

Analytical Tool for the University Data Management

Ivan Pastierik, Michal Kvet
University of Žilina
Žilina, Slovakia
pastierik2@stud.uniza.sk, michal.kvet@uniza.sk

Abstract—Every university needs some kind of information system that will allow the university to record students, teachers, student results and other elements related to the student's studies. Such software is called an academic information system. This paper describes the analysis, design, implementation and testing of similar system. However, unlike a typical academic information system, this system, which was created as part of a bachelor's thesis on the topic Analytical Tool for the University Data Management, also contains advanced analytical tools that will allow both students and teachers to analyse the course of study of individual students and find the weak points of teaching and educational process. In order to create the analytical tool, it was first necessary to analyse the transition of students through their studies. The Faculty of Management and Informatics of the University of Žilina was used as a source of this analysis. Based on this analysis, we designed a database that uses Oracle SQL language running on the Oracle Cloud technology, and then created a tool in the Oracle Application Express environment. The tool allows registering of students, their exams, teachers, and subjects. At the same time, it also contains functions for searching, sorting, and filtering according to various criteria. Another functionality is the ability of user to display summary and individual statistics for teachers, students, and study groups. Each student and teacher have their own account to access global statistics. The tool offers possibilities for future expansion, such as comparing students and advanced analysis using the functionalities of the Oracle Machine Learning service.

I. INTRODUCTION

Currently, there is a lot of software for registering and managing students and teachers, but few of them also offer any effective analytical tools that would enable collective or individual analysis of processes at the university.

An example of a tool for registering and managing students and their subjects is also the university portal of the University of Žilina [1]. This academic information system was analysed as a part of bachelor's thesis on topic Analytical Tool for the University Data Management [2]. The portal is designed to offer basic information for the student. The student has the opportunity to view a study schedule, information about accommodation, but the most important thing is that he can access his subjects and his assessment through this portal. This portal, however, provides only basic analytical tools for students like calculating mean of subject results display statistics for different subjects, for example how many students studied a given subject in the previous year while also showing the proportion of grades these students received.

The subjects are divided by semester, and for each subject the subject code and subject name are displayed in the

"Subject" column. In the "Teaching" column, you can open the subject's information sheet and access the subject's MOODLE page. The MOODLE system is used for the publication of course materials and student evaluation. The "Evaluation" column contains two items - exam dates and evaluation. The evaluation serves to provide student feedback on the subject.

The system of the University of Žilina is clear and intuitive to use and contains basic study analysis options. However, this system lacks more advanced analytical tools for the student, thanks to which he could easily compare himself with other students and find out his strengths and weaknesses.

As part of the bachelor's thesis, from which this paper is based, we also analysed the systems of various universities, other than the University of Žilina. We analysed academic information system of Unicorn Collage in Prague [3], academic information system of Comenius University [4] and academic information system of Technical University in Munich [5].

Based on this analysis, we have concluded that these systems have sophisticated ways to record students, their subjects and assessments, but these systems do not offer very detailed tools for analysing the student's studies. Based on this fact, we have decided to create a tool that would offer the possibility to register students, their subjects and evaluation, and the system would also offer both students and teachers the possibility to analyse the results and progress of the students' studies.

II. ANALYSIS

This chapter deals with a basic description of the processes of the University of Žilina related to the student. This analysis focuses on the processes of student registration for studies, assigning students to study groups, enrolling students in subjects, closing the grade and the overall course of study from the student's point of view [2].

A. Student

Person studying on university is called student. The student chooses the level of study at which he wants to study. There are three level of study on the University of Žilina – bachelor's studies, engineering studies and doctoral studies. If he meets the requirements for study at the given level of study and for given study program, he can submit an application for study. If the student meets the requirements for the study and has successfully completed the admission procedure, he is enrolled for the study [2].

The study program represents the focus and field of study. Each study program has a prescribed maximum number of applicants for the given study program. The study program can support full-time or part-time study. Each study program has assigned a couple of study groups based on number of students that study that program [2].

A study group is a group of students who study together and have a common schedule. If the student is participating in doctoral studies, the student is not assigned to a study group. Likewise, the student is not registered in the study group as long as he is participating in studies abroad through the Erasmus+ program [2].

B. Subject

The main elements of study are subjects. Each subject has its own guarantor, who is responsible for the curriculum and method of teaching the subject. Subjects usually consist of exercises and lectures. Each student is enrolled in subjects based on the study program and field he is studying [2].

We know three types of subjects – compulsory subjects, compulsory-elective subjects and optional subjects. A compulsory subject is a subject that a student must complete in order to participate in the state exam. Compulsory-elective subjects are subjects from which the student must have a prescribed number of credits. Optional subjects do not have a prescribed number of credits that the student must obtain from them, but it is convenient for the student to write them down if they are interesting subjects that will advance him further, or if the student needs an additional number of credits from the study in order to be able to successfully complete your studies. In order for student to pass subject, student needs to successfully pass final exam of that subject [2].

The final exam of a subject is the final verification of the student's knowledge of the subject matter acquired during the semester. The student has three attempts at the exam. Only the last attempt is taken into the final evaluation [2].

The result of the subject is understood as the evaluation of the subject during the semester and also after the end of the semester. The final grade of the course is represented by a grade from A to Fx , where A is the best grade and Fx means that the student failed the course. The final grade is awarded based on the number of points obtained from the semester and the number of points obtained from the final exam [2].

III. APPLICATION REQUIREMENTS

Before the actual design of the application, we collected the requirements for the application. These application requirements consist of requirements gathered by controlled interview from teachers and students, and requirements that result from the nature of the problem being solved.

These requirements can be divided into four groups [2]:

- Registration and management of teachers.
- Analysis.
- Registration and management of students.
- Non-functional requirements.

A. Registration and management of teachers

This group of requirements contains requirements that apply to the management of teachers in the application. This is mainly about their registration and their assignment to the student's study processes.

1) *Record of teachers, their positions and departments:* The user must be able to add, edit, delete and view teachers. When registering teachers, in addition to their name, surname, title, date of employment and date of termination of employment, it is also necessary to register their department and their position in the department [2].

2) *Record of taught subjects:* For each teacher, it needs to be possible to find out which exercises and lectures the is the given teacher teaching [2].

3) *Record of taught students:* For each teacher, it must be possible to find out which specific students is the given teacher teaching [2].

4) *Account creation for teachers:* When adding a teacher to the application, an account must also be created for the given teacher, with which he will be able to access the application [2].

B. Analysis

This group of requirements refers to requirements for analysis and manipulation of registered elements such as students, teachers or study groups.

1) *Filtering:* Filtering should be an important part of the application and it is necessary that individual elements can be filtered based on various criteria related to these elements [2].

2) *Sorting:* Sorting is also quite useful tool, thanks to which lists and tables of elements become clearer and, above all, it is easier and faster to find the specific elements [2].

3) *Displaying summary statistics:* In the application, it must be possible to find out statistics about groups and subgroups of elements [2].

4) *Displaying individual statistics:* For individual elements, it must be possible to display their statistics [2].

5) *Graphic representation of statistics in the form of a graph:* For given groups of statistics, either aggregated or individual statistics, it must be possible to display them graphically in the form of a graph [2].

C. Registration and management of students

This group of requirements discusses the methods of registering students, their studies, enrolled subjects and evaluation of students.

1) *Record of students*: User needs to have the ability to insert, edit, delete and display students and their personal data and personal number [2].

2) *Record of enrolled subjects of student and student results*: User should be able to insert, edit, delete and display student's subjects, number of points from semester and final exams of these subjects and their final grade [2].

3) *Record of study groups*: There also needs to be possibility to register study groups and assign students to these study groups [2].

4) *Records of students' studies, including participation in Erasmus+*: Each student has automatic registration of their studies, and the status of those studies is updated upon the completion of the academic year. Additionally, it is also necessary to record for each semester whether the student participates in the Erasmus+ program during that semester [2].

5) *Account creation for students*: When inserting a student into the database, it is necessary to create an account for this student, with which he can log in to the application [2].

6) *Bulk import of students and their enrolled subjects*: It must also be possible to insert students in bulk by importing a file that adheres to a predetermined data format [2].

D. Non-functional requirements

These requirements are important when choosing technologies to use in development of the application and determining the overall appearance and structure of the application.

1) *Responsive design*: Each page must be viewable and readable on different types of display devices such as mobile phones, tablets or computer screens. They must also support different aspect ratios and screen resolutions [2].

2) *System security*: The system must be regularly updated with patches for security holes and risks. Those updates must happen automatically. There must be no data leakage from the application. The data must also be encrypted and the application must be safe from the point of view of SQL injection. The system must be resistant to system outages. This means that the system must not run on a single server, but must be available via "Availability domains". Availability through "Availability domains" means that the application is dispersed among several data centres that are connected to each other in such a way that it can practically be said that they are in a local network and each of these data centres contains duplicate data of the application. That is, if one data centre fails, any of the remaining data centres will be used, which means that failure of the entire application becomes very unlikely [2].

3) *Creation of regular backups*: The system must be regularly backed up in case of file corruption or loss.

4) *Extensibility*: The application must be easily expandable with new functionalities in the future [2].

5) *Transparency of pages*: The pages must be clear and readable, there must not be a lot of information in one place. This clarity also applies to the analytical tools and graphs provided by this application [2].

IV. APPLICATION DESIGN

This chapter deals with the description of the technologies used in the creation of the application.

A. Oracle Cloud

Based on the application requirements and analysis, we have decided to implement the application using Oracle Cloud Infrastructure (OCI) technology. OCI technology is a comprehensive cloud service provided by Oracle, which is based on the principles of Infrastructure as a Service (IaaS) and includes various types of services, such as database services, services for data management, their integration and analysis, artificial intelligence, application development and others. OCI provides customers with a high degree of flexibility and scalability to build and manage applications in a modern cloud environment with high computing capacity. With its advanced tools and technologies, OCI enables the rapid development and deployment of applications and with that increasing the efficiency and competitiveness of businesses [2], [6].

Students' and teachers' access to this technology is ensured by the Oracle Academy Cloud Program. The Oracle Academy Cloud Program offers Oracle Academy members exclusive access to OCI at the Oracle Cloud Free Tier level, a set of services that teachers and students can continue to use indefinitely even after graduation. Thanks to this service, teachers and students can explore the full capabilities of the Oracle autonomous database for an unlimited time. Another advantage of this program is the courses and tutorials, thanks to which students and teachers learn new technologies in a short time [2], [6].

Part of this program is always free access to various services such as Oracle Application Express (APEX), which is a service for creating websites with the use of a small amount of code, SQL Developer Web, with which it is possible to access the Oracle autonomous database and also offers the possibility of managing and working with this database. Other services provided are SQL Notebooks for machine learning, Oracle REST Data Services for web interfaces and Oracle Instant Client for the most used programming languages. Students can develop in SQL, NoSQL, APEX, Java, Node.js, Python, PHP and Ruby [2], [6], [7], [8].

Oracle SQL (Structured Query Language) is a declarative programming language used to manipulate data in relational database systems. Oracle SQL allows users to define and modify data in database tables, execute queries and retrieve outputs [2], [9], [10].

Thanks to Oracle SQL users can create tables, indexes, views and other objects in the database. Thanks to integrity constraints, data in the database will always be valid and consistent. Transactions that ensure atomicity, consistency,

isolation and durability of data in the database can also be defined [2], [6], [9], [10] [11].

Another great advantage of Oracle SQL is, that it provides an option for data backup and recovery and protection against loss of data. It also enables users to troubleshoot performance issues and optimize queries using indexes and material views [2], [6], [9], [10] [11].

Currently, Oracle SQL is one of the most widely used languages for working with databases and its use has spread to various fields such as banking, financial services, telecommunications, healthcare and many others [2], [6], [9], [10] [11].

This language is fully supported and integrated into OCI and is used by the Oracle APEX [2], [11].

B. Oracle APEX

Oracle Application Express (APEX) is a fast and simple tool for creating web applications for Oracle databases. Oracle APEX makes it possible to create applications without the need for programming and using a web interface that is easily accessible through a browser [1], [2], [11], [13], [14], [15], [16].

The Oracle APEX is based on the principles of database programming, which means that all data and application code are stored directly in the Oracle database. This ensures data security and integrity, fast processing and simple data access control. Oracle APEX is fully integrated with the Oracle autonomous database, which ensures fast development, easy maintenance and high application performance [1], [2], [11], [13], [14], [15], [16].

An example of the use of the Oracle APEX can be the creation of various types of applications, such as internal web applications, content management systems, e-commerce applications and many others. Due to its ease of use, speed of development and full integration with Oracle Database, Oracle APEX is becoming a popular tool among application developers and companies looking for fast and efficient solutions for their business needs [1], [2], [11], [13], [14], [15], [16].

Oracle APEX consists of several main elements such as sites, regions, applications and processes. Pages are the basic unit of a user interface and usually contain one or more regions for displaying and interacting with data. Regions are the basic elements of a page layout and include tables, charts, forms and more. Applications are collections of pages and processes that together make up an entire application. Processes are parts of the application that provide calculations, data validation and other functions on the server [1], [2], [11], [13], [14], [15], [16].

The appearance of the application is ensured by the Universal Theme (UT) integrated directly in the Oracle APEX. UT is a predefined design pattern that enables fast and easy creation of web applications with high quality user interface. UT includes multiple elements such as header, footer, side menu, and content areas that can be customized to meet the

needs of a particular application [2], [17].

One of the main advantages of using UT is the speed and ease of creating web applications. UT includes ready-made graphic elements and styles that can be easily adapted to the needs of the application. It also allows developer to quickly change colours, typographic elements and many other settings, which simplifies the creation of a user-friendly application [2], [17].

Another advantage of using UT is its full responsiveness, which means that applications are displayed correctly on different devices such as computers, tablets and mobile phones. UT automatically adjusts to the screen size, ensuring that apps are always displayed correctly and in a user-friendly way [2], [17].

C. Data generator for sample data in database

Due to the fact that data model of the application is quite extensive, we decided to create a sample data generator for development and testing purposes, which will allow us to easily regenerate the data if there is any error in that data. At the same time, this generator will allow us to create more data samples in a relatively short time [2].

Data generator consists of two main parts – data generator and address generator [2].

Data generator part is used for generating data into the database. The generation of the database can be influenced by specifying different parameter values. Those parameters include range of data generation, which must be at least one and not more than five, because with a range greater than five, the time required for data generation increases dramatically. Other parameters are number of students and number of teachers that determine how many students and teachers will be generated. Additionally, if required, it is also possible to enter path to files containing input values for data generation. Generated data is output in the form of multiple .csv files, each of which represents one database entity [2].

Address generator part is used for creating one of the input files for data generation. This part is used to generate relatively real addresses that will be stored in a file. Before generating, it is necessary to enter the number of addresses to be generated. Additionally, it is still possible to enter four coordinates, which will represent the area from which the data will be generated. The last parameter that can be entered is the path to the output file. During address generation, each generated address is immediately saved at the end of the selected file. If the file does not exist, the generator will create it. Generation can be interrupted at any time without fear of losing the generated addresses [2].

V. IMPLEMENTING APPLICATION

The implementation chapter deals with the description of the application, problems during its creation and their solutions. The application is currently implemented in the Slovak language, but it is planned to be translated into various languages including English language.

A. User Accounts

One of the main requirements for the application is to secure access to the application using user accounts whose passwords are maintained in the database in the form of a hash. Since the application is intended for internal use within the university, we decided to solve the registration automatically when adding a student or teacher to the application, and vice versa, his account will be cancelled when he is removed from the application. This means that the application does not contain any registration form, but only contains a login form and a password change form. Username and password are created automatically when entering a student or teacher. There are three types of users in the application - student, teacher and administrator. Application users are stored in the user_account table, which consists of two columns: username and password [2].

The password is hashed using the APEX_UTIL.GET_HASH() function. This function accepts two parameters. The first parameter is the string itself to be hashed. To increase security, the string to be hashed is "salted". This means that it will be concatenated with another string that contains various random characters and at the same time this string is also used to verify the user's password. The second parameter of the function is set to NULL. This parameter is optional. If it was not set to NULL, the password would be "salted" from the data about the user's current session, which we would have to save during registration along with the name and password. So, the parameter is set to NULL because we want to use our own "salting" of the password [2].

Fig. 1 shows Oracle SQL script for creating accounts for administrator and all teachers and students in database. Teachers are stored in table "ucitel" and students are stored in table "student". Usernames of students are created by concatenating letter "S" with their personal number and usernames of teachers are created by concatenating letter "U" with their personal number. Personal numbers are stored in column "os_cislo". Passwords of teachers and students are created by concatenating their first name with their surname. Name is represented by column "meno", and surname is represented by column "priezvisko". String, that is used for salting is "LJ_7Hp1z3(y9cMxrm6_.ECpjH*S#4#hY)". Administrator is also created with username "A1" and password "admin".

```
DELETE FROM user_account;

INSERT INTO user_account(username, password)
VALUES('A1',
APEX_UTIL.GET_HASH(apex_t_varchar2 ('admin',
'LJ_7Hp1z3(y9cMxrm6_.ECpjH*S#4#hY)'), NULL));
```

```
INSERT INTO user_account(username, password)
SELECT 'S'||os_cislo,
APEX_UTIL.GET_HASH(apex_t_varchar2
(meno||priezvisko,
'LJ_7Hp1z3(y9cMxrm6_.ECpjH*S#4#hY)'), NULL)
FROM STUDENT join OS_UDAJE using(id_osoba);

INSERT INTO user_account(username, password)
SELECT 'U'||os_cislo,
APEX_UTIL.GET_HASH(apex_t_varchar2
(meno||priezvisko,
'LJ_7Hp1z3(y9cMxrm6_.ECpjH*S#4#hY)'), NULL)
FROM UCITEL;
```

Fig. 1. Creation of new accounts for all users in the application

The authorization scheme determines the type of user and thus also his powers in the application. User type validation is done by executing an SQL query that returns all users whose name is the same as in the APP_USER application variable, which is set to the username specified when the user logs in, and at the same time those users' names start with the character "S". If this SQL query returns any result, the logged-in user is considered a student [2].

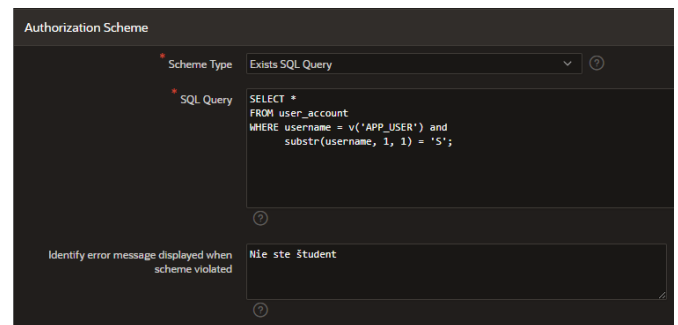


Fig. 2. Student Authorization Scheme - determining if a user is a student [2]

In the future, it is planned to use university LDAP accounts for accessing the application.

B. "Študenti" page

This page allows you to view and manage students within the application. The home page contains the students and analysis tabs. On the tab called "Študenti", which means "Students", there is an overview of students by level of study. In addition to students and teachers, the administrator has access to unclassified students and unclassified persons. Unclassified students include students who have been assigned a personal number but have not yet been enrolled in any specific study. Among unclassified people, there are people who do not have a personal number. The personal number will be lost when the student is deleted. These unclassified persons can either be deleted or assigned a study. Both students and persons are displayed in this report in the form of cards, and students can be sorted and filtered [2].

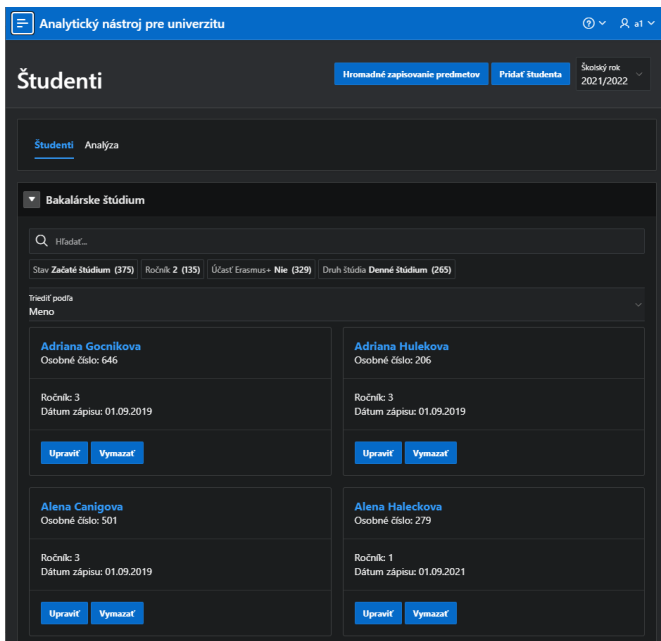


Fig. 3. Student overview page [2]

On the page with information about the student's studies called "Študent – detail", which means "Student – detail", it is possible to find out information about the student's studies [2].

The administrator can manage the student's studies in such a way that he can add and remove years of study, but only if he is in the last year of the student's studies so far. At the same time, the administrator can also close the student's studies, which will then allow him to enrol the student in a higher level of study. When adding a student's year, the administrator has the option to determine how the student completed the current year and the date of its completion, and then he is redirected to a form where he chooses the type of completion, the date of completion and whether the student participates in the Erasmus+ program for the next year in which the student is enrolled [2].

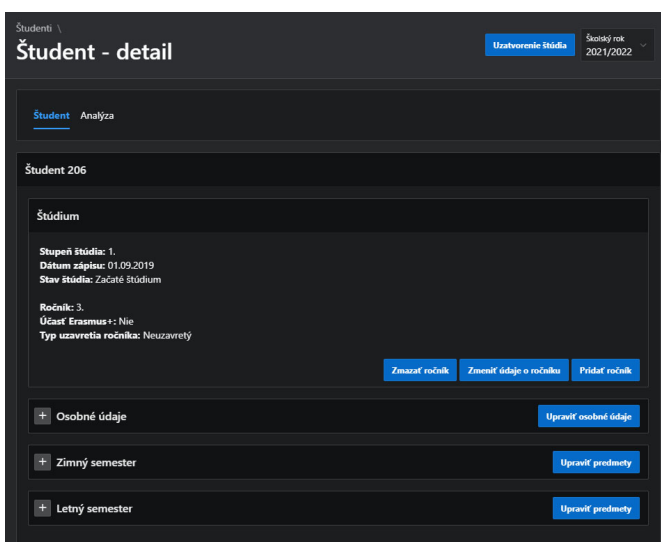


Fig. 4. Page with information about the student's studies [2]

On the this "Student - detail" page, it is possible to browse the student's subjects by semester, and at the same time, it is possible to edit, delete and add his subjects, provided that the student's year has not yet been closed [2].

When enrolling students in subjects, it is first necessary to choose the study group in which they will study in the given semester. The application allows having different study groups for the summer and winter semesters. After selecting a study group, the sections "Aktuálne zapísané predmety", which means "Currently enrolled subjects", and "Dostupné predmety", which means "Available subjects", are displayed. In the section of currently enrolled subjects, the currently selected subjects for the student are listed, and it is also possible to edit and delete these subjects. In the section of available subjects, the subjects that are available for his study program are listed. In this section, subjects can be searched and filtered [2].

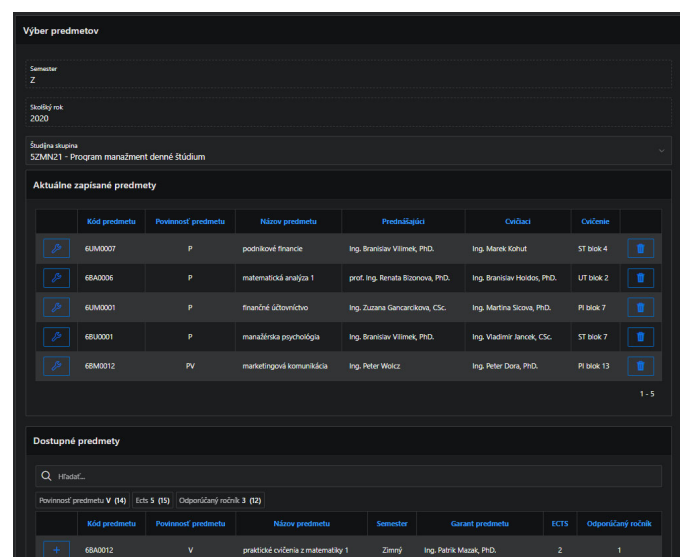


Fig. 5. Page with subject registration [2]

After clicking on the name of the subject that the student is taking at his faculty, from the "Študent – detail" page, the user is redirected to the page about the detailed evaluation of the selected subject. On this page, it is possible to view information about the subject, as well as the results of semester and final exams of the student. These semester and final exam results can be added and removed as long as this subject is not yet closed. After the subject is closed, the resulting grade is calculated. When the subject is closed, its results cannot be changed and this subject can only be reopened by the application administrator [2].

Closing the subject takes place by first checking whether the student has passed any final exam in the subject. If he did not pass any final exam, he will be given a grade of *Fx*. Otherwise, the grade will be set according to the grade from the last final exam [2].

Predmet							
Kód predmetu: 6BM0007							
Názov predmetu: manažérske prezentačné zručnosti							
Garant predmetu: doc. Ing. Marcel Sidor, PhD.							
Počet kreditov za predmet: 5							
Výsledná známka: A							
Stupnica							
A	93-100						
B	85-92						
C	77-84						
D	69-76						
E	61-68						
Fx	0-60						
Výsledky							
Zápočet							
Dátum zápočtu	Počet bodov	Počet bonusových bodov	Požadovaný počet bodov	Celkový počet bodov	Spravil zápočet		
16.12.2019	60/60	0	37	60	Áno		
Skúšky							
Číslo pokusu	Dátum skúšky	Požadovaný počet bodov	Počet bodov za skúšku	Celkový počet bodov	Známka	Skúšajúci	Spravil skúšku
1	22.01.2020	25	39/40	99/100	A	RNDr. Míriam Phaleková, PhD.	Áno

Fig. 6. Subject results page [2]

C. "Učítelia" page

"Učítelia" page, which means "Teachers" page, is similar to "Students" page. Teachers are displayed in the form of cards and can be searched and filtered. After clicking on the name of any teacher, information about the given teacher, his taught and lectured subjects and the number of taught or lectured students per subject will be displayed. Subsequently, the teacher or administrator can also open a specific subject, where he can see the list of students he teaches in the given subject as well as their personal number, level of study, year and result [2].

Učiteľ 84212		
Titul: prof. Ing., PhD.		
Meno: Marian		
Príjmenie: Gáspár		
Katedra: MIMÉ		
Posícia: Vedúci katedry		
Zamestnaný od: 08.05.2015		
Zamestnaný do: 15.01.2022		
Prednášané predmety		
<input type="text" value="Vyhľadávacie: Všetky textové sĺpce"/> <input type="button" value="Prejsť"/> <input type="button" value="Akcie"/> <input type="button" value="Znovunastaviť"/>		
Kód predmetu	Názov	Počet študentov
6BL0001	ekonomické a právne aspekty podnikania	68
6BA0006	matematická analýza 1	141
		Celkom: 2

Fig. 7. Teacher's page [2]

D. "Predmety" page

"Predmety" page, which means "Subjects" page, contains an overview of all subjects that students can enroll in for individual school years [2].

In order to display the list of subjects, it is first necessary to select the degree of study, the study program and the recommended year for taking the subject. Subsequently, it is possible to choose a specific subject. After choosing a specific subject, a page with the subject scale, the success rate of the subject's students and a table containing specific students enrolled in the selected subject will be displayed [2].

Predmety			
<input type="button" value="Vymazať predmety"/> <input type="button" value="Skopirovať predmety"/> <input type="button" value="Pridať predmet"/>			Školský rok: 2021/2022
Stupeň štúdia: Bakalárske štúdium			
Študijný program: informatika			
Odporúčaná ročník: 1			
<input type="text" value="Hľadať..."/>			
	Kód predmetu	Názov predmetu	Počet študentov
<input type="button" value="🔗"/>	6BA0002	algoritmická teória grafov	34
<input type="button" value="🔗"/>	6BA0005	diskrétna pravdepodobnosť	34
<input type="button" value="🔗"/>	6BT0007	teľovychovné sústredenie 1	8
<input type="button" value="🔗"/>	6BI0046	úvod do operačných systémov	12

Figure 8 Subjects page [2]

E. "Študijné skupiny" page

"Študijné skupiny" page, which means "Study groups" page, contains a list of study groups with the number of students. Groups can be sorted and filtered. Study groups are filtered based on degree of study [2].

Študijné skupiny		
<input type="button" value="Pridať študijnú skupinu"/>		Školský rok: 2020/2021
Stupeň štúdia: Všetky stupne štúdia		
Študijné skupiny		
Študijná skupina ↑	Počet študentov	
SZE11	2	<input type="button" value="🗑️"/>
SZE12	2	<input type="button" value="🗑️"/>
SZE21	4	<input type="button" value="🗑️"/>
SZE22	3	<input type="button" value="🗑️"/>

Fig. 9. Study groups page [2]

For each study group, it is possible to display the students who were in the study group in individual semesters [2].

Študenti v skupine 5ZE11 - zimný semester	
Os Číslo	Meno
156	Bc. Helena Orlickova
807	Bc. Beata Bullova
Celkom 2	

Študenti v skupine 5ZE11 - letný semester	
Os Číslo	Meno
156	Bc. Helena Orlickova
Celkom 1	

Fig. 10. A page with students assigned to a study group [2]

F. Analytical tools

The application offers a number of analytical tools, and it is also possible to find various interesting and useful statistics in the application, both about the student and the progress of his studies, as well as about teachers, individual study groups and subjects. The main analytical tools we use in the application include sorting, filtering and graphical display of data and statistics [2].

Sorting and filtering are important parts of any application that works with large amounts of data. In application, it is possible to sort and filter different regions such as tables, cards or, for example, charts [2].

Several graphs are also created in our application, recording statistics about students, teachers and study groups. In the future, graphs can easily be extended to other entities as needed [2].

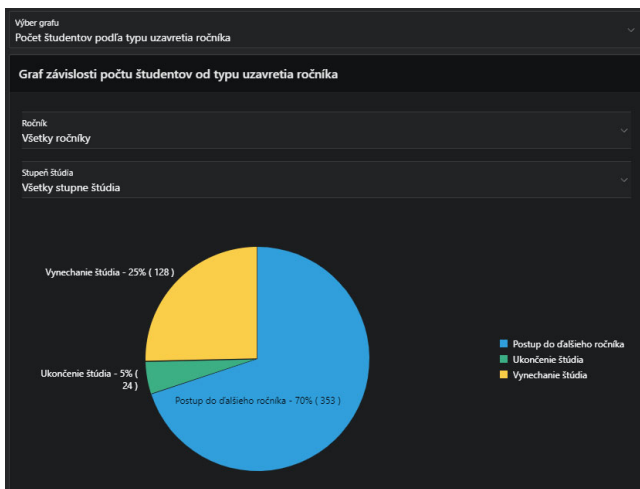


Fig. 11. graph of relationship between the number of students and the type of year closure [2]

Fig. 11 shows graph of relationship between the number of students and the type of year closure. The graph can be found on the “Students” page under the “Analýza” tab, which means “Analysis” tab. This graph shows how many students have completed their studies and how many are still studying. At the same time, it is possible to find out how they finished their

studies or grade. The chart can also be filtered by year and level of study.

Fig. 12 shows graph of relationship between the average number of ECTS and the study group.

The graph can be found on the “Study groups” page under the “Analysis” tab. From the graph, it is possible to read both the success rate of individual study groups, as well as how many ECTS, on average, individual students try to obtain per year. The chart can be sorted by study group and number of ECTS. The chart can also be filtered according to level of study and study program.

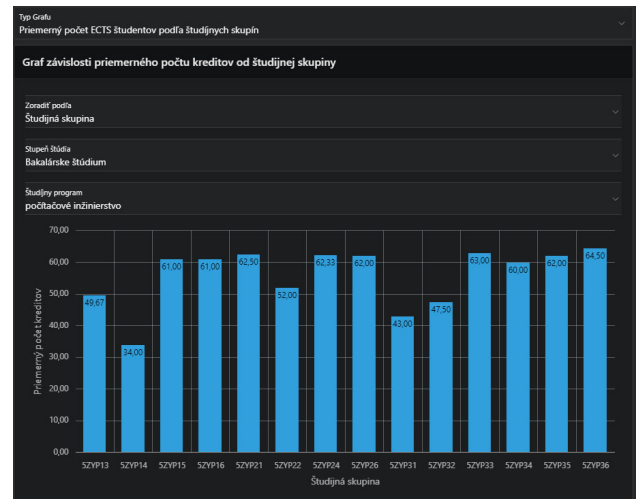


Fig. 12. Graph of relationship between the average number of ECTS and the study group [2]

VI. TESTING

Testing is an important part of application development. This chapter describes the ways in which we tested the data generator, whether the data is coherent and consistent in the database, as well as the ways of testing the application itself [2].

A. Data generator testing

We tested the data generator gradually as we created it. This means that we tested both the creation of data for individual entities, but also for a set of interdependent entities, and we also tested whether the data generator works as a whole. After the completion of the generator, we focused on testing the consistency and coherence of the data between the individual entities of the database. In case of any errors in generated data, thanks to the object-oriented design, the errors could be easily found and corrected. After each fixing of the generator, the modified part was first tested to see if it works independently and then the data generation for all entities was tested to see if the generator still works as a whole [2].

B. Testing data in the database

When entering data into the database, the correct data types, the uniqueness of the primary key, the basic mapping of foreign keys and also constraints of the values defined for some database entities were implicitly verified. After

successfully entering the data into the database, we visualized it during the development of the application and thereby also revealed some data deficiencies, which we adjusted directly in the data generator, which we subsequently tested and imported the newly created data into the database. This entire process was repeated until the data began to make sense or until the data met the minimum requirements to be considered sufficient [2].

C. Testing the Application

The application was tested during the entire creation process. When creating each new page, both the display of data and the responsiveness of the design and the functionality of the buttons located on the page were tested. In case the page works directly with a database through a form or predefined processes triggered after executing our defined action, we also test the changes directly in the database using various queries from the console integrated into the Oracle APEX environment. For forms, we tested both expected inputs and various random inputs to determine if the form correctly handles user input [2].

VII. CONCLUSION

The development of an Analytical Tool for the University Data Management followed the waterfall model of software development, incorporating knowledge from the field of Software Engineering. The software development phases included analysis, requirements gathering, design, implementation, and testing. By adhering to this model, the application development process became more streamlined and efficient.

The analysis phase helped to gain a better understanding of the application's needs and identified shortcomings in existing systems. Both domestic and foreign systems, as well as processes within the Faculty of Management and Informatics of University of Žilina, were analysed. This analysis informed the focus areas for requirements gathering, resulting in a domain diagram and a business use case diagram.

During requirements gathering, efforts were made to determine the application's functionalities and address requirements related to security, support, and user interface. The outcome was a requirements diagram and an application use case diagram. The application was then designed based on these requirements, utilizing both the identified needs and knowledge of faculty processes. The implementation took place within the Oracle APEX environment, with a focus on simplifying management and enhancing security through Oracle Cloud. Through testing, the functionality and correctness of the application were verified.

The application also holds potential for future expansion. For example, analytical tools utilizing machine learning techniques could be added to predict the progress of students' studies or identify students who have similar study patterns. The application could also include manual comparison of students and better automation of certain processes. The system could warn students based on similarities in their studies to other students who were unsuccessful. It could also notify a student if they enrol in too many or too few subjects, indicating an inadequate number of ECTS. The system could recommend to the student which subjects to focus on based on similarities in results with other students.

ACKNOWLEDGMENT

It was supported by the Erasmus+ project: Better Employability for Everyone with APEX (2021-1-SI01-KA220-HED-000032218).



Co-funded by the
Erasmus+ Programme
of the European Union



REFERENCES

- [1] Academic information system of the University of Žilina, Web: <https://vzdelavanie.uniza.sk/vzdelavanie/>.
- [2] I. Pastierik, *Analytical Tool for University Data Management*, Faculty of Management Science and Informatics – University of Žilina, 2023.
- [3] Academic information system of the Unicorn Collage in Prague, Web: <https://unicornuniversity.net/>.
- [4] Academic information system of Comenius University, Web: <https://ais2.uniba.sk/ais/start.do>.
- [5] Academic information system of Technical University in Munich, Web: https://portal.mytum.de/navigation_view.
- [6] M. Kvet, K. Matiaško and Š. Toth, *Practical SQL for Oracle Cloud*, EDIS, 2020.
- [7] Oracle Cloud Infrastructure Documentation, Web: <https://docs.oracle.com/en-us/iaas/Content/home.htm>.
- [8] P. Sakar, *Oracle Cloud Infrastructure for Solutions Architects*, Packt Publishing, 2021.
- [9] A. Nanda, B. Tierney, H. Helskyaho and M. Widlake, *Real World SQL and PL/SQL: Advice from the Experts*, McGraw Hill, 2016.
- [10] Oracle Database SQL Language Reference, Web: <https://docs.oracle.com/database/121/SQLRF/toc.htm>.
- [11] K. Matiaško, M. Kvet, V. Vestenický and V. Šalgová, *Rýchly vývoj dátových modelov a aplikácií v prostredí Oracle APEX*, EDIS, 2020.
- [12] A. Geller and B. Spendolini, *Oracle Application Express (APEX): Build Powerful Data-Centric Web Apps with APEX*, McGraw Hill, 2017.
- [13] A. Nujiten, I. Ellen-Wolff, L. Brizzi, *Oracle APEX Best Practices*, Packt Publishing, 2012.
- [14] *Oracle APEX Documentation*, Web: <https://apex.oracle.com/en/learn/documentation/>.
- [15] A. Png and H. Helskyaho, *Extending Oracle Application Express with Oracle Cloud Features*, Apress, 2022.
- [16] E. Sciore, *Understanding Oracle APEX 20 Application Development*, Apress, 2020.
- [17] Oracle APEX Universal Theme, Web: https://apex.oracle.com/pls/apex/r/apex_pm/ut/getting-started.