agile model. Thus the agile model was chosen as it accommodated all the requirements of the project. The following section describes the various goals, the requirements of the project.

4.1. Scope

The Project aims at identifying new indicators from the data available and enhance the already existing Learning analytics platform. The data is obtained from the students’ MOODLE logs and they are stored in the MOODLE data base. This data is used to perform the analytics.

Thus, the objectives of the project can be summarized as follows

a. **Data collection and wrangling**. The students' data is stored in massive databases which needs to be reduced to simpler databases.
b. **Design of a web based platform to show the various indicators developed**.
   The second important factor that the project delivers is a very friendly web page that gives visualization of the various indicators developed as a part of the project.
c. **Enhancing the Platform with new indicators.** The platform can be enhanced by designing new indicators. Indicators are special statistical features that characterize the learning behaviour of a student.

The scope of the project is summarized as follows

1. **JUSTIFICATION**: Once the project is completed, the same can be used to study the learning rates and the learning behaviour of students. The project also visualizes the results of the data into pleasing and informative graphs which makes it easier for the teachers and the analysts to draw inferences. The most important aspect of the project being the availability of indicators and the development of friendly and informative graphs.

2. **PRODUCT SCOPE:** The outcome of the project is a product by itself. The product is a platform that enables school teachers and managers to track the learning behaviour of students. It provides a clean backend for data extraction. As a result, a complete product is obtained which can be used to perform learning analytics on secondary school data.

3. **METHODOLOGY:** The project as stated earlier deals with data wrangling. To perform the data wrangling the Pentaho's kettle tool is used. This is the most important step in any data analysis project. The next step would be to use the data from simpler data bases to design a web page that visualizes the indicators in visually appealing charts and graphs. A very important step would be to design new indicators. This would use the result of ETL. These indicators is used for measurement of motivation.

   Java scripting and HTML was used for the front end and Servlets technology was used for the back end. The eclipse ide was used for the development and the Apache tomcat sever for hosting the website during the project. This being a research project, the main aspect in the project is to develop good indicators for learning analytics. No tool is used to monitor the project but have regular weekly meetings organized by professors to whom the progress is shown.

   The entire project can be explained as follows. After the data wrangling the simpler databases were used to query. A tool called the database browser is used for this purpose.

The result from the query is obtained as JSON objects and the JSON documents are then used for the visual representation of the indicators.

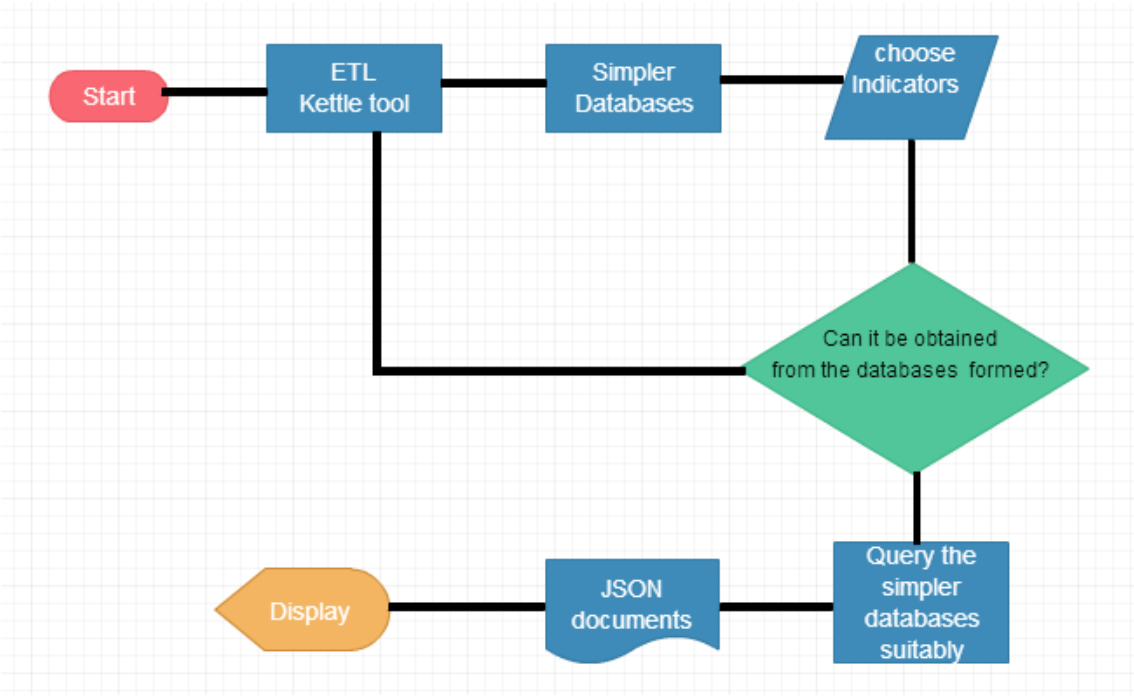

Fig. 4.2: Methodology

4. **ACCEPTANCE CRITERIA:** The product is to be widely used by secondary school teachers. The main aim of the system being able to extract the data, analyse and develop a friendly interface for the teachers that aid them in understanding the student's learning behaviour better. In addition, the data is interpreted on various factors and the results are summed up. On successful development of the product is tested with the actual data and the results are verified with the secondary school teachers and any change requested is appended to the already developed software.

5. **DELIVERABLES:** As mentioned earlier the outcome of this project is a product by itself. The product is a web based platform that performs all the activities from the extraction of data to visualization of the results.

6. **CONSTRAINTS:** The project's success mainly depends on the quality of the data that is obtained. This being a data analysis project the outcomes are better when the data is sufficient there by avoiding unnecessary assumptions which, leads to better promising results and inferences. Another constraint is the timeframe. All the phases of the project needs to be done within the deadline.

4.2. Project Planning

Based on the observations the various steps in my project is defined as follows,

1. 24x7 Timeline design

This refers to a dashboard design that captures the MOODLE accesses logs of students and visualize them as a graph. This data when visualized can be used to study indicators already existing. Just like any software Project this system has two components, a front end and a back end. The back end deals with the extraction of data from a common database and the front end deals with visualization of the data extracted from the back end.

This part of the project involves the following resources. The Eclipse IDE is used for the project development and programming. Thus, the major requirement would be the following software resources namely, Eclipse, the database browser, the apache tomcat server. The hardware resources that this phase of the project requires is a desktop computer with the windows 8.1 operating system installed.

A computer software designer who designs the various parts in the webpage and the computer engineer who directs the project are the prime human resources that's required.

2. New Indicator design

This phase of the project concerns with observing the data for new indicators and design them suitably to extract sense out of the raw data. The new indicators can be obtained by studying the data thoroughly. This is a very fuzzy phase in the project because it cannot be quantitatively described the number of indicators that we might be able to design in the given time frame. This phase would require the assistance of a Statistician who specializes in designing of indicators. Thus, the Human resources would include an additional statistician.

3. Integration

This is a very important activity of the project because this deals with integration of the already existing platform with the dashboards that were designed. The integrated software is tested for bugs and run with new data and the performance of the new system is measured. This phase of the project also requires the same resources as that of the previous two phases. This is the most important phase of the project, because this involves the compilation of all the parts of software that was built so far. This would require the additional support from the designer.

The following steps are carried out and these are common to almost all the stages of the project.

1. **Requirements:** The requirements are studied that gives a brighter picture on the nature of the software that is intended to build. This is the most important process in the project building activity.
2. **Design:** Design concerns with the analysis of tools that might be required to build the software. The various design paradigms and technologies are studied and the best that suits the project is chosen. For an instance my region of interest lies in the design of the back end activity. The various technologies and paradigms are either using JSP or servlets. The two technologies are compared and the servlet technology is chosen and the software design is carried out after that.
3. **Construction:** The design pattern that was chosen is implemented on the project and this mainly deals with the coding of the software.
4. **Testing:** The testing is an activity in which the software developed is tested with various test data sets. This activity helps us identify bugs and rectify the same.
5. **Deployment:** Deployment deals with putting the software developed so far in the actual platform and see how the software reacts to the actual environment.

The last phase of the project doesn't contain requirements and design phases because this just involves the integration of the various software components developed thus far. This doesn't require a specific design pattern and doesn't require a requirements gathering too. The most important aspects of this phase is the testing. This is to test how the software behaves on load.

4.3. INITIAL GANTT CHART

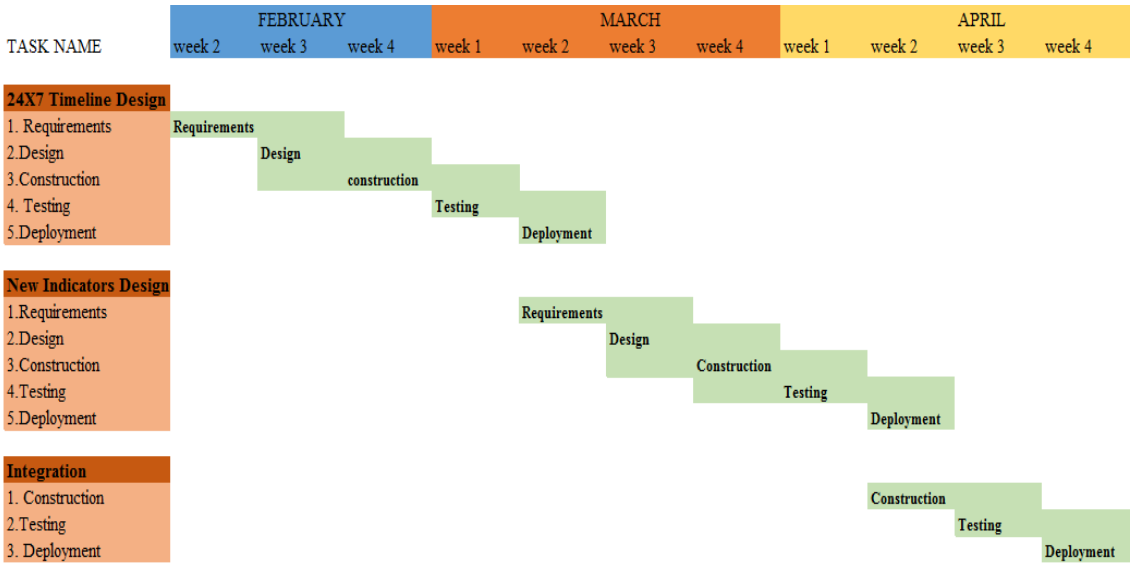

Fig.4.3. Initial Gantt chart.

The Explanation of the project planning and the deviation from the original schedule

This Gantt chart shows the activity only till April. This is only a tentative plan and this shows only the first few steps that the project takes. Based on the further developments a few other steps and additions can be made. This is because this being an incremental project the exact steps are not known yet. The meetings and the documentation work is simultaneously done. The mentor meets us on a weekly basis and present him the deliverable for the week.

Each subsection within the project involves the activities, Requirements, Design, Construction, Integration and Testing be carried out in this order. The phase two of the project doesn't require the first step in the project be carried out. The first step gives an intuitive idea to development of a few new indicators. Additionally, the last stage of the project requires the first two steps to be carried out. This is the most important and difficult step in the project.

The initial Gantt shows activities only till April. Once the timeline is done March, 4 new indicators were developed as a part of the project which were then integrated with the platform. There is change from the initial Gantt because as mentioned earlier the indicator design is a very creative activity. This completely bolsters on how well the data is understood. Thus in the remaining time the indicators were designed and developed.

The various design technologies are studied and the best methodology is chosen. The data is available and the required data is alone extracted from the main database. The construction phase involves the actual coding of the software. The construction and testing may introduce a delay in the delivery of the project because they involve the actual engineering of the software. Similarly, the indicator design is an activity that involves a keen observation of data and this also introduces a delay in the product delivery. Thus, the indicator design involves a lot of mental activity. In case the projects experiences unexpected delay, the integration can still be performed because, the integration is independent of the phase 2 of the project. We aim to complete the project before the deadlines as the requirements have been rightly identified and the project moves at the right pace. The construction activity improves with time. Initially it involves a bit of training and this improves as time goes. The testing activity and the construction are totally inevitable and there is definite need for these two activities. Though they act as the source for a delay they have to be surpassed indicators design involves a lot of mental work over actual work. This being a research project the number of indicators is not proposed. This in turn depends on the quality of data. In case the data is not adequate certain assumptions can be made and the indicators can be identified. Adding to this, based on the data available some data can be generated based on the requirements of the project. When the right assumptions are made a qualitative bunch of indicators may be designed. The final Gantt chart is described in the adjoining figure.

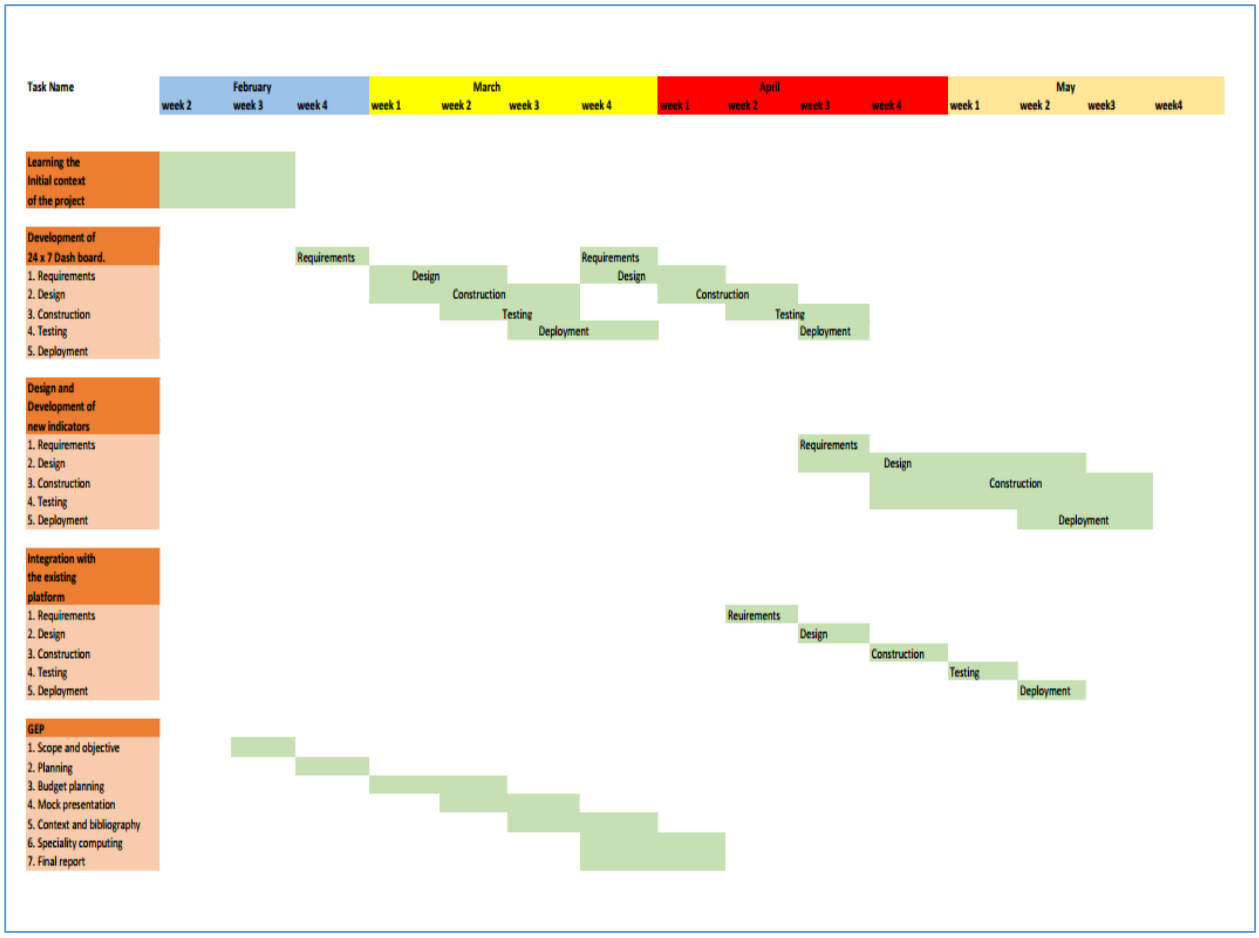

Figure 4.4. Final Gantt chart

4.4. Requirements Engineering

The formal scope of the project can be described by giving an analysis of Functional and Non-functional requirements. Functional requirements describe the behaviour of the system as it relates to the system's functionality. Non-functional requirements elaborate performance characteristics of the system.

4.4.1. Functional requirements

R.1 Access logs

A plot which visualizes the daily login activity of a course drilled down by time: month, week, day or a particular student, a group and required module types in any order. It has three summary plots providing different intuitive orthogonal views of the data visualized.

R.1.1. Daily login activity. A summary plot which shows the part of the week a student or a class is very active which dynamically adjusts according to the various levels of the drill down.

R.1.2. Hourly login activity. A summary plot which shows the most active part of the day for a student or a class which is consistent with other plots.

R.1.3. Density calendar. An actual calendar based display which shows the density of activity for a student or a class on a daily basis, which is in sync with the rest of the plots.

R.2 Indicators

Indicators are statistical features that summarize and characterize the learning characteristics of a student. All the indicators are ultimately used to compute a new wholesome indicator called Motivation Index.

R.2.1. Delivery Rate. This indicator reflects % of pending obligatory tasks (usually with deadlines) a student has completed during the day.

R.2.2. Curiosity Rate. This indicator reflects % of pending non - obligatory structured tasks (usually without deadlines) a student has completed during the day.

R.2.3. Forum Access Rate[5]. This indicator reflects % of non – obligatory and non-structured tasks (thus without deadlines) a student has completed during the day.

R.2.4. Agility Level[1]. This indicator reflects the time student takes to access the activity for the first time.

R.2.5. Resilience Level[1]. This indicator reflects the % of today's activities from a particular subject, done in a sequence.

R.2.6. Engagement Level. This indicator reflects how active particular student is on a given day, in comparison to his best performance during the last 14 days.

R.2.7. Priority Level. This indicator reflects how much priority on a given day a student gives to a particular subject in comparison to the subject he is most committed to on that day.

R.2.8. Competitive Level. This indicator reflects how active a student is, on a given day, compared to the most active student on that day.

R.2.9. Effort Level[6]. This indicator reflects how much effort a student places in a given window of 15 days.

[5] Designed and developed by as a part of my TFG. The agility rate was enhanced by me.
[6] This indicator was introduced, defined, designed and developed by me.

R-3 Development of application interface

R.3.1. Data visualization. The data analysis performed must be summarized and viewed through interactive graphs or plots displayed in the application. This requires front end development to convert the indicators into graphs.

R.3.2. Dashboard development. All the plots must be contained in a dashboard. Every teacher has personal login credentials to access their respective dashboard in order to view the analytics.

4.4.2. Non-functional requirements

NFR.1 Effectiveness of indicators

The indicators designed as a part of the project should be statistically efficient and correct and produce meaningful results. Not all data available may be very useful, thus, those that are used to develop the indicators should be sensible and meaningful.

NFR-2 Efficiency of Algorithms.

Indicators themselves are algorithms. The best method should be to extract the statistical inference from the available data and the algorithms developed should be scalable with a minimum computational cost.

NFR-3 Intuitive and interactive interface

Data visualization primarily concerns displaying huge amounts of raw data in a simple and an intuitive way to the client. Interaction of the client with the data in real time to interpret data in multiple dimensions is a key requirement of the project. An interface that satisfies these requirements and which has a good user experience has to be designed and developed.

NFR-4 Handling Large Amounts of data

This project involves large amount of data. To give an estimation the project deals with data of around 500 students that currently contains more than one lakh rows in just one table. The table exponentially grows. The tables are updated every day by choosing the required data, the daily interactions, from the MOODLE databases by the ETL process. Thus the software developed should be capable of handling large amounts of data.

4.5. Practical Aspects

This section discusses the various Practical aspects of the project. The main goal of any project is justified and reached only if it is realized into some practical purpose. This applies to the product I have developed as a part of my TFG too. The main qualitative objective is to improve the quality of education in the state. Hence in this section a few usecases are provided identifying the various

actors in the scene and their usecases. This has been developed from a very high level point citing a few instances where it can be useful.

As a remainder the existing platform is enhanced by a new dashboard that shows the Access logs and a few summary plots that is clearly described in the requirements section of this chapter. It is also enhanced with a few indicators to measure motivation.

4.5.1. Main users of the system

The pilot version of this software is mainly developed for teachers. This software currently developed will give some information on the learning characteristics of the students enrolled in the current course of the teacher. Thus one of the main users of the system will be **the teachers.**

Another main user of this system can be **the headmaster or the school manager.** They will have different use of the system, they can view the entire performance of the school as a whole. This may not currently be applicable as only a pilot version of the software is developed.

Another very characteristic user of the system could be the **Psychologists or the "tutor"**. This is because the artefact developed provides information about the learning behaviour of the students which can be used to study the preparedness and analyse the problems that a student potentially faces.

Though **the students are not the direct users** of the software developed at the moment, they are the ones affected as a result of the software developed. So they form one of the most important class of stakeholders. This section may be studied in sync with the stakeholders defined in the first chapter of the report.

4.5.2. Use cases

Ali Bahrami in his book on Object Oriented Systems Design Use cases as follows. "A use case corresponds to a sequence of transactions, in which each transaction is invoked from outside of the system (actors) and engages internal objects to interact with one another and with the system's surrounding". The use case description describes what happens with in the system. It becomes even clearer when the Use cases can be represented as a set of diagrams.

UC-1. Headmaster/School Manager Meeting with the Teachers of a particular class.

Actor- Headmaster, teachers of a class.

In this case the headmaster might not be really interested in viewing all the personalized plots of the students. It is enough if we display just the summary plots and the aggregated indicators that gives a blue print study of how well the class is motivated. This would be some useful information for the school manager. In a nutshell describes how the class has improved as a whole. This can also leave information on how student react to different teachers!

UC-2. Tutor/Psychologist meeting with the student.

Actor- Tutor/Psychologist, student, graphs.

Graphs required. Personalized plots of access logs and the personalized aggregate plots. This enables the psychologist to capture the learning behaviour and offer advice on altering the learning behaviour of the student to improve performance.

Indicators used. All would be really helpful but the Resilience Level and the Forum Access rates hits on how well the student is motivated at the personal level.

UC-3. Parent- Teacher Meeting.

Actors- Parents, Teacher.

Graphs required. Access Logs and the summary plots. A pictorial artefact just like the ones we have developed provides a clearer insight on the student's learning behaviour.

Indicators used. All the indicators may be used but a comparison how the class average is with respect to the student's might prove well effective.

UC-4. Teacher and the Student interacting.

Actor- Teacher and the student.

Graphs Required. All the graphs developed in this project may be used by the teacher to understand how well the student is motivated.

Indicators used. All the indicators may be used as each provides a whole new dimensions.

UC-5. Teacher interacting with the system.

This use case is one of the most useful and interesting one. The system of indicators and the plots themselves leave some comments about the teacher's effect on the class. The class average can be used to say about how well the teacher has reached to the students. This can be a tool for introspection too. This is an indirect usage of the system.

4.5.3. Use case diagram.

Again quoting Ali Bahrami, "A use case diagram is a graph of actors, a set of use cases enclosed by a system boundary, communication associations between the actors and the use cases, and generalizations among the use cases". Now the previous use cases are translated in to the following diagram.

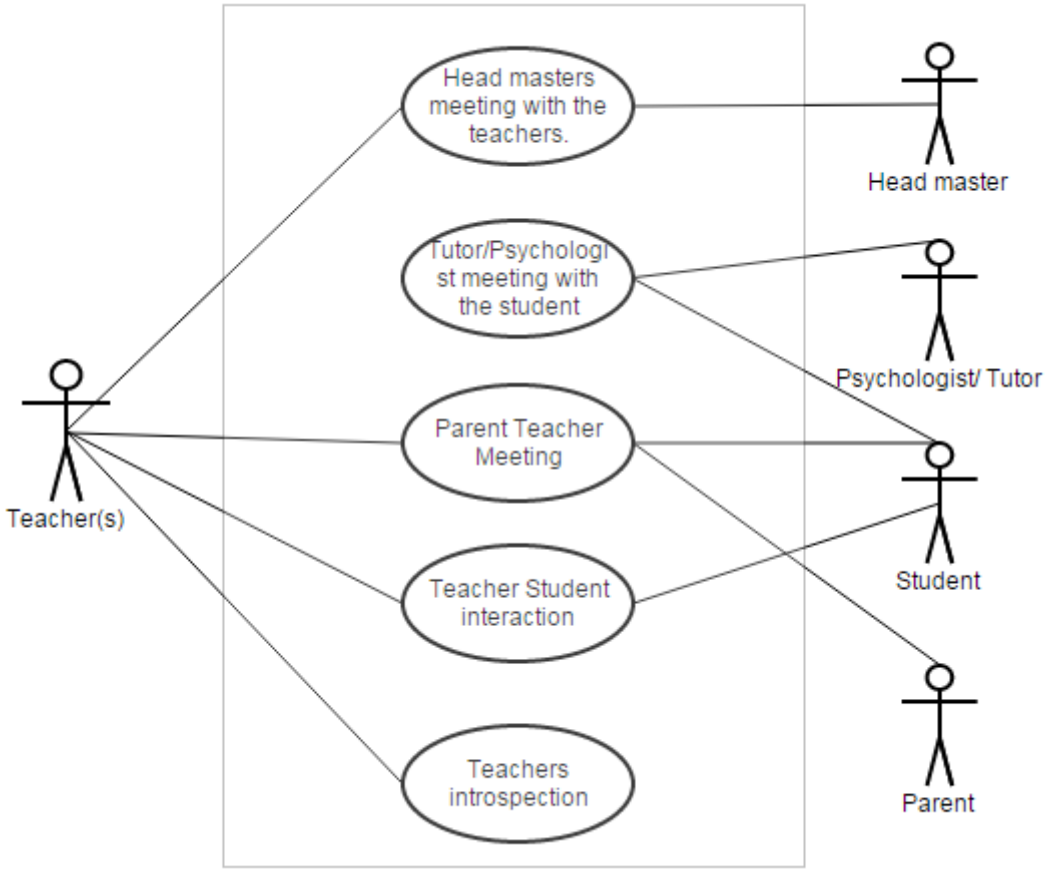

Fig. 4.5. Use case Diagram

4.6. Process Methodology

The project is concerned with data analysis and enhancement of existing platform, so there will be multiple iterations for addition of new features. So it requires continuous planning and execution which will be done at the beginning of each cycle. This methodology is termed as Agile project Management.

Agile software engineering combines a philosophy and a set of development guidelines. The pervasiveness of change is the primary driver for agility. The philosophy encourages customer satisfaction and early incremental delivery of software; small, highly motivated project teams; informal methods; minimal software engineering work products; and overall development simplicity. [1]

Many agile process models have been proposed and are in use across the industry. Among the most common are Adaptive Software Development, Scrum, Dynamic Systems Development Method (DSDM), Extreme Programming, Agile Modeling and Agile Unified Process.

The **Agile Unified Process** (AUP) adopts a "serial in the large" and "iterative in the small" philosophy for building computer-based systems. By adopting the classic UP phased activities: **inception, elaboration, construction, and transition**. AUP provides a serial overlay (i.e., a linear sequence of software engineering activities) that enables a team to visualize the overall process flow for a software project. However, within each of the activities, the team iterates to achieve agility and to deliver meaningful software increments to end users as rapidly as possible. Each AUP iteration addresses the following activities:

Modeling. Models of the problem domains are created. However, to stay agile, these models should be "just barely good enough" to allow the team to proceed.

Implementation. Models are translated into source code.

Testing. The team designs and executes a series of tests to uncover errors and ensure that the source code meets its requirements.

Deployment. Focuses on the delivery of a software increment and the acquisition of feedback from end users.

Configuration and project management. Project management tracks and controls the progress of the team and coordinates team activities.

The process model closely resembles AUP as the requirements keep changing dynamically and have periodical technical reviews every week, where software developed in that period is displayed along with the changes made in the previous weeks. Within the week, the following software development methodology: *inception*, *elaboration*, *construction*, and *transition*.

4.7. Budget Planning

Budget planning is one of the most important phase of the project management. In this phase the budget for the project is analysed. The main aim of this activity is to provide an optimized overall budget for the entire project. The expenditure from various aspects such as software costs, hardware costs, license costs and human resource costs is examined in order to provide a wholesome value. One important factor to note is that the budget that is described in this section is subject to change and it may increase depending on the unexpected obstacles. For an instance when the expected results are not obtained with a particular software, we may have to go in for another software that may incur extra installation and operational charges.

4.7.1. Budget Estimation

In this section I, aim at performing a budget estimation for the software. The overall expenditure is classified into three categories namely hardware, software and human resources. One very important factor that needs to be considered is that only an estimate of the total cost is described. This may vary depending on the systems in use. To calculate the amortization the following factors namely, first the overall life of the hardware or software in use is considered and the project will be completed in 6 months. Hence, the amortization cost comes one eighth of the actual life of the component.

4.7.1.1. Hardware Budget

The hardware budget gives a detailed analysis of the various hardware elements used by the project, its actual and amortized cost.

S.NO	Hardware Component	Useful Life	Total Cost	Amortized Cost
1.	Personal Computers	4 years	1000 Euros	125.00 Euros
2.	High powered Servers	1 year	470 Euros	235.00 Euros
3.	Total		1470 Euros	360.00 Euros

Table 4.1. Hardware budget.

4.7.1.2. Software Budget

The software budget shows an estimate for the various software used in the project along with the estimate of the software costs. It is a myth that the software doesn't get old with time just as a software gets but it wears out with time. Thus, for every software there is a fixed time during which it gives maximum performance. In addition, freeware software and open source software incur no cost.

S.NO	Software Component	Useful Life	Total Cost	Amortized Cost
1.	Microsoft Windows 8.1	1 year	56.49 Euros	56.49 Euros
2.	Eclipse for Java EE Developers	3 years	0.00 Euros	0.00 Euros
3.	Database browser	3 years	0.00 Euros	0.00 Euros
4.	Java JDK and JRE	3 years	0.00 Euros	0.00 Euros
5.	Apache tomcat server	3 years	0.00 Euros	0.00 Euros
6.	Pentaho PDI Kettle	3 years	0.00 Euros	0.00 Euros
7.	Microsoft Word (documentation)	4 years	119.00 Euros	29.75 Euros
8.	Total		175.49 Euros	86.24 Euros

Table 4.2. Software budget.

4.7.1.3. Human Resource Budget

The human resource budget deals with the overall expenditure spent on human resources. There are 5 kinds of people involved in the project each person having a fixed job role in the project. We have a computer Engineer who deals with the project planning and project initiation activity. He mentors the proper functioning of the team and guides the team. The software designer is responsible for design of the proposed software and aims at efficient development of the software. He designs the tools used in the construction phase of the software. The software developer takes care of the coding of the software using the tools planned by the designer. The test Engineer tests the software and checks of the software functions properly and is bug free. The statistician helps in designing the indicators that assist in assessing the learning characteristics of the students. Based on their profile they have different wages.

S.NO	Role	Price per hour	Total hours Worked	Total price
1.	Computer Engineer	50 Euros	80	4000 Euros
2.	Software Designer	35 Euros	160	5600 Euros
3.	Software Developer	25 Euros	100	2500 Euros
4.	Software Test Engineer	20 Euros	100	2000 Euros
5.	Statistician	25 Euros	55	1375 Euros
6.	Total		500	15475 Euros

Table 4.3. HumanResources budget.

4.7.1.4. Total Budget

The following table summarizes the total budget for the project. This encompasses the hardware, software and human resources budget.

S.NO	Nature of Resources	Budget
1.	Hardware Resources	360.00 Euros
2.	Software Resources	86.24 Euros
3.	Human Resources	15475.00 Euros
4.	Total Expenditure	15921.24 Euros

Table 4.4. Total budget.

4.7.2. Linking to Planning Phase

As mentioned in the previous chapter planning and budgeting activities of software project management go hand in hand. It is clearly mentioned in the previous section, the various resources required. The hardware and software are going to be used for the entire course of the project. They recur a perennial budget. But on the other hand the human resources change on the nature of the activity performed. The software designer, the test engineer and the software developer works for most of the time because they are involved in all activities of the project but on the other hand the computer engineer and the statistician don’t work on all phases of the project, the number of hours they contribute to the project is considerably lesser.

4.8. Sustainability

Sustainability is a key factor in any project design. The project is evaluated based on three factors of sustainability namely economic sustainability, social sustainability and environmental sustainability.

4.8.1. Economic Sustainability

This section deals with the economic sustainability of the project. In this document specify the budget specified is the estimation of the project. From the estimation it can be said that this will be the maximum bound on the budget for the project. This takes into account all the factors namely the hardware costs, software costs and human resource costs. It is difficult to do a similar project at a lower cost than this. This is the most feasible and the best budget plan. The budget may exceed the calculations only during unexpected times. If, the plan is adhered to correctly, then it can be guaranteed that the project may be completed at a much lower cost than the estimated ones. Also, the product that is aimed at developing here is tested with all kinds of data and aimed at building a very high quality software which in turn provides a durable software that will not wear out easily. Also, this being a Learning Analytics project, this is done with collaboration with schools which gives better results

Most of the software used in the project is open source which has zero product cost and the only software that incurs cost is the operating software without which the software cannot be developed. The hardware required is nothing but computers that becomes a mandatory part of any project in the present days.

4.8.2. Social Sustainability

The project aims at developing web based platform to perform learning analytics based on the MOODLE logs of secondary schools. This is indirectly going to analyse the learning characteristics of the students and provide a feedback both to the learners and their teachers. This is going to improve the quality of teaching and learning in the state. All this requires is a simple computer connected to the internet. This has very keen social motive and this project when completed is going to improve the standard of learning in schools. Thus, this has a great social responsibility. This is in turn justifies why this project has a great social sustainability.

4.8.3. Environmental Sustainability

From the sections of temporal planning and the budget planning we understand that we have a computer running throughout the project. If, we make an assumption that the amount of energy used by a single computer comes to around 250 watts. And given that we spend 500 hours on the project then the energy expended is 125KW. This amounts to 48.125 kg of CO2. This is indeed a high amount but well within the permissible limits. This can be reduced by reducing the code size which is possible by reusing the already existing code. But the project is actually environmentally sustainable.

4.8.4. Ratings

The ratings for sustainability based on Christian Felber is also described. Find below the sustainability matrix. The various scores for assessment is also clearly mentioned. The project is highly sustainable. In most of the cases it is going to make the project better and it wouldn't worsen the scenario.

sustainable?	Economic	Social	Environmental
Planning assessment	Economic Viability 8:10	Improved quality of Life 8:10	Resource Analysis 6:10
Outcomes assessment			
Risks assessment			
Total			

Fig.4.6. Ratings.

CHAPTER 5

Design and Implementation

This is the most important chapter of the entire report. This completely discusses the work done by me as a part of my TFG in the inLab's Learning Analytics project. The back end for a dashboard that plots the access logs and the very important summary plots was designed by me and in addition to this a set of 4 indicators was designed and developed by me. Of the four indicators I defined one and implemented an important functionality to the indicator developed by Ms. Mylavaruppu Pranathi as a part of her TFG. This section also discusses the design, the algorithms that were developed and the flow of the project. The results and the analysis is also discussed.

5.1. Overall flow of the Project.

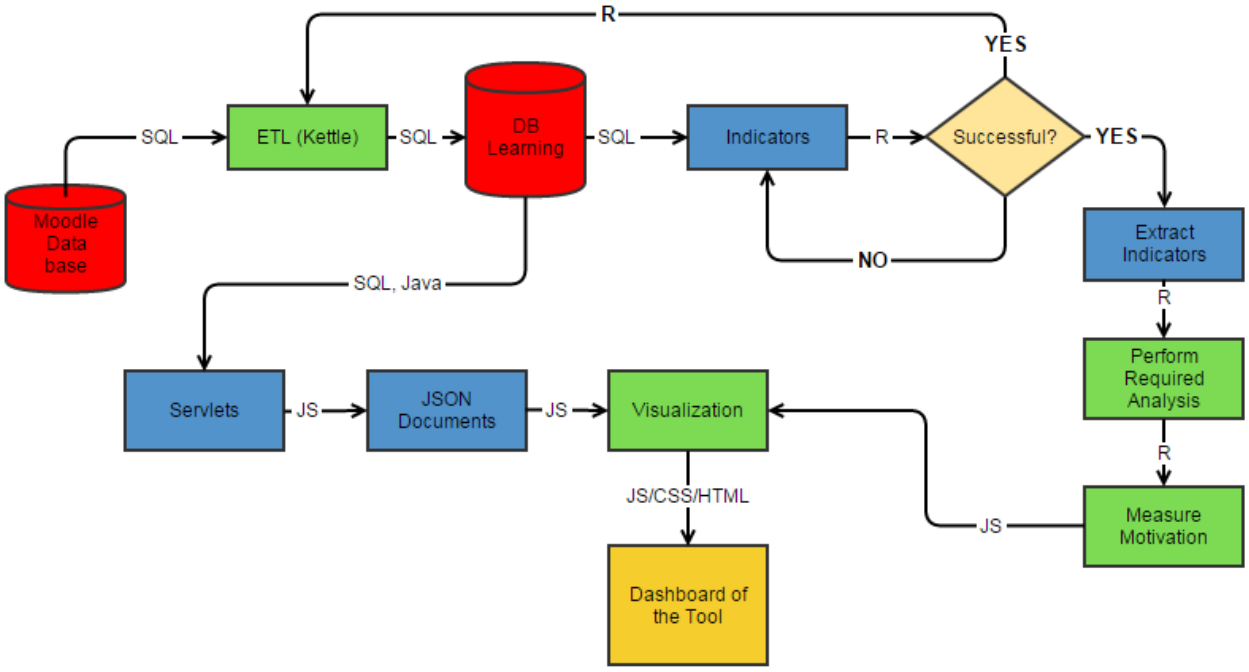

Fig. 5.1. Overall flow of the project

The above figure represents the overall flow of the project. The areas marked red refer to the various Databases used in the project and the ones in blue represents those that I worked on. The Green ones are those that was developed by other members in the inLab's Learning analytics team. The Motivation measurement and the analysis was done by Ivan Vukic. The diagram above specifies in which language the programs were developed. In this chapter the areas done by me are explained. The first half of the chapter explains the design and implementation issues of enhancing the already existing dashboard. The second half of this chapter discusses the various indicators developed and their algorithms, their importance.

Before getting into the design and implementation characteristics, it is worth mentioning that the entire work was carried out on the DBLearning that hosted about 28 tables. The tables were reduced from the huge MOODLE tables and the tables were updated on a daily basis. The ER diagrams are described in the appendix section of this report. They may be referred to for better clarity. The programming tools used in this development process were described in chapter 3, theoretical foundations. In addition the programs for indicator were development was written in R. A bunch of new tables were created to host the newly developed indicators. The design aspects become more clear when this section is read along with the foundations from chapter 3.

The next section covers the fundamentals in the development of the dashboard.

5.2. Design and Development of the Dashboard.

At the outset, a graph representing the access logs of all the students (about 500) was required. The entire backend activity for this was handled by me, on the other hand the visualization was done by an inLab teammate, Pranathi. This plot was necessary to study the login activities. The project at inLab aimed at measurement of motivation of the students. Then it seemed a good idea to study the daily login activities of the students.

5.2.1. Rolling up and Drill down.

The plot was governed by a set of filters namely, course, the students, the group, the date and the module type. On closer introspection it could be very evidently identified that these could be organized as various levels of drill down. At this juncture it is very important to define what drill down is.

Han and Kamber (2006) in their book on data mining define drill down as the process of climbing up from higher level summary to lower level summary or detailed data or introducing new dimensions. The opposite of this process, i.e. moving up to a more high level information from a low level system is rolling up. The filters were just one way of implementing rolling up and drilling down.

Let us consider,

Course id : $\{ C_1, C_2, \ldots\ldots, C_C\}$,
Modules : $\{ M_1, M_2, \ldots\ldots, M_M\}$,
Student S:$\{ S_1, S_2, \ldots\ldots, S_n\}$,
Groups G: $\{ G_1, G_2, \ldots\ldots, G_m\}$. (Each groups is a set of distinct students).
i.e. $G_1 \cup G_2 \cup \ldots \cup G_m = C_i$.
and $\forall\ i, j \leq m$, $Gi \cap Gj = \Phi$.

A course may or may not have groups. The data can mathematically be described as follows. Now this very clearly follows a drill down. The data initially is visualized for the entire class as a whole.

The data can be visualized only for a particular group and then for each student within the group. This very clearly forms a set of drills down. Pictorially this can be represented as follows,

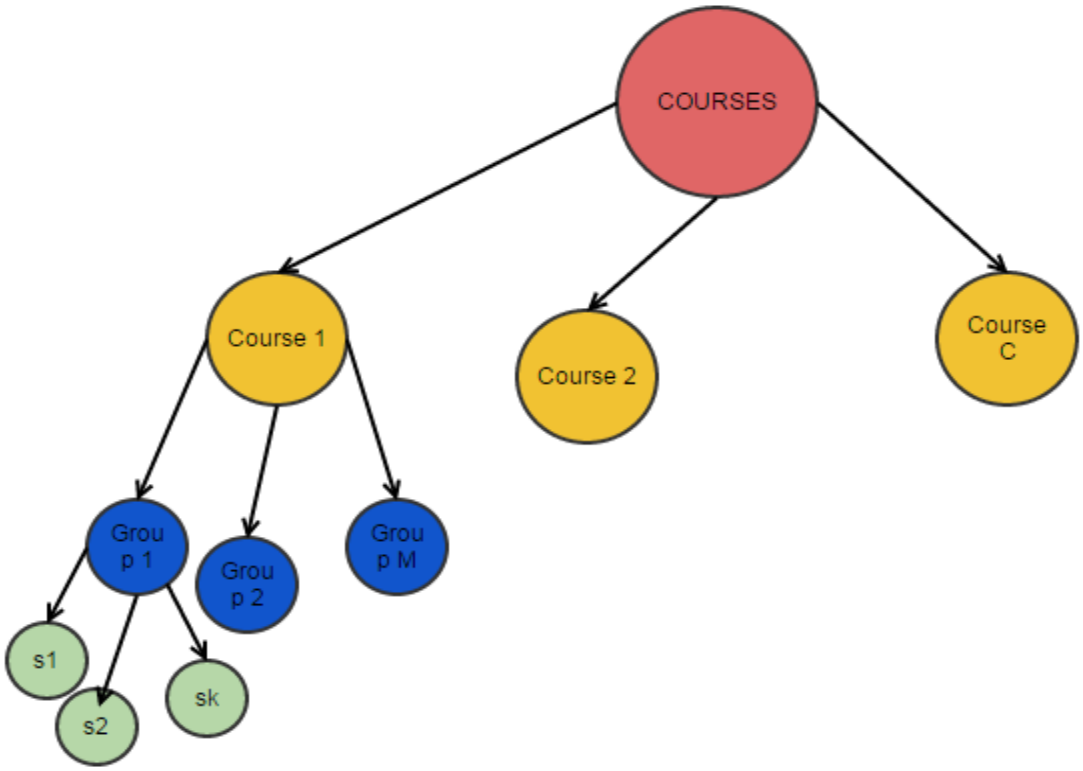

Fig.5.2. Various levels of Drill down

The above figure describes the various levels of drill down. Similarly another possible dimension to view the data is through the time dimension. The data can be viewed aggregated for a month, drilled down by a week, drilled down to a day. Another dimension was viewing through the module type. The modules can further be classified as mandatory or optional.

Having understood how the drill down works and, these were developed as set of filters. Next let us discuss how the plot was developed along with a simple tiny algorithm governing it.

5.2.2. The Access Logs.

The access logs was the first plot to be developed. The X-axis contained the dates, the Y axis contained the various segments of the day. The Y axis was divided into 5 parts, 0 hours-8 hours, 8 hours-14 hours, 14 hours-18 hours, 18 hours-21 hours and 21 hours-24 hours.

A servlet was written that connects to the database and retrieves only those required by the user and casts them into the JSON format required by the front end. It is notable to mention that the front end used the highcharts.js library. Hence the JSON required to be processed. And similarly the front communicated its requests to the backend through the HTTP GET request. The filters were implemented for each courses. So to navigate to the access logs page the course id was the parameter needed.

Now consider the pseudo code for implementing the above specified graph.

Algorithm 1: Printing of Access Logs.

```
1. If  Chosen courseid is NULL
      a. Report error.
      b. Break.

2. Else if chosen courseid != NULL
      a. Set curs  = courseid
      b. If  groupid != NULL
           i.  Set Groupid = Groupid.
      c. End if.
      d. If studentid != NULL
           i.  Set Alumne = studentid.
      e. End if.
      f. Set URL Parameters (curs, Groupid, Alumne).
      g. Create a GET_Request (URL).
      h. Invoke_Servlet ('timeline.java', URL).
      i. If Success
           i.  Data <- Response Text from GET Request.
          ii.  Data <- Parse to JSON (Data).
         iii.  Print_Graph (Data).
      j. Else.
           i.  Console ("Error").
      k. End If.

3. End If.
```

Procedure: Servlet_Timeline (GET (Request_URL))

```
1. Set Curs=Request.curs
2. Set Studentid= Request.Alumne.
3. Set Groupid=Request.Groupid.
4. Connect to the database (DBLearning).
5. SQL<-CreateSQL (curs).
6. If(studentid!=NULL)
      a. Append (SQL, studentid)
7. If(Groupid!=NULL)
      a. Append (SQL, Groupid).
8. Result <- Execute_Query (SQL).
9. Json <- Format_to_JSON (Result).
10.     Write (Response, Json).
11.     END.
```

The above algorithm gives a general outline of the procedure of how the Access Logs was computed. The Backend involved handling the Requests from the user and translating into the URL and invoking the corresponding Servlet. And the servlets were written in java that connected to the database and performed the necessary computations. The "if" statements in these algorithms became active whenever the options was chosen or simply when a select event was triggered. The results were returned to the JS program that invoked the servlet and prints them as a graph. The Final Graphs are displayed at the end of the section.

Note-1:

It should also be noted that the drills down by modules was generated as a 'button event' and those by dates was realized by zooming in through the graph.

It is always important to capture information that is sensible and interesting from data which is the whole point of analytics. Thus the Access Logs was transformed into 3 summary plots that represented the same data in multiple dimensions. The different summary plots were a plot of daily login activity, hourly login activity and a density calendar. For each of these I handled the backend.

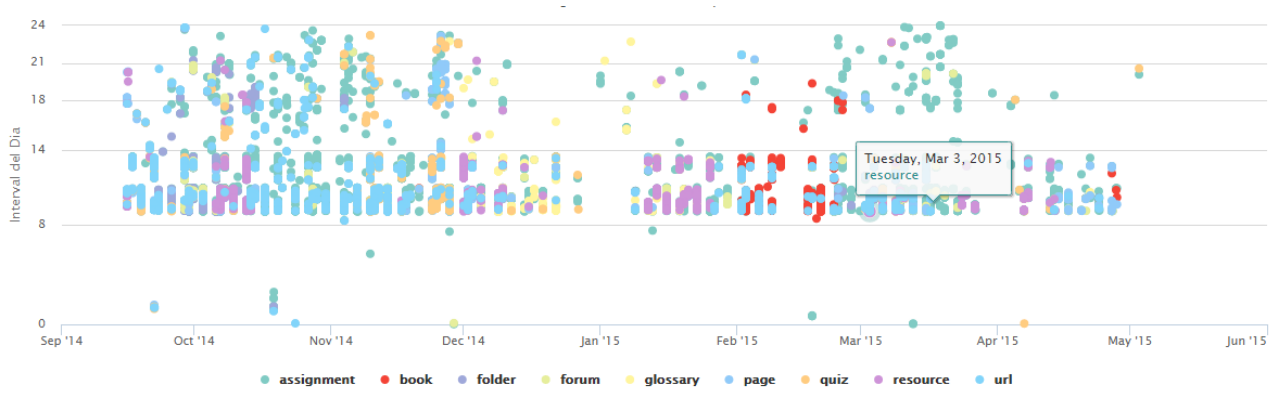

Fig.5.3. Access Logs

5.2.3. Daily Login Activity.

It is even more interesting if there is a chart that says the most active day of the week, for the data that is represented in the Access Logs. This means that a smaller plot that depicts the most active day of the student is visualized that takes the courseid, studentid, groupid, the X-Axis extremes(the time range represented by the access logs plot) and the modules that are selected. That is whenever the Access Logs changes the smaller graph should also change correspondingly.

This graph can be extremely useful, for an instance this can give some information as to which day a student is very active. If he works only during weekends, this could mean that the student isn't a very regular person. This indirectly leaves comments about his motivation.

```
Algorithm 2: Summary plot 1. Daily login activity.

1. If  Chosen courseid is NULL
      a. Report error.
      b. Break.

2. Else if chosen courseid != NULL
      a. Set curs  = courseid
      b. If  groupid != NULL
           i.  Set Groupid = Groupid.
      c. End if.
      d. If studentid != NULL
```

```
          i.  Set Alumne = studentid.
     e. End if.
     f. Date_min=Get_Left_Extreme (Acceslogs.XAxis).
     g. Date_max=Get_Right_Extreme (Accesslogs.XAxis).
     h. If All modules Active
          i.  Set Modules<- NULL Set.
     i. Else
          i.  Modules<-Modules U (Inactive moduleType).
     j. Set URL Parameters (curs, Groupid, Alumne,
        Date_min, Date_max, Modules).
     k. Create a GET_Request (URL).
     l. Invoke_Servlet (‘DailyLoginActivity.java’, URL).
     m. If Success
          i.  Data <- Response Text from GET Request.
         ii.  Data <- Parse to JSON (Data).
        iii.  Print_Graph (Data).
     n. Else.
          i.  Console (“Error”).
     o. End If.

3. End If.
```

Procedure: Servlet_DailyLoginActivity (GET (Request_URL))

```
1. Set Curs= Request.curs
2. Set Studentid= Request.Alumne.
3. Set Groupid= Request.Groupid.
4. Set Date_min= Request.Date_min.
5. Set Date_max= Request.Date_max.
6. Modules = Request.Modules.
7. Connect to the database (DBLearning).
8. SQL<-CreateSQL (curs, Date_min, Date_max, Modules’).
9. If(studentid!=NULL)
     a. Append (SQL, studentid)
10.  If(Groupid!=NULL)
          i.  Append (SQL, Groupid).
11.  Result <- Execute_Query (SQL).
12.  Create_Array(Week,7)
13.  While Result ! = NULL
          i.  Temp<- Compute_day(Result.date)
         ii.  Hash temp to the right bucket and increment
              the bucket by 1.
        iii.  Result<-Result+1
14.  End Loop
15.  Json <- Format_to_JSON (Week).
16.  Write (Response, Week).
17.  END.
```

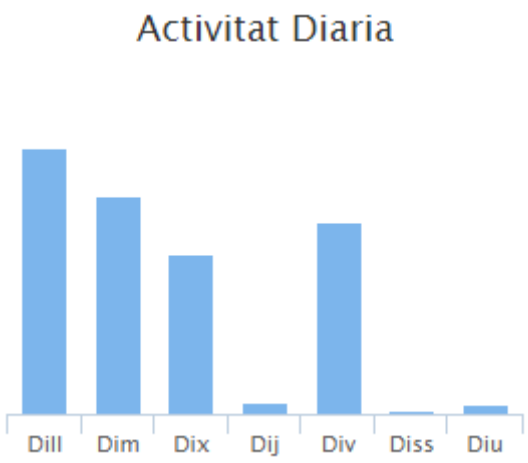

Fig.5.4. Daily login Activity

5.2.4. Hourly Login Activity.

This graph is totally similar to the previous one. This graph represents in which part of the day a student is very active. That is which hour of the day a student/class accesses the portal most frequently. The design is totally similar to the previous plot. So the implementation is not explained. It is very similar to the previous algorithms. This graph should also be consistent with the access logs, in other words whenever the Access logs changes the Hourly login graph should also change accordingly.

The algorithm is not mentioned here as it is totally similar to this previous one. Like for an instance if a student accesses at around 12 Noon, this says that he/she does the work during the break time. This could mean the student is not very motivated to do the work. This could give raise to very serious and interesting observations.

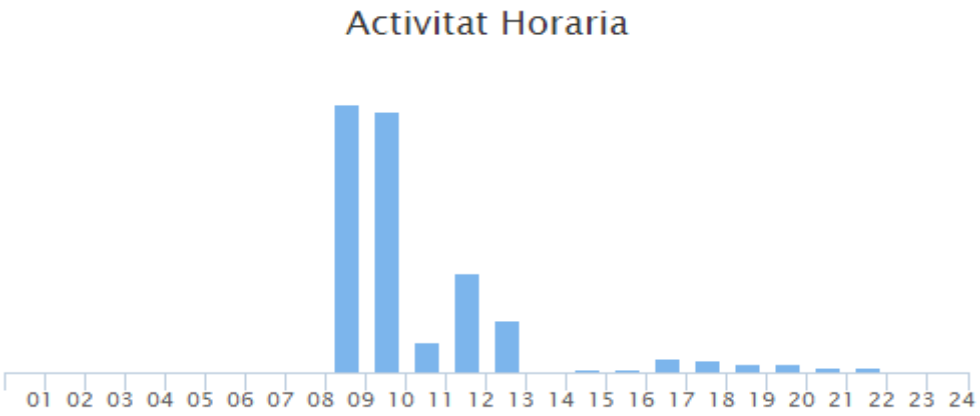

Fig.5.5. Hourly login Activity

5.2.5. Weighted Calendar.

The weighted calendar is another summary plot developed as a part of the TFG. This used the cal heat map.js for the front end. This is a simple intuitive way to see the Activity density of the data represented by the access logs plot. All this requires is the count of activities performed on each day. The calendar also provided two buttons that can be toggled to move forward and reverse directions. Based on the number of activities done on that particular day the colour used to shade the date changes. For an example if 5th of March had 20 activities performed while the 3rd of March had only 5, the 5th of march has darker colour than that of 3rd. This is a very intuitive way to pictorially study the activity density on any given day. This is again a visual indicator of motivation.

All these are in sync with the Access Logs plot. Whenever these is a change in the Access Logs these plots must change suitably. The summary plots convey a lot more information than the other plots. The algorithms is very similar to the previous ones and a similar logic is being followed. Hence this doesn't require a special description.

The graphs are displayed at the end of the section. To make things clear, I only handled the backend.

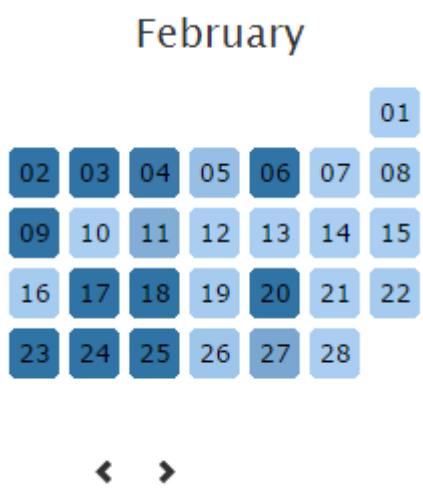

Fig.5.6. Weighted Calendar.

Note-2:

There are three major design considerations,

1. It is of utmost importance that the schema is understood very well, because the proper databases need to be queried.

2. All the four plots were displayed in the same page. The smaller graphs were just a projection of the major Access logs plot. Thus the smaller plots should also represent only the data presented by the Access Logs graph. Hence the three smaller plots should change whenever the Access logs plot changes. This was handled by using proper event triggers taking parameters from the filters and the Access logs plot as illustrated by the algorithm, in the previous section.

3. Though each of the graphs are independent of one another they are hosted on the same page. Hence it is a great necessity that all the graphs are displayed at the same time. It was explained in chapter 3 how AJAX handles requests asynchronously, It is quite evident that the Access logs takes a bit longer to get ready while the smaller ones get ready for display very quickly. Hence they must wait and all the graphs should be displayed at the same time.

The ultimate and the penultimate design issues are handled with the help of proper AJAX methods.

This concludes the first half of this chapter. Having understood how the graphs were designed and implemented, in the next chapter the most important aspect of the project: the design, development and implementation of indicators is described.

5.3. Design and Development of Indicators.

This section deals with the design and development of indicators of motivation. As stated in chapter 3, a bit of theoretical foundation on motivation would give a brighter picture on the need for indicators and their importance. This section further enhances by throwing light on the specific indicators that I developed as a part of my TFG. I have organized this entire section into 4 subsections giving the intuition, the algorithm used, a model simulating the outputs and design issues if any.

In the project on Learning Analytics by inLab I had defined one indicator and designed and developed 3 indicators in addition to modifying the indicator developed by Pranathi. This activity of designing and development of indicators is a very creative and a very stimulating process. This is my most favorite segment of the entire project. The reasons can be summarized as follows,

1. All that is available for this process is a database which hosts about 28 tables. The statistical or mathematical definitions are given, but it is not very clear as to which tables must be queried in order to get the right data. This requires very clear understanding of the tables and their schemas in addition to a very clear definition of the queries.

2. Another reason why this process could be creative is because defining indicators that could measure motivation requires proper understanding of the context of motivation and the information in the tables should be tapped completely in order to produce the best results from what is available.

In this project I developed the following indicators, Forum Access Rate, Resilience Level and The Effort Level. I modified the Agility Rate by writing a piece of code to fill the missing values. All the programs were developed in R. This language was chosen because this was compatible with the ETL tool, namely Kettle. This segment discusses the beauty of the indicator with the algorithm followed and a sample simulation.

5.3.1. Forum Access Rate.

The Forum Access Rate is the simplest indicator. But it is one of the indicators that is very useful. It is quite intuitive enough to think that those students who are more motivated tend to access the forum than those that are less motivated. This was experimentally verified too. It is also under stood that there are more zeros than ones because it sounds logical to argue that it is not expected to access the Forum every day.

The working of this Indicator is very simple. The Database has a table called Accesses that records the various accesses made by students or simply in other words the click details of all the students are stored in this Table. The table is queried suitably. This can visually be realized with the help of the access logs plot by choosing the required course and the student and hiding all the other modules except Forum.

This indicator needed to be computed for each student in each course on a daily basis. Refer the adjoining algorithm to understand how the Forum Access Rate functions. If there is a forum Access by a particular student in a day, 1 is assigned and other wise 0.

$$\text{FARate} = \begin{cases} 1, & if\ Forum\ Accessed \\ 0, & otherwise \end{cases}$$

Algorithm 3: Forum Access Rate.

```
ComputeForumAccessRate()
   1. Begin
   2. For Each Course Ci
        a. Curs <- Course_id(Ci)
```

```
        b. For Each Student Sj ∈ Ci
             i.  Set date_ini <- Max_date in the ForumAccess
                 Table.
            ii.  If date_ini = NULL
                    1. Date_ini<- Beginning date of the term.
           iii.  Set date_final <- Current_date.
            iv.  Loop (date_ini <= date_final)
                    1. SQL <- Query_Accesses(Ci, Sj,
                       date_ini, ModuleType='Forum')
                    2. Result <- Execute_Query (SQL).
                    3. If Result = Φ
                          a. Update <- INSERT (ForumAccess,
                             Ci, Sj, date_ini, 0).
                    4. Else
                          a. Update <- INSERT (ForumAccess,
                             Ci, Sj, date_ini, 1).
                    5. End_If.
                    6. Execute_Query(Update).
                    7. date_ini <- date_ini + 1.
             v.  End Loop
        c. End For.
    3. End For.
    4. End.
```

Simulation of Sample output: Forum Access Rate.

This is how the indicator works. This is just a sample out for 10 days for a course C_1, and 5 students.

Days	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}
S_1	0	0	1	0	1	0	1	0	0	1
S_2	0	0	0	1	0	0	0	0	0	0
S_3	1	0	0	0	0	0	0	1	0	1
S_4	0	0	0	0	1	1	0	0	0	0
S_5	1	0	0	0	0	0	0	0	0	1

Table 5.1. Simulation of Forum Access Rate

The above table is a sample calculation. The values are not original. The same is calculated for all the students and for all the courses every day.

5.3.2. Resilience Level.

This section discusses an indicator namely the resilience level. This indicator is a measure of persistence. This is a very intuitive indicator because it is an evidence of persistence. The persistence to do an activity or a course directly reflects how well a student is motivated. Thus in this indicator the persistence is translated into a measure of motivation per course. The working of the indicator is demonstrated below.

In this indicator the accesses of a student on a particular day is measured. It checks in every two hour window the activity of a student. That is in a given series of activities, the number of breaks

a student takes is computed. The definitions of breaks is described later in this section. The length of the longest such sequence, without any breaks, is computed. For a clearer understanding refer to the algorithm and the simulation of a sample output.

This also has small design issue. This actual efficiency and the performance of this indicator could be justified only if a student has enrolled into more than one course, because the persistence could be checked if the student sticks on to one course before he moves to the next one. In the inLab's project only a pilot version of the platform is created i.e. only one course from each school is used for the verification of the platform.

A student has taken a break, if anyone of the two things happen,

1. The difference between the current and the next activity time stamp is greater than 2 hours.
2. If the current and the next activities belong to different subjects.

Algorithmically, if anyone of these happen the counter is reset. To normalize the count value it is divided by the number of activities a student has accessed in that particular subject. The algorithm for the computation of Resilience Level is described below.

$$\text{ResilienceLevel} = \frac{Length\ of\ the\ Longest\ Sequence\ without\ Breaks}{The\ total\ number\ of\ activities\ for\ that\ course}$$

The resilience level is measure for each course, each student on a daily basis.

Algorithm 4: Resilience Level.

```
ResilienceLevel()

1. Begin
2. For Each Course Ci
      a. Curs <- Course_id(Ci)
      b. For Each Student Sj ∈ Ci
           i.  Set date_ini <- Max_date in the
               ResilienceLevel Table.
          ii.  If date_ini  NULL
                  1. Date_ini<- Beginning date of the term.
         iii.  Set date_final <- Current_date.
          iv.  Loop (date_ini <= date_final)
                  1. SQL <- Query_Accesses(Sj, date_ini)
                     returns courseid, time
                  2. Result <- Execute_Query (SQL).
                  3. Sort (Result, time).
                  4. Temp <- ComputeResilienceLevel (Result,
                     curs).
                  5. Update <- INSERT (ResilienceLevel, Ci,
                     Sj, date_ini, Temp).
                  6. Execute_Query (Update).
                  7. date_ini <- date_ini + 1.
           v.  End Loop
```

```
      c. End For.
3. End For.
4. End.
```

Procedure: Compute the resilience Level.

```
ComputeResilienceLevel (R, curs).

1. Set val <- 0.
2. Set count1 <- 0.
3. Set count2 <- 0.
4. Set max<- -∞.
5. Loop (R != NULL)
      a. val <-1.
      b. Count1 <-count1 + 1.
      c. If R.courseid = curs
           i.  count2 <-count2+1.
      d. End If.
6. End Loop.
7. Set count <- 0.
8. If val = 1
      a. If count2 = 1
           i.  Return 1.000
      b. End If.
      c. Else If count2= 0.
           i.  Return 0.000.
      d. End If.
      e. Else
           i.  Set i <-0
          ii.  Loop ( i <= count1)
                  1. Set j<- i+1.
                  2. If R[j].courseid = R[i].courseid AND
                     R[i].courseid = curs
                        a. If R[i].hour–R[j].hour<= 2
                             i.  Count <- Count +1.
                        b. End If.
                  3. Else.
                        a. If Max <= Count
                             i.  Max <- Count.
                            ii.  Max <- Max +1.
                           iii.  Count <- 0.
                        b. End If.
                  4. End If.
                  5. i <- i + 1.
         iii.  End Loop
          iv.  Return Max/count2.
      f. End If.
9. End.
```

Simulation of sample output: Resilience Level.

The following table simulates the output for a random day, D_i for a particular student S_j and it is also assumed that the student has been enrolled in three courses C_1, C_2 and C_3. It is also given

that he accessed the portal for 12 times on D_i. The Resilience Level is computed for all the three courses.

S. No	1	2	3	4	5	6	7	8	9	10	11	12
Course id	C_1	C_1	C_1	C_2	C_1	C_2	C_2	C_2	C_3	C_3	C_1	C_3
Time.	8.45	8.55	9.15	9.30	10.00	10.15	11.00	2.00	3.15	3.30	4.30	4.25
Running count For C_1.	1	2	3	0	1	0	0	0	0	0	1	0
Running count For C_2.	0	0	0	1	0	1	2	1*	0	0	0	0
Running count For C_3.	0	0	0	0	0	0	0	0	1	2	0	1

*Note it is not 3, because the difference in time stamps is greater than 2 though the activities belong to the same course.

Table 5.2. Simulation of Resilience Level

Computation of Resilience Level using the above formula:

1. For course $C_1 = \frac{3}{5} = 0.6$.
2. For course $C_2 = \frac{2}{4} = 0.5$.
3. For course $C_3 = \frac{2}{3} = 0.667$.

5.3.3. Effort Level.

This indicator is a very simple but could potentially be a define motivation from a different dimension. This indicator was defined and designed by me. This indicator also measures two aspects, one the motivation, second the cognitive index. The number of accesses and the number of submissions a student makes is counted for a time window. The Accesses refer to the number of clicks a student makes and the number of submissions a student makes refer to the attempts he makes.

Both the accesses and the submissions are considered because, consider the following cases. **Case 1.** Consider two students A and B. Let both the students make the same number of submissions but Student A makes 10 Accesses and let Student B make 6 Accesses. This shows that Student A is more motivated than Student B, because he seems to be more persistent. This

also comments on the cognitive capability of Student B, I.e. Student B requires a lot of trials before he submits.

Case 2. Consider two students A and B. the number of Attempts Student A makes is less than that of B's. Hence the effort level of student B is higher.

This indicator is also computed for each course and each student and every day. The formulae used for the computation of Effort and Cognitive indices is described below.

$$\text{EffortLevel} = (Attempts + 1) \times (Accesses + 1).$$

$$\text{CognitiveIndex} = \begin{cases} \frac{\textbf{No.of Attempts}}{\textbf{No.of Accesses}}, & if\ No.of\ Accesses \neq 0 \\ NA \quad , & if\ No.of\ Accesses = 0 \end{cases}$$

Note-3. The EffortLevel is not calculated as (Attempts) x (Accesses) because if number of attempts is zero and number of accesses is non-zero, then the product becomes zero which gives a wrong indication that the Effort put by a student is zero which is false. To avoid such misleading ales we add one to whatever the observed value is and make the product. This doesn't create any errors because this heuristic is followed for all data in the set. This is computed only for mandatory tasks because submissions are only for obligatory tasks.

This is computed for each course, each student and every day. The algorithm and a sample simulation is described in the following section. This is computed on a 15 day window for better results, i.e. the number of accesses and the number of submissions in a 15 day window is counted.

```
Algorithm 5: Effort Level.

EffortLevel ()

1. Begin
2. For Each Course Ci
     a. Curs <- Course_id (Ci).
     b. For Each Student Sj ∈ Ci
          i.  Set date_ini <- Max_date in the ForumAccess
              Table.
         ii.  If date_ini  NULL
                 1. Date_ini<- Beginning date of the term.
        iii.  Set date_final <- Current_date.
         iv.  Loop (date_ini <= date_final)
                 1. SQL <- Query_Accesses Join Modules ON
                    Moduleid (Curs, Sj, date_ini,
                    Mandatory=1)
                 2. Result <- Execute_Query (SQL).
                 3. Accesses <- count (Result).
                 4. SQL <- Query_Attempts (Curs, Sj,
                    date_ini, date_ini-15)
                 5. Result <- Execute_Query (SQL).
                 6. Attempts <- count (Result).
```

```
                    7. Effort_Level <-
                       (Attempts+1)x(Accesses+1).
                    8. If Accesses = 0
                         a. Cognitive_Index <- 'NA'
                    9. Else.
                         a. Cognitive_Index <-
                              Attempts / Accesses.
                    10.    End If.
                    11.   Update <- Insert (EffortLevel, Curs,
                       Sj, date_ini, Effort_Level,
                       cognitive_Index).
                    12.   Execute_query (Update).
                    13.   date_ini <- date_ini + 1.
              v.  End Loop.
        c. End For.
  3. End For.
  4. End.
```

Simulation for sample output: Effort Rate.

This simulation is a sample result of some random student S_j and for continuous 10 days.

Days.	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}
No.of Accesses*	0	1	4	14	21	23	23	23	24	27
No.of Attempts*	0	0	3	4	13	15	15	15	19	20
EffortLevel	1	2	20	75	308	384	384	384	500	588
CognitiveIndex	NA	0	0.75	0.29	0.62	0.54	0.65	0.65	0.80	0.74

*The accesses and submissions made between (D_i-15) and D_i for mandatory activities is counted.

Table 5.3. Simulation of Effort Level.

5.3.4. Enhancing the Agility Rate by filling the missing values.

Agility Rate is another very interesting indicator. The mathematical and statistical definitions of the Agility Rate is given the chapter 2. This indicator was not fully developed by me, it was developed by Pranathi, a team member in inLab. This indicator is used to measure how agile a student is, i.e. in simpler terms how long a student takes to do a specific task from the date of availability is computed by the indicator. The result is days in order to normalize this we compute the Normalization (Sigmoid(the difference between the date of availability and the date of submission in days)). A small assumption is made, that the date of availability is taken to be the date of first submission of the module. But on closer introspection of the equation we discover that whenever there is no submission the Agility Rate is NA. Now filling the missing values is a very important thing. For further analysis, there cannot be any missing values.

5.3.4.1. Ways of filing missing values.

The simplest way of missing the missed values ('NA') is to **simply replace them by 0s**. Though this is very efficient way, this is not very meaningful. According to the context, in this case, an agility rate with 0 means that the student responded to an activity after a very long time, more than a week which is not the case here. An NA here simple means that he has not accessed the portal. It is also seen that majority of the entries were NA. This could give a wrong idea that the student is less motivated. So this means that the student just didn't have any work to be submitted. So this method is discarded. Hence another method is followed.

Another sensible method would be to **replace the missing values with the average agility rate of that student**. Though this could be a very good method, this is ruled out because this could mean this does not smooth of agility rate. Also there are a lot of cells with NA. The other points that actually contain data is over ridden by the average rate, which is not expected. To overcome this, the following method is formulated.

The following method is adopted. The missing values are replaced with the nearest value that has value. For an instance consider a window d_i and d_j that contains values and the intermediate values are missed,

$$\text{FillMissingValues} = \begin{cases} di, between\ di+1\ and\ \frac{di+dj}{2} \\ dj, between\ \frac{di+dj}{2}+1\ and\ dj-1 \end{cases}$$

The algorithm for this explained in the next section.

```
Algorithm 6: Filling missing values.

Fill_missing_Values()

1. Begin
2. For Each Course Ci
     a. Curs <- Course_id (Ci).
     b. For Each Student Sj ∈ Ci
          i.  SQL <- Query_AgilityRate(Curs, Sj, AgilityRate
              <> NA)
         ii.  dates <- Execute_Query (SQL).
        iii.  If dates is NULL
                1. Set all the values in the AgilityRate to
                   0.
                2. Return.
         iv.  End If.
          v.  While (dates != NULL)
                1. Date1 <- dates.
                2. Date2 <- dates.next.
                3. Date_avg <- (Date1 + Date2)/2.
                4. Update <- Set Agility rate =
                   Date1.agilityRate in (Date1,Date_avg]
                   and Date2.aglityRate in
                   (Date_avg,Date2).
```

```
                    5. Execute_Query(update).
                    6. dates <- dates.next.
            vi.  End Loop.
        c. End For.
    3. End For.
    4. End.
```

Simulation of a sample output: Filling missing values

The output is simulated for a random student S_j belonging to a curs C_i. This demonstrates how the Algorithm works.

Days	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{10}	D_{10}	D_{10}	D_{10}	D_{10}	D_{10}
Agility Rate	NA	NA	NA	0.35	NA	NA	0.60	0	NA	NA	NA	1	NA	NA	NA	NA
Missed Values filled.	0.35	0.35	0.35	0.35	0.35	0.60	0.60	0	0	0	1	1	1	1	1	1

Table 5.4. Simulation of Filling Missing Values.

Note-4. In this report only simulations of the algorithms are shown because, all these computer algorithms were done on databases and the R scripts were attached to the ETL process. And just to make things clear simulations of the algorithms are described here.

In addition to these 4 indicators 4 more indicators were developed by the inLab Learning Analytics team. Ivan Vukic computed PCA on these indicators and performed some statistics and gave concluding concrete results about the motivation index.

In this chapter the various algorithms developed for the indicators and their efficiency is described and how the access logs plots and the summary plots were developed is discussed. This chapter also very clearly says the importance of indicators. In the next chapter the conclusions, the scope or future work and the discussions are discussed.

CHAPTER 6

Conclusion.

In this the final concluding results on the study carried out and the various scope for future result and its learning outcomes are discussed. This is being a research project, there is a scope for a lot of future work which might bring a whole new dimensions to the problem. There are also a few suggestions that are included that can make the project scale great heights.

6.1. Conclusions and Results.

In this section the conclusion and the various results are produced. A bit of context might help here. The TFG has been concentrated on Learning Analytics. The Moodle based Agora data is analyzed to identify the patterns and perform analytics on the data. A system of indicators are identified. The various indicators that were identified are designed and developed. In addition to this a visualization of access logs and a bouquet of summary plots are developed.

The various design issues and the implementation aspects was provided in the previous section. The significance of the indicators can be realized only when the entire study undertaken is thoroughly understood. The indicators developed are highly instrumental in the measurement of a student belonging to a particular student. The whole point about indicators is that the essential factors related to motivation are captured. A concrete result about the correctness of indicators cannot be produced because the results have yet to be experimentally be verified. The measurement of motivation is described in Ivan Vukic's report. The indicators were intuitive enough and they made perfect sense when described. These indicators were captured from theory and the most important aspect is realizing these indicators from the data that is available in hand. But from the results it was identified that the indicators performed well and gave satisfactory results.

Another major thing that needs to be understood well is that only a pilot version of the project is developed. Certain indicators perform well only in a competitive environment. For example the resilience level's actual behavior can be understood only when there are more than one course for a particular student.

The tables in the database grows at the rate of 500 rows every day. This is just to give a brief information on the size of the data that the project handles. When the size of the data keeps increasing the system naturally cannot perform really as expected. This degenerates into a big data project. In the next section the various potential pitfalls are described.

6.2. Potential Pitfalls and Suggestions.

1. Scalability. Scalability is one major potential pitfall. The system that is aimed at developing needs to scale to the size of the data. The current system performs reasonably well but the web pages and the illustrations produced may not be very responsive enough when the size of the data is high. Scalability is an important aspect that needs to be worked on. There is a definite increase in the amount of data.

2. Potential Data raise. The system has a keen increase in the amount of data. Only a pilot version of the project is developed. When a full version of the project is developed, the amount of data is just going to increase. The system must be elastic enough to handle this dense data. The programs developed must be capable enough to handle the data.

3. Big data opportunities. The project can be transformed into a big data system because for the same reasons as the previous. The system performs better when a big data style is undertaken. An online processing system needs to be used.

4. Algorithmic complexity. The algorithms developed as a part of the system is pretty simple and the complexity may be improved better. The indicators are computed for everyday for every student and every course. This was done this way because the statistical part of the project required this schema. A better computational methodology may be designed that captures the same essential data but at a much lower cost.

5. Choice of databases and data model. The current data model is the relational data model. A tree schema such as the couchDB or the MongoDB may be used that enhances as the performance further without compromising on the computational efficiency.

6. Proper indexing of database tables. The database needs to be properly indexed which enables faster retrieval of queries.

These are some design considerations that may be incorporated in the project that improves the performance of the system.

6.3. Scope for Future Work.

This being a research project, there is a very high possibility of future work. As a result of the study undertaken, a decent level of structuring the data has been achieved. In additional some interesting observation in the form of indicators has been captured. Motivation, a very abstract phenomenon has been measured as an index in this project. Though it cannot be confidently said that the project measured motivation, it can be said a systematic approach for the measurement of a factor that may exhibit motivation has been laid out.

1. The current project only performs exploratory data analytics. The data base and their schema has been well understood as a part of this project and a system of indicators has been developed. But the correctness of the indicators have not been validated. This requires experts from the corresponding field of motivation and psychology. The indicators need to be tested with a lot of different and diverse data.

2. From the computational aspects, Educational data mining an annex to Learning analytics may be performed. One example is design of a clustering algorithm that automatically groups the students such that each group has a distributed set of individuals that help one another and the overall state of art of the class improves.

3. This study undertaken doesn't account for performance. Performance of the students may be gathered and a correlation study of performance and motivation can be conducted and that may give raise to very interesting observations.

4. A system may be developed to alert student whenever their motivations drops or some of the indicator go low. This in a way ensures the class remains motivated.

5. Indicators for teachers may be developed and deployed and provide insights to them may be provided. For an instance if one student in a class is not motivated, this means the fault lies with the student. If the whole class fails to be motivated then it is a silent indicator to the teacher.

6. Translate it to a scalable, comparative and a robust tool with very few approximations and very few errors and a high performance.

6.4. Learning outcomes.

This section summarizes the various learning outcomes of the project. This is the most important section of my TFG because this shows the things that I have learnt by carrying out this study.

1. Design and construction of indicators for aspects for which no potential indicators exist.
2. Data handling. Extraction of required data from huge databases and proper organization of the same.
3. Using a web based platform to perform analytics and visualize the results in an easy and a comprehensive manner.
4. Project management that ensure smooth running of the project.
5. Making sense out of massive data that give raise to interesting conclusions and observations.

CHAPTER 7

Bibliography

1. Ali Bahrami, “Object oriented Systems Development”, Use case diagrams. 1999.

2. Berhenke Amanda, Miller Alison L., Brown Eleanor, Seifer Ronald, Dickstein Susan (2011) - Observed Emotional and Behavioral Indicators of Motivation Predict School Readiness in Head Start Graduates - University of Michigan, Ann Arbor, Combined Program in Education and Psychology. Early Childhood Research Quarterly (Impact Factor: 1.67). 10/2011; 26(4):430-441.
   DOI: 10.1016/j.ecresq.2011.04.001 Source: PubMed.

3. Bessant, K. (1997).” The development and validation of scores on the mathematics information processing scale (MIPS)”. Educational and Psychological Measurement, 57(5), 841–857.

4. M.A. Chatti et al., “A Reference Model for Learning Analytics”, IJTEL, 2012.

5. Chelladurai Packianathan (2006) - Human Resource Management in Sport and Recreation -2nd Edition, ISBN-13: 9780736055888.

6. Chen, Y. (2004). Research on inquiry learning assessment in distance education. Master Dissertation, Beijing Normal University, P.R. China.

7. Christian Glahn et al., “Smart Indicators on Learning Interactions”, Creating New Learning Experiences On a Global Scale, Second European Conference on Technology Enhanced Learning, 2007, pp 56.

8. Gage Nathan and Berliner C. David (1984), Educational Psychology, (Boston: Houghton-Mifflin).

9. Han and Kamber, “Data mining, concepts and techniques”, Chapter 1, 3rd Edition, 2006.

10. Ioannis Dimopoulos et al., “Using Learning Analytics in Moodle for assessing students’ performance”, 2013.

11. Maren Scheffel et al., “Quality Indicators for Learning Analytics”, Educational Technology & Society, 17 (4), pp 117–132.

12. Miriam Ramrez, “Learning Analytics, DMKM Report”. 2014.

13. Nevid, J. S. (2013). Psychology: Concepts and applications (4th ed.) Belmont, CA: Cengage Learning.

14. Ourania Petropoulou et al., “A new learning analytics tool in Moodle for assessing students’ Performance”, Bulletin of the IEEE Technical Committee on Learning Technology, Volume 16, Number 1, January 2014.

15. Romainville, M. (1994). Awareness of cognitive strategies: The relationship between university students’ metacognition and their performance. Studies in Higher Education, 19(3), 359–366.

16. Ryan S.J.D. Baker Et Al., “Educational Data Mining and Learning Analytics”, 2013.

17. Scott Murray, “Interactive Data Visualization”, O’reilly books, 2013.

18. Scholl W. Richard (2015), The University of Rhode Island, http://www.uri.edu/research/lrc/scholl/webnotes/Motivation.htm as of 20.02. 2015.

19. Tanya Elias, “Learning Analytics: Definitions, Processes and Potential”, January 2011.

20. Veronica Diaz, et al., “Learning Analytics A Report on the ELI Focus Session”, May 2012.