Data Visualisation System for E-learning: Overview of System Design

Božidar Kovačić, Vanja Slavuj
University of Rijeka
Faculty of Informatics and Digital Technologies
Radmile Matejčić 2, 51000 Rijeka, Croatia
{bkovacic, vslavuj}@uniri.hr

Miodrag Sretenović
MI&DA Ltd.,
Šulekova 29, 47000 Karlovac, Croatia
m.sretenovic@mida.hr

Abstract*. One of the ways in which distance teaching and learning practice and results may be improved is to collect relevant data from various student activities performed during a course and analyse them so as to be able to draw relevant conclusions. Such data usually includes, but is not limited to, the data obtained from evaluation activities aimed at evaluating the set learning outcomes. The collected data is subjected to the defined learning analytics procedures by both the experts (e.g., educational decision-makers) and non-experts (e.g., course instructors) in learning analytics, so a visual representation of data may help in increasing the transparency and comprehensibility of the results. Designated learning data visualisation systems play a vital role in that regard and may offer a number of customised graphical displays of data.*

The paper at hand argues in favour of learning data visualisation helping to design better and more successful distance learning setups. It aims to describe a newly developed system for learning data visualisation which uses the data collected from a learning management system and allows it to be automatically prepared, analysed, and visualised for educational decision-making. The emphasis in the paper is put on the design and implementation of the system (all of its components), its back-end elements, and database design.

Keywords. data visualisation, database design, distance learning, e-learning, educational decision-making

1 Introduction

Possibly the main motivation for developing systems that provide visualisations of learning data is to use the data to create a better and more in-depth understanding of the learning process (Sedrakyan et al., 2019). Thus, visualising data related to different types of tests and graded learning activities within an e-learning system offers an opportunity to more easily perceive and better comprehend large amounts of data (Lowe & Matthee, 2020), to quickly and efficiently interpret the data (Donohoe & Costello, 2020), to make informed decisions about the learning and teaching process (Mavrikis et al., 2019), and to identify and monitor learning trends revealed by the data. This, in turn, makes a solid basis for increasing the quality and success of knowledge acquisition (Lang et al., 2022).

As is the case with other tools for online or distance learning, learning management systems (LMSs) gather data regarding the results of graded learning activities, allowing export of data in a number of standardised data formats (Black et al., 2008). Using this type of data, teaching and learning experts are able to evaluate students' learning achievement in a more precise and informed way, as well as to improve their understanding of the educational setup by:

1. visualising data on graded activities, and
2. analysing the learning success of individual students, groups of students, or entire student populations for different graded activities.

The goal of this paper is to present the overall architecture of the system for learning data visualisation that we designed and developed to extend the possibilities of an LMS used at our university, and to report on the progress of its development. A previous paper (Kovačić, 2022) focused solely on the implementation of the data import module (including the underlying database design) which uses exported LMS files with data on graded learning activities. This paper extends the system description and presents how the system is designed in order to prepare customisable visualisations for different activity types, emphasising the implementation of other modules necessary for data visualisation.

This paper is structured as follows. Section 2 gives an overview of background work and describes the current state of development of the system for learning data visualisation. Section 3 elaborates on the design of the data visualisation system, specially emphasising the extension and implementation of its database to accommodate novel functionalities for data preparation, import, and analysis. Section 4 concludes the paper and charts future developmental paths for the system.

2 Background Work

In this section we provide an overview of related work on systems for data visualisation and describe the system for learning data visualisation developed at our institution as it currently stands.

2.1 Literature Overview

The review of relevant scientific sources reveals that learning data visualisation, in the context of distance or online education, is a thriving area of research. It attracts practitioners who are concerned with the implementation of data visualisation systems (e.g., learning dashboards) on the one side, and educators who employ them as part of their teaching setups on the other. This area or research often relies on the broader fields of learning analytics and educational data mining by using their techniques to prepare the data for visualisation itself (Romero & Ventura, 2019). Researchers have stressed the importance of data visualisation literacy in both personal and professional life as crucial for making sound decisions (Börner et al., 2019).

Relevant literature also reveals a number of examples of implemented learning data visualisation systems. For example, Santoso and colleagues (Santoso et al., 2018) describe the development of their comprehensive learning dashboard intended for lecturers, implemented as a plugin to the well-known Moodle LMS. Their work focused on developing a tool called SCELE that allowed course instructors to collect and visualise data so as to facilitate their observation and analysis of learning data. Data visualisation itself was carried out using various JS and CSS libraries.

Bodily and colleagues (Bodily at al., 2018) developed a dashboarding system that supports students during online learning. It consists of two separate dashboards for content and skill recommendation (recommendations made based on the gathered data about gaps in student knowledge and skills) that are changing in real-time.

Naranjo and colleagues (Naranjo et al., 2019) presented CloudTrail-Tracker, an educational dashboarding tool for course instructors that provides a visualisation of resource usage (for the selected time period) by students in a massive online open course, as well as specific activities carried out by a particular student. The system may also be used by the students (overview of learning progress) and system administrators (overview of resource consumption).

Han and colleagues (Han et al., 2020) developed a dashboard to support collaborative argumentation in face-to-face contexts. It consists of two dashboards: one intended for students and one for course instructors. The used collaboration system (Trello) recorded learning behaviours of the students (i.e., made activity logs) that were then transferred into a designated database using an API. Following data analysis, students and teachers received colour-coded feedback on the activities they were engaged in.

Gutiérrez and colleagues (Gutiérrez et al., 2020) report on LADA: a learning analytics dashboard intended for academic advisors in their decision-making, based on predictive analysis. The tool was implemented in the context of higher education to be used when needing to make difficult academic choices about students' studies.

2.2 Current State of the Developed System for Data Visualisation

The developed system for learning data visualisation stores data regarding course instructors, courses, students, and testing/evaluation activities. The number and type of evaluation activities may vary across courses, but this variety is addressed by the system. The type depends on the way the course is carried out (e.g., evaluation during the semester in combination with a final exam, evaluation during the semester without a final exam, workshop modality, etc.) as well as the evaluation strategy opted for by the course instructor.

If provided appropriate files containing the results of evaluation activities, the system offers users visualisations of such data (e.g., course instructors, the management, educational decision-makers, etc.) without the need for a system administrator to change the settings or make any further adjustments to the file and its contents. Such an automated data import functionality allows users to define their own structure of records in the table designated for importing data from the selected (exported) file. During this automated process, a set of metadata is generated on the imported results about evaluations so as the system is able to link the data on evaluation activities from the file with the tables for storing data on the evaluation results in the visualisation system. The automated import functionality is carried out using an original algorithm for automatic data import, which was made possible by extending the relational data model so as to accommodate new metadata (Kovačić, 2022). The structure of the system is given in Fig. 1.

The developed system for data visualisation enables the user to visualise the results of evaluation activities in a course that have been defined in advance (e.g., different types of activities, specific grading methods, or unique requirements for passing an activity) based on the imported data.

The data visualisation system consists of the following modules (Kovačić, 2022):

- Interface – an interactive communication channel between the system and the users.

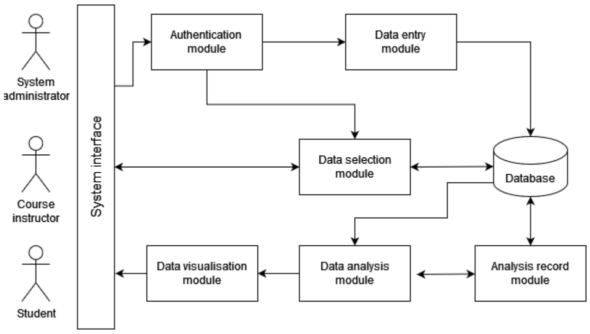

Figure 1. Structure of the system for data visualisation

- Authentication module – after entering a combination of username and password, the system authenticates the user, defines their role within the system (i.e., system administrator, course instructor or student), and authorizes the user to work with certain parts of the system.
- Data entry module – the system administrator enters data on courses, study groups, institutions, course instructors, and students[1] into the system, while the course instructor imports the data regarding the results of evaluation activities using an import file.
- Data selection module – course instructors and students can select a particular course and specific data regarding evaluation activities in order to create a customised visualisation (applying data-selection criteria).
- Data analysis module – provides a statistical analysis of the selected data regarding evaluation activities and prepares them for visualisation.
- Analysis record module – stores data on all data visualisations performed by the system in order to allow for a comparison between the current visualisation and previously generated ones.
- Data visualisation module – generates various types of diagrams to be presented as part of data visualisation.

After a successful login using their credentials, the user is authenticated and authorised to work with specific functionalities of the system. In the preparation step preceding user's interaction with the system, the administrator is responsible for entering or updating data on system users and courses so as to allow the users to work with only those courses assigned to them. Once data import regarding evaluation activities for a particular course has been completed, the course instructor selects specific data points that need to be included in the visualisation of data and analyses the results presented by the system. Records kept by the system allow the course instructor to compare and monitor students' learning progress and make appropriate teaching and learning interventions in the course. On the other hand, students are shown the results of their own learning endeavours, which may be compared to the (Kovačić, 2022) results of their fellow students, thus instigating comparison, encouraging competitiveness, and increasing motivation to perform better.

[1] Entering data on students is optional as the system allows for automatic data entry using the import file (at a later stage).

3 System Design and Implementation

In this section we provide an in-depth description of how the system for data visualisation is designed and implemented based on the previously provided description of its functionalities, with a special emphasis on the underlying database.

3.1 Technological Basis of the System for Data Visualisation

The technology stack used to implement the system for learning data visualisation includes the following elements: (1) MySQL relational database management system (RDMS), (2) Apache HTTP server, (3) Phalcon framework, and (4) Chart.js system for data visualisation.

The MySQL RDMS, Oracle's free and open-source software, is used to implement data storage for the data visualisation system. It is characterised by high reliability and ease of use, providing an environment for integrated development, design and administration of databases.

The Apache HTTP server was used to develop and manage an open-source HTTP server that hosts the application. The associated Apache project aims at providing a secure and extensible HTTP server in compliance with the modern HTTP standards (Apache HTTP Server Project, 2023).

The Phalcon framework was used in order to enter the data into the system, store it, select and prepare it for visualisation, and update the archive of performed analyses. Phalcon is an open-source full-stack PHP framework optimised so as to provide the lowest overhead in applications that use the MVC (model, view, controller) architectural pattern (Phalcon framework documentation, 2023). This particular architectural pattern separates the application into three main logical components in which a model represents data the system is working with as well as the rules governing its use, a view represents the user interface of the application, and a controller enables the flow of data between the models and associated views (The MVC architecture, 2023).

In the process of learning data visualisation, graphical representations of data are implemented using the Chart.js JavaScript library. This is a simple and flexible library that provides numerous chart types, plugins and customisation options, many of them available from and maintained by the community (Chart.js documentation, 2023).

3.2 System Implementation

3.2.1 Authentication Module

The main task of the authentication module is to store data regarding the users of the system, and to link user data to data about the courses (and other forms of organised educational units for which evaluation activities are carried out). The module enables the following activities:

- verifying user's identity by validating the username and password provided by the user,
- linking users to courses for which data visualisation is performed,
- linking courses with study groups and institutions.

Relational schema implementing the authentication module of the learning data visualisation system is given below, and the accompanying entity relationship diagram (ERD) is provided in Fig. 2.

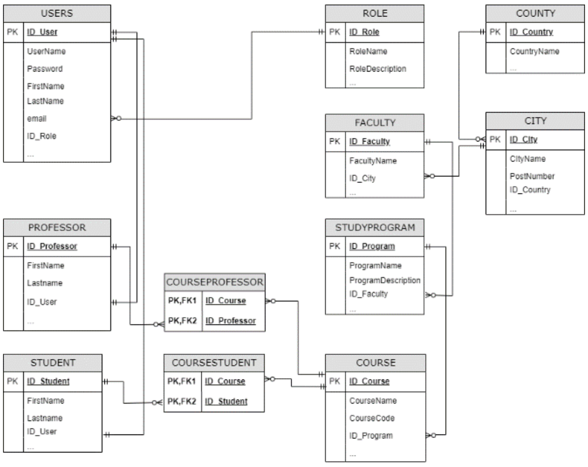

Figure 2. ERD of the system's authentication module

USERS(ID_User, UserName, Password, FirstName, LastName, email, ID_Role)
ROLE(ID_Role, RoleName, RoleDescription)
COURSE(ID_Course, CourseName, CourseCode, ID_Program)
STUDYPROGRAM(ID_Program, ProgramName, ProgramDescription, ID_Faculty)
FACULTY(ID_Faculty, FacultyName, ID_City)
CITY(ID_City, CityName, PostNumber, ID_Country)
COUNTRY(ID_Country, CountryName)
PROFESOR(ID_Professor, FirstName, LastName, ID_User)
STUDENT(ID_Student, FirstName, LastName, ID_User)
COURSEPROFESSOR(ID_Course, ID_Professor)
COURSESTUDENT(ID_Course, ID_Student)

Verification of user's identity is achieved using the USERS relation which contains the username, password, first and last names of the user, and the assigned user role in the system. System administrator enters data regarding course instructors into the USERS relation, which are then transferred into the PROFESSOR relation (course instructors are not allowed to change the data on their own). Finally, the administrator enters the data regarding the course assigned to a course instructor (relation COURSEPROFESSOR). There are two ways of entering data regarding the students:

1. the system administrator enters the data directly into the USERS relation, which are then transferred into the STUDENT relation and linked with the courses in the COURSESTUDENT relation.
2. using the automated process of importing data from the import file, during which data is entered automatically into the STUDENT and COURSESTUDENT relations. If the students are allowed access to the system for learning data visualization, their data is added to the USERS relation and the system generates and send them a password (which is to be changed upon their first access to the system).

The ROLE relation contains data identifying the roles system users may be assigned and the accompanying description of each role. The COURSE relation stores data such as course names, course identifiers, as well as relationships with the study programme (relations STUDYPROGRAM, FACULTY, CITY, COUNTRY).

3.2.2 Data Entry Module

The data entry module is responsible for storing the data regarding the results of evaluation activities within a course and linking the imported data with data on courses, course instructors, and students. The module enables the following activities:

- selecting courses and files for data import,
- linking data from the import file to the relation for storing data regarding the results of evaluation in the system for data visualisation,
- reviewing imported data.

Relational schema implementing the data entry module of the learning data visualisation system is given below.

COURSE(ID_Course, CourseName, CourseDescription)
PROFESSOR(ID_Professor, FirstName, LastName, …)

COURSEPROFESSOR(ID_Course, ID_Professor)
STUDENT(ID_Student, FirstName, LastName, …)
COURSESTUDENT(ID_Course, ID_Student)
GROUP(ID_Group, GroupName, GroupDescription, …)
STUDENTGROUP (ID_Course, ID_Student, ID_Group)
STRUCTURE (ID_Structure, StructureName, StructureDescription)
CATEGORY (ID_Category, CategoryName, CategoryDescription, ID_Structure)
COURSEDATA(ID_Results, ResultName, ResultDescription, ID_Course)
COURSESTRUCTURE(ID_Course, ID_Structure)
COURSEDATASTRUCTURE(ID_Course, ID_Structure)
COURSERESULTDATA (ID_CRD, ID_Course, ID_Profesor, ID_Result, ID_Structure, ID_Category, ID_Group, ColumnNumber, ColumnName, FileColumnName)
RESULTS (ID_Result, ID_Student, Column1, Column2, …., ColumnN)

Course instructors select one of the courses assigned to them and add new tuples into the relation for storing imported data (relation COURSEDATA). Only then are they able to specify the import file (its location on the user's computer). Data entry is facilitated by the automatic process which consists of two steps:

1. The course instructor selects the way the course is carried out (i.e., modality) and an arbitrary number of categories (activity which includes multiple data points from the import file, e.g., evaluation activity which is carried out in several groups of students). This is implemented by relations STRUCTURE, COURSEDATASTRUCTURE, and CATEGORY. Then the data header from the import file is read (table header regarding the results of evaluation activities). The course instructor then selects the data to be imported (table columns from the import file), decides on the modality in which the course is carried out, and an arbitrary category for the selected data point. After each selection is made, data is stored in the COURSERESULTDATA relation and a column from the RESULTS relation is chosen to store the selected data point from the import file.
2. Data is imported using the automated process, i.e. by reading the data on each student (tuple from the table in the import file) and transferring it to the RESULTS table (data point regarding the column of the RESULTS relation used to store the data from the file is defined by the COURSERESULTDATA relation).

After all the data have been entered into the system, selecting a group of students for which the analysis will be performed may be carried out in one of two ways:

1. The course instructor defines groups of students and links them with to data from the import file (during the first step of data import).
2. Using the automated process which takes each data point that has been previously included in the category and records only those students who have the same value as the selected data point. The student group is created subsequently (relations GROUP and STUDENTGROUP).

After importing the data, the course instructor may opt to review the modality in which the course is carried out, along with all the evaluation activities in the course that have been entered into the system using the import file and the overview of all the entered data (relations COURSERESULTDATA and STUDENTGROUP).

The ERD of the data entry and data analysis modules is provided in Fig. 3.

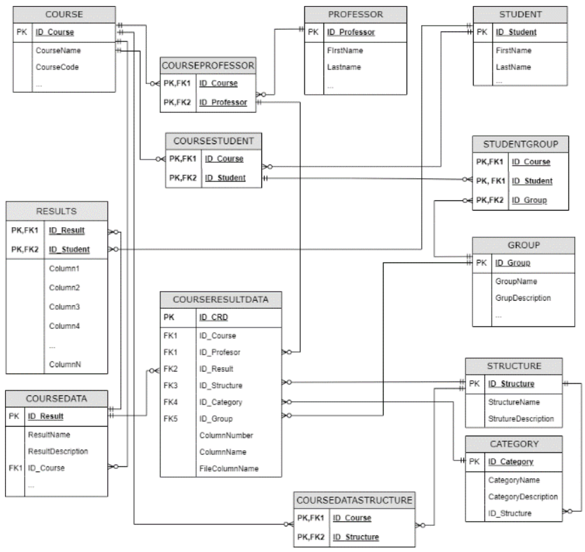

Figure 3. ERD of the system's data entry and data analysis modules

3.2.3 Data Analysis Module

The data analysis module is in charge of receiving the dataset that needs to be visualised by the system and performing statistical analyses using the data (e.g., maximum values, minimum values, averages, standard deviations, etc.). The analysed data is then rendered as a diagram of a particular type, depending on the characteristics of the data or the setting associated with a diagram type. The analysis is performed for each selected dataset that needs to be graphically displayed, and is entered into the ANALYSE relation.

System's data analysis module is based on the same relational data model as the data selection module (see Fig. 3). The main difference between the two is that the data selection module does not deal with the data from the ANALYSE relation, while the data analysis module performs statistical operations for the selected dataset and calculates statistical values in the ANALYSE relation.

3.2.4 Data Selection Module

The purpose of the data selection module is to display the data imported into the data visualisation system regarding evaluation activities, categories, and modalities in which the course is carried out.

It allows the course instructor or the student to select a (sub)set of data that needs to be visualised by the system, as well as the criteria for sorting and/or filtering this data. Relations ANALYSE, ANALYSESTRUCTURE, and ANALYSEGROUP are used to store the relevant data. In the last step, the system is able to select all the data about students from the RESULTS relation and prepare it in the format required by the diagram-generating subsystem (i.e., for data visualisation).

The ERD of the data selection module is provided in Fig. 4.

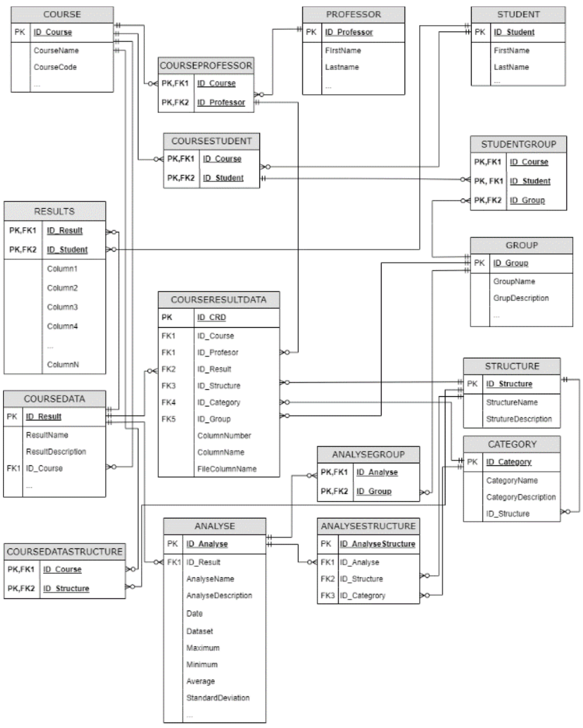

Figure 4. ERD of the system's data selection module

3.2.5 Data Visualisation Module

The data visualisation module displays diagrams in the process of data visualisation by using the data in the ANALYSEGRAPH, ANALYSEGRAPHELEMENT, GRAPHTYPE, and GRAPHELEMENT relations.

The GRAPHTYPE relation keeps records regarding the data about diagrams used for data visualisation. Records related to the selected diagram type are stored in the ANALYSEGRAPH relation. Values necessary for the diagram to be rendered on the screen (descriptive data such as diagram title, diagram description, axis names, etc.) are kept in the GRAPHELEMENT relation. Finally, additional user settings for the selected diagram type are stored in the ANALYSEGRAPHELEMENT relation.

All the data contained in the above described relations are entered into the relations only if the user wants to use them for comparison with any subsequent data visualisations. This allows the system to store both the dataset for data visualisation and configuration settings for diagram display.

3.2.6 Analysis Record Module

The purpose of the analysis record module is to store data on all previous data visualisations in order for the user to be able to compare them to the newly performed ones. This allows the users to detect changes, keep track of trends, and identify problems or difficulties in the learning and teaching setup. Furthermore, it allows the users of the system to make informed decisions concerning the observed issues and to improve the educational process as a whole.

The relational schema of the data analysis, data visualisation, and analysis record modules is given below.

ANALYSE(ID_Analyse, ID_Result, AnalyseName, AnalyseDescription, Date, Dataset, Maximum, Minimum, Average, StandardDeviation)
ANALYSESTRUCTURE(ID_AnalyseStructure, ID_Analyse, ID_Structure, ID_Category)
ANALYSEGROUP(ID_Analyse, ID_Group)
GRAPHTYPE(ID_Graph, GraphTypeName, GraphTypeDescription)
ANALYSEGRAPH(ID_Analyse, ID_Graph, JSON, Decription, SelectionCriteria)
GRAPHTYPELEMENT (ID_GraphType, ID_Element, ElementName, ElementDesc, DefaultValue)
ANALYSEGRAPHELEMENT(ID_AnalyseGraphElement, ID_Analyse, ID_GraphType, ID_Element, Value)

Both data visualisation module and analysis record module use the same relational data model. The main difference, however, is that the Data visualisation module writes data needed for current graph visualisation into the ANALYSEGRAPH relation, while the analysis record module uses previously generated data from the ANALYSEGRAPH relation to visualise data in previous data analyses runs.

The ERD of the data visualisation module is provided in Fig. 5.

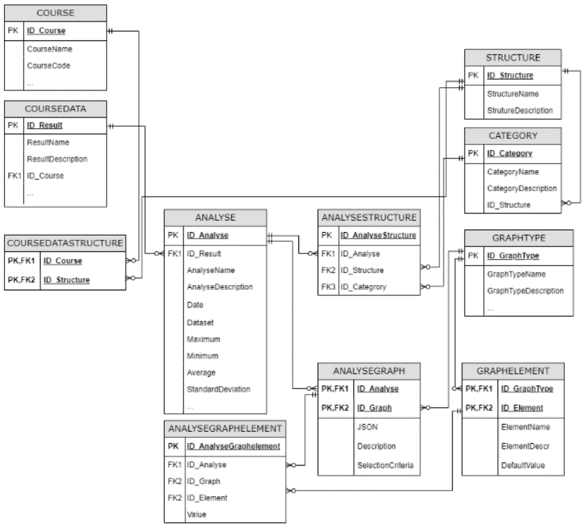

Figure 5. ERD of the system's data visualisation module

Fig. 6 offers an example of overview of student learning data (after import), data selection, and data visualisation as a series of screenshots taken directly from the described data visualisation system.

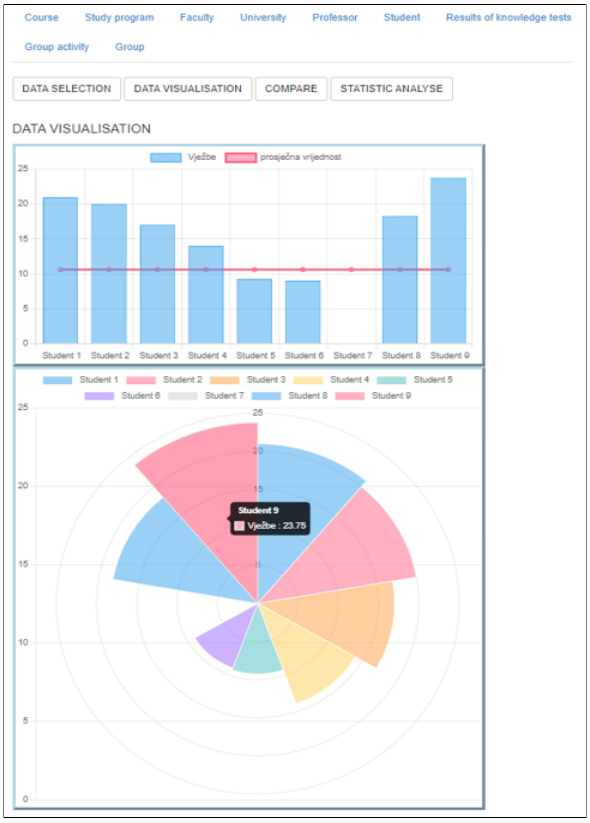

Figure 6. Screenshots from the system

4 Conclusion

This paper focused on the description of the system for learning data visualisation developed at our institution. A system that enables data analysis and its subsequent graphical representation, may aid all the stakeholders (i.e., course instructors, students, decision-makers, etc.) in making well-informed and effective decisions that improve both teaching and learning practices. The developed system allows learning analytics experts and non-experts to perform analyses on student generated data (evaluation activities) by importing the data and automatically preparing it (metadata creation) for further analysis.

Special emphasis was given to the implementation of the system's back-end and the design of its database, which enables the various functionalities of the system and its six interconnected modules.

Future work on the system will be aimed towards improving the results of analysis of data selected for visualisation, so as to enable the discovery of new trends in knowledge acquisition and improve decision-making during the teaching and learning process.

Acknowledgments

This work was supported by the University of Rijeka projects uniri-drustv-18-73 and uniri-drustv-18-224.

References

Apache HTTP Server Project. (2023). Retrieved from https://httpd.apache.org/

Black, E. W., Dawson, K., & Priem, J. (2008). Data for free: Using LMS activity logs to measure community in online courses. *The Internet and Higher Education*, 11 (2), 65–70. doi: https://doi.org/10.1016/j.iheduc.2008.03.002

Bodily, R., Ikahihifo, T. K., Mackley, B., & Graham, C. R. (2018). The design, development, and implementation of student-facing learning analytics dashboards. *Journal of Computing in Higher Education*, 30, 572–598. doi: https://doi.org/10.1007/s12528-018-9186-0

Börner, K., Bueckle, A., & Ginda, M. (2019). Data visualization literacy: Definitions, conceptual frameworks, exercises, and assessments. *PNAS*, 116(6), 1857–1864. doi: https://doi.org/10.1073/pnas.180718011

Chart.js documentation. (2023). Retrieved from https://www.chartjs.org/docs/latest/

Donohoe, D., & Costello, E. (2020). Data Visualisation Literacy in Higher Education: An Exploratory Study of Understanding of a Learning Dashboard Tool. *International Journal of Emerging Technologies in Learning (iJET)*, 15(17), 115–126. doi: https://doi.org/10.3991/ijet.v15i17.15041

Gutiérrez, F. et al. (2020). LADA: A learning analytics dashboard for academic advising. *Computers in Human Behavior*, 107, article 105826. doi: https://doi.org/10.1016/j.chb.2018.12.004

Han, J., Kim, K. H., Rhee, W., & Cho, Y. H. (2020). Learning analytics dashboards for adaptive support in face-to-face collaborative argumentation. *Computers & Education*, 163, article 104041. doi: https://doi.org/10.1016/j.compedu.2020.104041

Kovačić, B. (2022). Extension of e-learning data visualization system for different knowledge reviews datasets. In L. Gómez Chova, A. López Martínez, & J. Lees (Eds.), ICERI2022 Proceedings (pp. 7396–7403). IATED Academy, Valencia. doi: 10.21125/iceri.2022.1882

Lang, C. et al. (2022). What is learning analytics? In C. Lang, G. Siemens, A. F. Wise, D. Gašević, & A. Merceron (Eds.), *Handbook of Learning Analytics – Second edition* (pp. 8-18). SoLAR, Vancouver. doi: 10.18608/hla22.001

Lowe, J., & Matthee, M. (2020). Requirements of Data Visualisation Tools to Analyse Big Data: A Structured Literature Review. *Responsible Design, Implementation and Use of Information and Communication Technology*, 2020 Mar 6; 12066, 469–480. doi: 10.1007/978-3-030-44999-5_39

Mavrikis, M., Geraniou, E., Gutierrez-Santos, S., & Poulovassilis, A. (2019). Intelligent analysis and data visualisation for teacher assistance tools: the case of exploratory learning. *British Journal of Educational Technology*, 50 (6), 2920–2942. doi: https://doi.org/10.1111/bjet.12876

Naranjo, D. M., Prieto, J. R., Moltó, G., & Calatrava, A. (2019). A Visual Dashboard to Track Learning Analytics for Educational Cloud Computing. *Sensors*, 19(13), article 2952.https://doi.org/10.3390/s19132952

Phalcon framework documentation. (2023). Retrieved from https://docs.phalcon.io/5.0/en/introduction

Romero, C., & Ventura, C. (2019). Educational data mining and learning analytics: An updated survey. *WIREs Data Mining and Knowledge Discovery*, 10(3), e1355. doi: 10.1002/widm.1355

Santoso, H. B., Batuparan, A. K., Isal, R. Y. K., & Goodridge, W. H. (2018). The development of a learning dashboard for lecturers: A case study on a student-centered e-learning environment. *Engineering Education Faculty Publications*, Paper 236.

Sedrakyan, G., Mannens, E., & Verbert, K. (2019). Guiding the choice of learning dashboard visualizations: Linking dashboard design and data visualization concepts. *Journal of Computer Languages*, 50, 19–38. doi: https://doi.org/10.1016/j.jvlc.2018.11.002

The MVC architecture. (2023). Retrieved from https://docs.phalcon.io/3.4/en/mvc