A Learning Management System for Tracking and Assessment of Online Team Projects

Hasan M Jamil
Department of Computer Science
University of Idaho, USA
jamil@uidaho.edu

***Abstract*—Online education platforms demand that reliable and measurable assessment and feedback generation systems be developed to ensure quality. In particular, classes requiring large software development projects need step wise requirement specifications, verification and validation to effect proper assessment in phases. Such monitoring and assessment support is virtually non-existent in contemporary tutoring or learning management systems. In this paper, we propose a novel implementation of a team-based project management and assessment system for online database classes.**

***Index Terms*—Requirement elicitation, software specification and validation, mapping specification to requirements.**

I. Introduction

Concepts of data management and processing are fundamental components of computer science curricula, often offered as a multi-part database management systems sequence covering depth and breadth. With the change in technology, application domains and time, the content and focus have evolved . However, despite the dynamic nature and the evolution of content of database courses [7, 8], most instructors included a project-based learning (PBL) component [1, 3, 21]. This choice is firmly grounded in the conviction that PBL improves learning outcome and increases effectiveness [6, 17].

In a first database course, the implementation projects are often an application that most students are familiar with, such as a university, library, a grocery store, or a hospital system database. These projects are chosen to reduce students' cognitive loads and to leverage their familiarity with the application. Students design the application using a database management system such as MySQL, Oracle, SQL Server, or even MongoDB. Usually they are required to design a conceptual model using tools such as entity-relationship (ER), or UML diagrams, then design a 3NF/BCNF scheme following proper normalization before implementing. They are also required to design web-based or graphical interfaces for end-users that serve as front-ends for several queries and updates over the back-end database. Such an exercise typically leverages all the theories students learn in the database course.

A software development life cycle (SDLC) is set in motion when the project starts. In traditional offerings [16], students naturally follow a waterfall model of SDLC [20] as they progress toward the final project delivery at the end of the semester, receiving instructor's feedback on various components along the way, and adjusting accordingly (as in [16]). Such a model is simple and easy to manage manually from an instructor's point of view, but not desirable as a learning experience or from an evaluation standpoint for students because students do not have the opportunity to correct their design errors, or re-calibrate their mistakes [15]. An agile SDLC model [20], on the other hand, offers an opportunity to iteratively design and, if needed, refine the implementation using the feedback and validation of the requirements in phases. In this paper, our goal is to design a database course project management and monitoring system using an agile SDLC model.

II. Scaffolding of Database Course Projects

Traditional scaffolding of database course projects include modules such as conceptual design, normalization and querying exercises that are woven into a database application implementation project in the laboratory [9] along with an evaluation scheme [3, 13]. While the projects remain a significant component of the courses, their assessment remains largely manual, even when a laboratory support system is used [5, 6]. While manual assessment is difficult and time consuming in even a face-to-face teaching environment, it is even harder in online settings with support only from intelligent tutoring systems [12], largely because in courses involving PBL components in particular, online does not necessarily imply automation, Unfortunately, automated graders for database courses are rare [11] and almost non-existent for database projects [16].

A. Requirement Specification

A first database class project usually has a description of its schema, queries and updates that need to be implemented. While in the eighties, it was enough to just test the SQL crafting skills of a student [1], in modern courses, we often require students to implement a complete database application with graphical or web-based user interfaces to support the functionalities specified in the project description. A systematic assessment of a project today preferably needs to follow an improved and step wise evaluation described in [16] with online delivery of the course to a larger community, and automation to support scalability. However, no such systems could be found that follow evaluation procedures similar to Pigford's [16].

1) Instructor's View of the Project Specifications: Beyond the project itself, an instructor may be interested in establishing a project plan and timeline, identifying deliverables, regularly consulting with the project team to stay abreast of the progress, offering presentations of the project at different stages of development, documenting and reporting, testing and validating the application at every stage of implementation, and evaluating both the implementation and the presentations.

Admittedly, many of these components are inherently manual and human intensive and cannot be delegated to an autonomous system. Implementing a team database class project warrants a high overhead of both instructor and student time, expertise, and knowledge of development tools. Many students enter a first database class without adequate skills consonant with substantial software development projects. It is thus imperative that instructors prepare the teams, pace the implementation and monitor the progress efficiently, carefully and fairly in the interest of judicious use of everyone's time within the semester.

An online management system for such projects must then consider including tools and features to reduce instructor engagement and time to the extent possible. In ProTrack, we incorporated six management tools to help monitor, manage and assess team projects. They are as follows.

a) Syllabus and Project Description: ProTrack requires linking with an electronic description of the syllabus. In particular, it requires a detailed description of the project, its deliverables, and the metrics and rubric using which the project will be evaluated. This information is used to autonomously engineer the entries of the remaining monitoring and evaluation tools. For example, the project points must be distributed over the deliverables and other evaluation components according to instructor's project progression plans. Figure 1 shows a sample syllabus description form used in ProTrack using which instructors steer the implementation and evaluation plan.

b) Project Plan: While instructors list the deliverables, students design a project plan using the form in Fig 2 in which they are free to organize the development and delivery plan according to their convenience within the parameters of the instructor's requirements. For example, if the instructor suggested the use of XAMPP, and query 4 to be delivered no later than phase II, the student plan must adhere to these requirements.

The project plan also includes team composition, leader designation, primary responsibilities, and phase wise deliverables. Each deliverable in the plan has an evaluation column and a comment/feedback box that are instructor edited on a scale as described in the syllabus. The evaluation is color coded relative to the weight and importance of the deliverables where red indicates unmet requirement, and green means requirement met. Students are able to view these evaluations and comments as feedback to see if they have met the deliverables. Unmet deliverables automatically move to the next phase in the project plan.

c) PERT Chart: Once the project plan is designed, a Performance Evaluation and Review Technique (PERT) is

Syllabus

Project Title:	BlogBase Database for Current Affairs
Total Points:	60
Learning Outcomes:	*By the end of the course, students are expected to be able to model, design, implement and deploy a BCNF/3NF relational database application on a server for the end-users complete with web-based user interfaces.*
Competencies Earned:	*Students will learn to use Flask, MySQL, CKEditor and integration of PDF based newsletter authoring and editing tools.*
Description of the Project:	*affairs. An editorial board will moderate the submissions and thus will have the authority to edit or disallow content written by an author, or retract contents. Readers will be able to comment on articles, and on other readers' comments in*

Requirements:

System	Software
MySQL	CKEditor
XAMPP	WordTune
	PDFCreator
	Faker/Mockaroo

Number of Phases:

4	7th February, 2023	5th March, 2023	3rd April, 2023	10th May, 2023

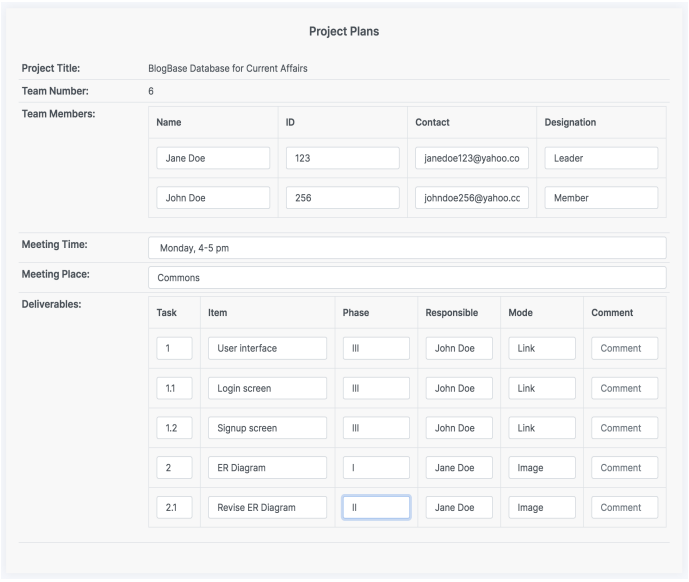
Deliverables:

Item	Phase	Mode	Assessment Standard	Necessity	Weight
ER Diagram	I	Image	Competent	Required	10%
BCNF Model	II	Text	Accomplished	Required	20%
Query 1	Open	Link	Competent	Expected	40%
User interface	III	Link	Accomplished	Required	30%

Assignment Rubrics:

	Assessment			
	0.0.....3	3.1.....6	6.1.....10	10 max
Category	**Developing**	**Competent**	**Accomplished**	
User Interface design	a. Some of the functionalities are implemented. b. The interface did not meet aesthetic standards expected. c. Tool choices were flawed.	a. Choice of tools are reasonable. b. Layout is acceptable and some of the functionalities are implemented.	a. Student delivered a fully functional web interface for user interaction for all the expected functions. b. Made prudent choice of tools. c. No known bug or errors present. d. The layout and color schemes are visually pleasant.	

Fig. 1. Project description in the syllabus linked to project plan and evaluation.



Project Plans

Project Title:	BlogBase Database for Current Affairs
Team Number:	6

Team Members:

Name	ID	Contact	Designation
Jane Doe	123	janedoe123@yahoo.co	Leader
John Doe	256	johndoe256@yahoo.cc	Member

Meeting Time:	Monday, 4-5 pm
Meeting Place:	Commons

Deliverables:

Task	Item	Phase	Responsible	Mode	Comment
1	User interface	III	John Doe	Link	Comment
1.1	Login screen	III	John Doe	Link	Comment
1.2	Signup screen	III	John Doe	Link	Comment
2	ER Diagram	I	Jane Doe	Image	Comment
2.1	Revise ER Diagram	II	Jane Doe	Image	Comment

Fig. 2. Students' project plan based on the project requirements.

generated automatically, but remains editable so adjustments can be made in the project plan. The editing is circular – editing the project plan changes the PERT chart, and vice-versa. There are two renderings of the PERT chart – a two dimensional tabular view, and graph like view showing dependencies among the deliverables and the components. Figures 3 and 4 depict its two renditions.

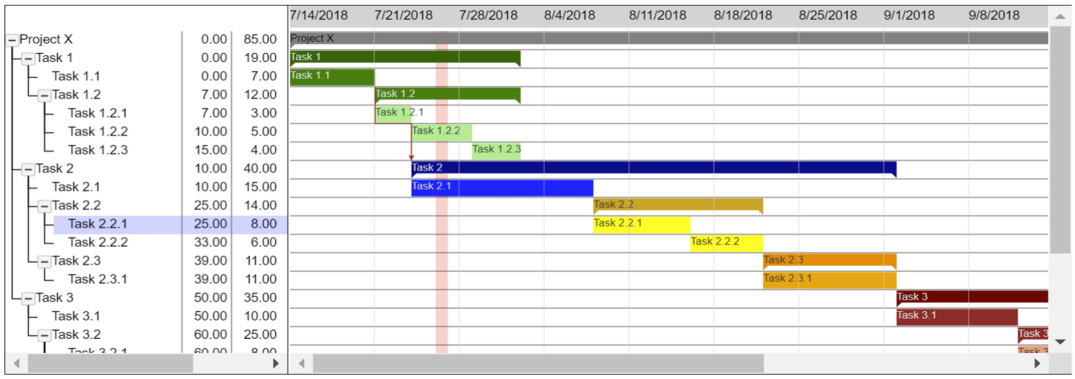


Fig. 3. Drilled-down sub-task level detailed Gantt chart description of the project plan.

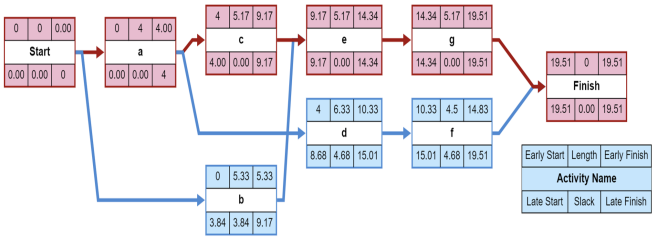


Fig. 4. Alternative PERT chart view of the project plan showing dependencies.

d) Project Calendar of Deliverables: A 16-week calendar is also automatically generated (shown in Fig 5) based on the syllabus, project plan and the PERT chart to show the entire course schedule, including the project. The calendar is not editable by students.

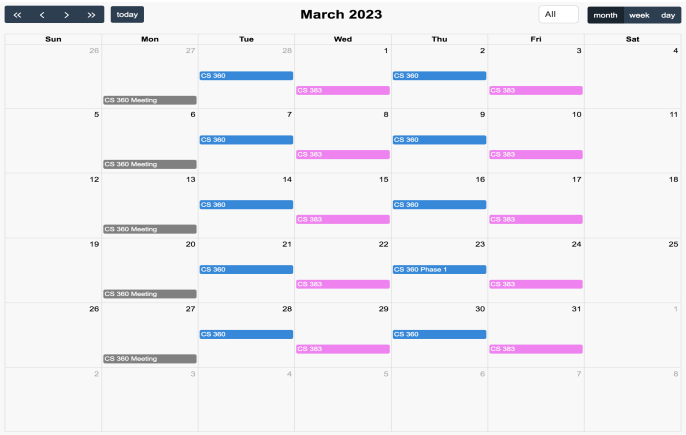


Fig. 5. Project plan embedded in the course calendar.

e) Project Testing, Validation and Evaluation: Testing and validation of the project is the most important and involved component of the entire ProTrack project management system. ProTrack generates a phase wise form for students to fill out against the scheduled deliverables, and the instructor to review. In ProTrack, all deliverables must be web accessible. Largely, students initiate testing and validation of the deliverables by submitting a link to the code fragment that can be executed just by clicking on it. However, submission in other forms is also allowed.

Students optionally are able to report known errors or bugs, and highlights, in the form of strengths and weaknesses of the software segment or system component. Instructors are able to provide comments or feedback and grade the submission on a scale from 1 to 10 displayed, and adjusted relative to the weight of the component. The evaluation, along with instructor feedback, are linked to the feedback panel discussed in section II-A1b. The evaluation panel also has three groups of vertical bars to indicate project progress – for the current deliverable or item, cumulatively for every phase, and overall. In the evaluation for a phase, only the points in a phase are considered and plotted over a 0% to 100% scale with color coding, where red means poor and green means good. Cumulative evaluation shows progress up to a phase including the previous phases, whereas the overall evaluation includes sum total of all phases.

Figure 6 shows the testing, validation and evaluation interface as seen by students. In this interface, students also certify requirements that they are required to meet without any evaluation by checking a box, e.g., use of JavaScript or XAMPP, or Latex on OverLeaf for report writing. Instructors can optionally choose to inspect the student workspace online to review the code base, the database tables, empty the tables, repopulate tables with data generators such as Mockaroo or Faker, or selectively insert or delete rows, and test the system using new data.

2) Student's View of Meeting Project Requirements: Except for a handful, student perspective of the deliverables tends to be linear, i.e., students prefer to find the shortest path from specifications to implementation and a hyper focus on grades, without too much attention to learning outcomes – skills improvement, broadening of system knowledge, etc. Therefore, the agile SDLC model used in ProTrack helps correct these lapses, usually present in traditional waterfall SDLC model based implementation, by sending students back to the 'drawing board' in multiple checkpoints by giving feedback following validation processes supported in the system.

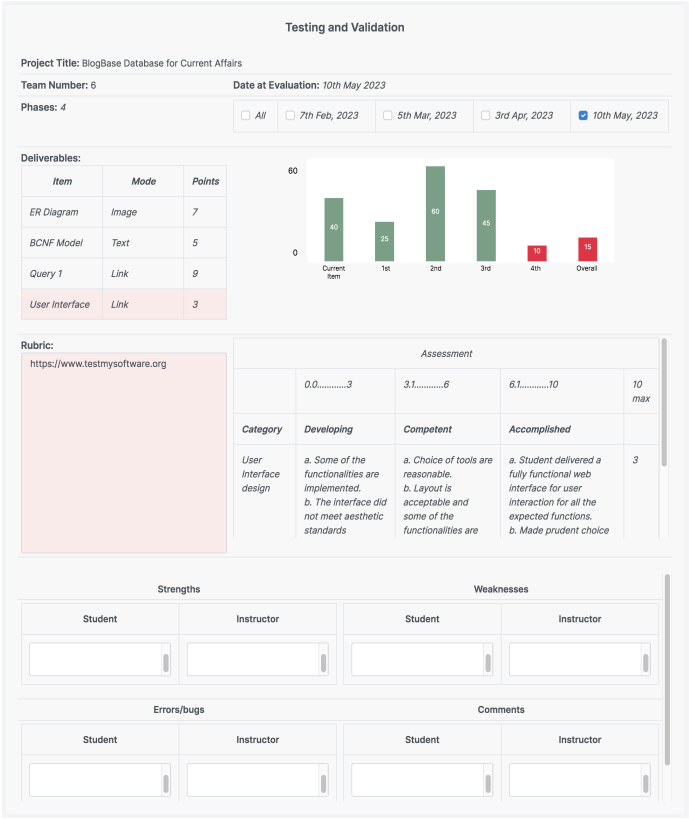


Fig. 6. Multi-phase inter-linked project evaluation and feedback system report.

B. Rubrics as Guidance and Assessment Tools

Design and use of rubrics in CS course projects offer clarity of expectations, and direction on design goals to significantly enhance learning experiences and boost student confidence [4, 19]. In section II-A1a, the rubric we have used in Fig 1 is a simple one. A whole body of research exists on the science and art of rubric design. We plan to address this issue in a separate and independent research in the future.

III. Automated Testing and Validation

Even when a description for a database project is available, each team's approach to its implementation is unique. Therefore, designing a generalized and automatic validator is not yet feasible. Additionally, the human computer interaction (HCI) component of the application is inherently human dependent and needs to evaluated manually for aesthetics, usability and accuracy. For example, the discrepancy or inaccuracy in a user interface with respect to the query it executes in the back-end cannot be easily detected. This is because the query depends on the underlying scheme and the mental transformations the designer makes on the interface variables and interface conditions before mapping them on to the query. A complex series of such requirements to implementation translations make any automation unfeasible, It is perhaps the very reason that no known database project management system for monitoring and assessment exists.

In ProTrack, we remain close to the spirit of Pigford [16] and design an online computerized edition of her model. We note that despite the availability of many automation and monitoring technologies in software engineering, management of laboratory projects, testing and validation, and thus their assessment remain largely manual. However, we see opportunities to achieve partial automation in validation, and transitively, authentic assessment. In the sections to follow, we explore a possible approach to automating the validation and assessment of a database project against a project specification.

A. Query Equivalence

One of the key ingredients needed to test if an implementation satisfies the application specifications is to establish equivalence of a reference query Q_r and a test query Q_t [14], where query Q_r is known to satisfy the specification or requirement, i.e., $Q_r(D) \equiv Q_t(D)$ and they both return identical responses over any and all instances of a database D. Research show that such a determination is theoretically intractable though for a small subset of query classes solutions exist [11]. However, no solutions known to exist when two databases are semantically equivalent, i.e., $D_r \equiv D_t$, and if $Q_r(D_r) \equiv Q_t(D_t)$ could be tested. This is important because the reference query Q_r is defined over the database D_r while the student designed database D_t is arguably different.

B. Alternative to Query Equivalence

Similar to testing queries for equivalence, testing is two database schemas express identical information is also very hard [2] in general. Therefore, it is necessary that we find an alternative but effective method to validate a design against a requirement. We approach the solution as follows.

1) Requirement Specification: Requirement specification for database design is rare and only one single early academic citation could be found [18]. The requirements addressed in the article are truly elementary and completely manually managed by experts. In the interest of scale, we need an approach amenable to automation. We believe that designing a transformation function from a semi-formal requirement description to ER diagram is feasible. At least, an instructor can specify the ER diagram as a starting point, along with a set of functions as query specifications.

Once an ER diagram is available as a reference, establishing equivalence with a student design is simpler [10], which they are required to complete as part of the project. Once schema equivalence is established, all we need to do is ensure that ER to relational schema transformation by the student is correct, for which algorithms exist. Similarly, sophisticated algorithms exist to transform a relational schema in to 3NF or BCNF, and test for correctness of a translated schema.

2) Validation: We then use a text to SQL translation algorithm over the student's normalized scheme to obtain Q_r for each query, and use student queries as Q_t. Instead of testing for equivalence algorithmically, we execute both Q_r and Q_t over the student database D and check to see if $Q_r(D) = Q_t(D)$, and thereby validate the project requirements.

C. System Assumptions

The primary goal of ProTrack is aiding instructors to effortlessly manage and monitor the progression of a large number of course projects online as automatically as possible to save instructor time. We therefore adopt a "best effort" principle and offer to craft solutions as a suggestion. The suggestions complement the presentations and demonstrations students are scheduled to make anyway. The ProTrack suggestions are then used as investigative tools as opposed to evaluation tools to reasonably validate the project requirements and help assess their merit. Naturally, we assume that the ER schema generation and equivalence testing tool are sound. Similarly, we assume that ER to relational schema mapper and 3NF/BCNF testing algorithms are correct as well. Finally, we assume that the text to SQL query translator is also accurate.

IV. Conclusion

Research has established that students learn abstract concepts of SQL best by doing [1], and PBL is one of the best ways to ensure a solid learning outcome [6]. Despite this , there remained a void for a project management system to efficiently and effectively monitor and assess database course projects, until now.

The ProTrack system tracks the developments in [16] and improves the functionalities significantly in order to make it scalable in the internet age for online delivery of database courses, manage multi-phase database course projects and support effective assessment. Though ProTrack follows Pigford [16], it stands in contrast with it in many ways. First, it completely and automatically personalizes the forms used in project description, monitoring, and performance evaluation. Secondly, it offers investigative tools requirement validation, and supports real time testing of the user interfaces and query executions, features that are completely manual in [16]. Not only is ProTrack fully digital and online, it also supports translation tools from project requirements to ER scheme generation, testing equivalence (or similarity) of two conceptual schemes, mapping conceptual schemes to 3NF/BCNF, and mapping natural language queries to SQL, and significantly expedites evaluation.

Though ProTrack is designed to help manage database class projects online, it also aims to support automated grading of various components of a database design project spread over multiple interlinked implementation phases. It does so by internally linking all the project deliverables in a PERT chart with switches to actively generate feedback indicating if the project requirements have been met, making it easier for both the instructor and students to stay in sync, and informing students of their standing in real time. However, a more automated validation of student implementation remains an active research focus that requires substantial theoretical development, as part of our future efforts.

Acknowledgement

The author thankfully acknowledges that the ProTrack web interfaces were prototyped by Kallol Naha.

References

[1] J. Atkins and M. Henry. A database management system project for an undergraduate database design course. In *ACM annual conference on The range of computing: mid-80's perspective, Denver, Colorado, USA, October 14-16*, pages 266–270, 1985.

[2] C. Beeri, A. O. Mendelzon, Y. Sagiv, and J. D. Ullman. Equivalence of relational database schemes. *SIAM J. Comput.*, 10(2):352–370, 1981.

[3] A. N. Borodzhieva. Using project-based learning in the course "databases" on the topic "relational algebra". In *MIPRO, Opatija, Croatia, May 23-27*, pages 1369–1374, 2022.

[4] M. L. Cruz, G. N. Saunders-Smits, and P. Groen. Evaluation of competency methods in engineering education: a systematic review. *Eur J. of Engg. Edu.*, 45(5):729–757, 2020.

[5] M. Cvetanovic, Z. Radivojevic, V. Blagojevic, and M. Bojovic. ADVICE - educational system for teaching database courses. *IEEE Trans. Educ.*, 54(3):398–409, 2011.

[6] C. Domínguez and A. J. Elizondo. Database design learning: A project-based approach organized through a course management system. *Comput. Educ.*, 55(3):1312–1320, 2010.

[7] J. R. Driscoll, P. A. Honkanen, W. A. Shay, and J. C. Peck. Database courses with realistic student projects (panel session). In *ACM SIGCSE, USA, February 17-18, 1983*, page 124, 1983.

[8] M. Goldweber, M. Wei, S. G. Aly, R. K. Raj, and M. F. Mokbel. The 2022 undergraduate database course in computer science: what to teach? *Inroads*, 13(3):16–21, 2022.

[9] E. P. Holden. Assessment of an introductory database course: a case study. In *ACM SIGITE, Cincinnati, OH, USA, October 16-18, 2008*, pages 131–138, 2008.

[10] S. Jajodia, P. A. Ng, and F. N. Springsteel. The problem of equivalence for entity-relationship diagrams. *IEEE Trans. Software Eng.*, 9(5):617–630, 1983.

[11] M. Karimzadeh and H. Jamil. An intelligent online SQL tutoring system. In *IEEE ICALT 2022, Bucharest, Romania. July 1-4, 2022*, pages 212–213, 2022.

[12] C. Kenny and C. Pahl. Automated tutoring for a database skills training environment. In *ACM SIGCSE, St. Louis, MO, USA, February 23-27, 2005*, pages 58–62, 2005.

[13] R. R. Leeper. A project course in database. In *ACM SIGCSE, Washington, DC, USA, 1990*, pages 78–80, 1990.

[14] A. Y. Levy, I. S. Mumick, Y. Sagiv, and O. Shmueli. Equivalence, query-reachability, and satisfiability in datalog extensions. In *ACM PODS, May 25-28, USA*, pages 109–122, 1993.

[15] C. Lüer. Transition from a waterfall-based capstone course to an agile model. In *FECS, July 14-17, 2008, Las Vegas, Nevada, USA*, pages 461–467. CSREA Press, 2008.

[16] D. V. Pigford. A management system for monitoring and assessing the group-oriented database project. In *ACM SIGCSE, USA, February 19-20, 1987*, pages 9–18, 1987.

[17] B. K. Seyed-Abbassi. A SQL project as a learning method in a database course. In *ACM SIGCPR, St. Louis, Missouri, USA, April 1-3, 1993*, pages 291–297, 1992.

[18] N. C. Shu, H. K. T. Wong, and V. Y. Lum. Forms approach to requirements specification for database design. In *ACM SIGMOD, USA, May 23-26, 1983*, pages 161–172, 1983.

[19] M. Souza, E. Margalho, R. M. Lima, D. Mesquita, and M. J. Costa. Rubric's development process for assessment of project management competences. *Education Sciences*, 12(12), 2022.

[20] T. Thesing, C. Feldmann, and M. Burchardt. Agile versus waterfall project management: Decision model for selecting the appropriate approach to a project. In *CENTERIS/ProjMAN/HCist, Vilamoura, Portugal*, volume 181 of *Procedia Computer Science*, pages 746–756. Elsevier, 2020.

[21] S. Xu, L. F. Pollacia, and S. Lai. Analysis of students' database design skills in capstone projects. In *ACM SE, Virtual Event, USA, April 15-17, 2021*, pages 199–203, 2021.