International Journal of Innovation and Research in Educational Sciences
Volume 2, Issue 2, ISSN (Online): 2349–5219

# Projects in Information System Design and Fostering Learning in Israeli High Schools: A Pilot Study

**Nayif Awad**
Science and Technology Education,
Ben-Gurion University of the Negev, Israel.
Science and Mathematics, Sakhnin College, Israel.
Email: awad_nayif@yahoo.com

**Moshe Barak**
Science and Technology Education, Ben-Gurion University of the Negev, Israel

*Abstract* – **This paper reports about a study for creating a learner-centered environment in the context of teaching information system design (ISD) in high school. The participants were 40 students from two schools who prepared individual graduating projects, for example, an information system for a small business or a medical clinic. Data were collected through observations in the class, interviews with students, analysis of their products and portfolios, and feedback questionnaires.**

**The study revealed that giving the student the opportunity to choose projects that were meaningful to them personally or related to their family business fostered their motivation on the task. The majority of students greatly improved their programming skills related to ISD, but many of them only partially completed all the design phases, namely, inquiring into the customers' needs, and planning (for example, drawing data flow diagrams). Two factors influenced this situation. Firstly, the fact that the students had some basic knowledge and experience in programming before they approached the graduating project, but knew very little about the ISD process. Secondly, while the students worked on programming in an interactive advanced technological environment, they performed the system design manually, for example, drawing design diagrams using a word processor.**

**To foster a constructivist-learning environment in information system development, we suggest teaching concepts of information system design at early stages in high school. We recommend starting to engage students in small, open-ended design tasks and gradually increasing the scope and complexity of these assignments.**

*Keywords* – **Design, Contextual Learning, Information System, Programming, Project.**

## I. Introduction

### *Background*

Educators are increasingly recognizing that one of the major objectives of education in general, and science and technology education in particular, is to promote an individual's general skills, such as independent learning, problem-solving, critical and creative thinking, and teamwork, beyond teaching a specific subject matter. To foster students' higher-order cognitive skills, instruction in school should shift from traditional teacher-instructed lessons to more constructivist-guided schooling, such as problem-based learning and project-based learning [4], [5], [31]. Several researchers [16], [25], [29] have acknowledged the advantage of the project method in computer sciences for producing graduates who are competent information technology practitioners over the teaching of specific language syntax or programming methods in an in-depth fashion. However, introducing projects into the high school curriculum is not an easy task, for several reasons. Firstly, this instructional approach requires shifting most of the responsibility for learning from the teacher to the student; secondly, preparing a comprehensive project requiring the integration of knowledge from different domains is exceptional in traditional schooling; and thirdly, engaging students in challenging open-ended tasks is often in contrast with preparing them for the obligatory pencil-and-paper exams common in many educational systems. Reference [14] specifically points out that managing project work in this area is particularly problematic because "computer science projects are often complex, marrying design, human communication, human-computer interaction, and technology to satisfy objectives ranging from consolidation of technical skills through provoking insight into organizational practice, teamwork and professional issues, to inculcating academic discipline and presentation skills".

Despite the fact that projects in computer sciences present a challenge for both students and teachers, many Israeli computer science teachers encourage their students to prepare a graduating project in their final year of high school. In these projects, students deal with the task of developing an information system for a customer, such as a small business or a medical clinic. The product development fuses two processes: conceptual system designing according to the customer's needs and given constraints, and building the system in practice through programming using tools such as Visual Basic or C.

This study addresses a case of applying the project method in the context of teaching computer sciences in high school, with a focus on information system design. The general objective of the research was to explore how the students dealt with the task of information system development and the contribution of this assignment to fostering their system design and programming competencies.

### *The Process of Developing an Information System*

The literature describes an information system as composed of people, machines and methods organized to collect, process, transmit and disseminate data that represent a user [1]. For example, an information system for a business application frequently handles different kinds of data about customers, suppliers, employees, stock management, accounting, etc. Following are some common concepts in the curriculum for teaching information systems design at secondary and tertiary levels [33]: The nature of systems, natural systems versus man-made system, automated systems; roles in designing an information system, such as users, management, system designers, programmers and operations personnel; structure charts; Data Flow Diagrams (DFD); State-Transition Diagrams (STD).

*Models of information system development*

In the literature, the process of designing a new artifact or system is often presented as consisting of the following stages (in various variations): 1) identifying a problem or a need; 2) researching and setting specifications for the required solution; 3) generating alternative solutions and choosing the optimal one; 4) planning; 5) implementing; 6) testing and evaluating; and 7) improving. However, educators have understood increasingly that design is not a linear process as it appears from this seven-stage list, but rather a more iterative process that often involves moving backwards and forwards from one point to another in the design process.

Students in school learn the linear model of information system development, often called the Project Life Cycle [7], [33], and teachers rarely relate to other models that is discussed in the professional literature, such as the V-model or the Spiral model, which better demonstrate the different cycles of developing a project. The question of how students handled the different phases of system development is addressed later in this paper.

*The role of drawings and sketches in information system design.*

Diagrams, drawings and sketches are important tools for information systems designers, as well as in many areas such as mechanical engineering, electronics, architecture and computer science [10]. All of the above-mentioned terms, according to [34], mean "a simplified and structured visual representation that shows entities and relationships representing the architecture or implementation of a software system." These diagrams might represent any of the architectural aspects of a software system, e.g., class inheritance, data flow, flow charts, state machines, sequence diagrams, database tables and relationships, architectural layers diagrams, and relationships between servers and clients. Drawings, diagrams and sketches play a major role in communication as they externalize internal thought, making it visible to the individual and to others, reifying the mental model for others to act upon [32]. Certainly, many would agree with the proverb that sometimes a "picture is worth a thousand words." However, while professional designers today do all the design and drawing by sophisticated technological tools, this is not always the situation in schools, as shown in the following parts this paper.

*Issues related to successful programming*

As previously stated, a significant part of developing an information system is programming in various languages, such as C or Visual Basic. The literature points to a range of roles and guidelines for good programming, such as using global variables sparingly, using library functions, replacing repetitive expressions by calling on a common function, identifying bad input and recovering it if possible, ensuring that input does not violate the program's limits, ensuring that all variables are initialized before use, using recursive procedures for recursively defined data structures, and using data arrays to avoid repetitive control sequences [21]. Later in this paper, we will relate to students' performance in programming.

## II. Conceptual Framework

*The Constructivist Learning Theory*

The psychological and educational literature from the past century has been greatly influenced by the constructivist learning theory, which was developed by leading figures such as [9], [26]. Cognitive-constructivism asserts that individuals actively construct new knowledge, meaning and understanding from their experience, rather than obtaining knowledge from external resources, for example, the teacher. Theories of situated cognition [8] contend that knowledge is inseparable from the contexts and activities within which it is acquired. Learning occurs when learners process new information or knowledge in a meaningful way that makes sense within their own frames of reference, their inner world of memories, experiences and responses.

*Learning information systems design in a constructivist-based approach*

Reference [27] specifically writes about the advantages of shifting information system education from traditional instruction to a constructivist pedagogy, or what they call, from the 'teaching paradigm' to the 'learning paradigm.' In the 'learning paradigm' according to these authors, the aim of the teacher is not to convey subject matter to the students but "to create environments that bring the students to discover and construct knowledge for themselves, to make students members of communities of learning that make discovery and solve problems" (p. 170).

Applying a 'learning paradigm' in school also requires a change in the focus of assessment from examinations aimed at measuring achievements to promoting learning and providing formative feedback. Assessment, in this view, is made by following the learning process and portfolios rather than by final exams. Reference [11] points to the advantages of using principles derived from the constructivist epistemology to provide the learner with the knowledge and higher-order skills necessary to understand and perform database analysis and design effectively as a professional practitioner. These authors suggest a range of guiding principles for designing such a learning environment: allowing learners to choose an authentic project grounded in professional context; encouraging learners to take responsibility for learning and be aware of knowledge construction; allowing learners to develop their own process to reach a solution; providing learners with the opportunity to experience and appreciate other experiences through interaction and collaboration; providing learners with feedback and support; and encouraging reflection in the class.

In the following parts of this paper, we will show how these views of learning and assessment were put into

International Journal of Innovation and Research in Educational Sciences
Volume 2, Issue 2, ISSN (Online): 2349–5219

effect in the current study.

*Contextual Learning*

The term contextual learning has to do with learning that relates to a learner's diverse life contexts such as at home, during leisure time, social or environmental activities, or in the workplace [12]. Contextual learning is reality-based learning that provides the students with opportunities to make meaning of their disciplinary knowledge and solve problems within a real-world context [20]. Contextual learning is not only about what students learn but also about how they learn. Educators are increasingly recognizing the educational advantages of instructional strategies such as problem-based learning and project-based learning (PBL) in science and technology in which students work in teams to investigate or solve real life questions and problems [28].

*Project Based Learning (PBL)*

As previously mentioned, the current study addressed a case of engaging students in designing and constructing an information system within the framework of learning computer sciences in high school. Project-Based Learning (PBL) in science and technology is a learner-centered instructional method that engages learners in cycles of investigating, planning, constructing, evaluating and improving a useful artifact or system.

The application of PBL in the classroom is often a difficult task, for several reasons. If the problems presented to the students are too well structured, closed-ended or simple on the one hand, or too abstract or unrealistic on the other, no contextual learning is achieved and students become frustrated or discouraged. If the problem or question that drives a project is not designed to make a connection between activities and the related underlying conceptual knowledge, students are caught up with 'doing' with only little significant learning [6], [13]. Reference [30] shows that the problem-solving process requires dealing with five larger elements: procedural fluency, conceptual understanding, strategic competence, adaptive reasoning and productive disposition (see also [22]). These five elements are not developed quickly and easily, and learning environments must be organized carefully through years of instruction to meet this challenge.

*A Technological Learning Environment*

Technology offers powerful tools for the realization of a constructivist-learning environment by means of simulating real-life situations or connecting the classroom context to the outside world, such as the community, business or practitioners in science and technology. Many authors showed that a computer-based learning environment typically involves the use of numerous self-regulatory processes, such as planning, knowledge activation, metacognitive monitoring and regulation, strategy deployment and reflection [2], [3], [19], [23]. Technological environments that support learning are often presented as 'cognitive tools' because these technologies can assist learners not only in accessing information, but also in representing ideas, developing ideas and collaboration with others. Likewise, computer and information technologies are also regarded as 'metacognitive tools' since they can guide students in making decisions regarding their learning goals, identifying how much support is needed from contextual resources, and effectively receiving and using feedback from their tutors and peers or from technological means. Although this brief review of the role of technology in fostering learning and cognitive development is taken from the general literature on using technology in education, it is very closely related to teaching and learning concepts of information system design. In this field, computers are simultaneously the subject learned, such as programming, as well as the means for design, problem solving, idea expressing and collaboration between team members.

## III. Research Objectives and Guiding Questions

So far, we have seen that introducing projects on information system design into the school curriculum is well grounded in the literature on reform-based learning and instruction. However, as noted earlier in this paper, computer science projects are often complex tasks requiring the consolidation of theoretical knowledge, social aptitudes and technical skills, and it is not easy for teachers to apply this instructional strategy in traditional schooling. Therefore, the research was guided by the following questions:

1. How do students handle the different phases of designing an information system?
2. What are the factors that help or hinder students to accomplish the information system design?

## IV. Methodology

*Participants and Setting*

The participants in this study were 40 12th grade students (aged 17-18) from two cities in northern Israel. Each student prepared an individual graduating project in information system design and attended a final oral exam. More details on the content of the projects and the working stages are provided in the Findings section. All of the students who majored in computer science were considered high-achievers in their schools and studied concurrently advanced mathematics and physics courses

*Methods of Data Collection*

The research adopted a qualitative approach in order to expose as many learning patterns as possible, mainly the students' actions and thoughts during their work on the projects. Researchers [24], [18] specifically discussed the advantages of qualitative research in the areas of computer sciences and information systems development. These authors show that the qualitative approach helps in understanding social and educational phenomena from the point of view of the participants, and its particular social and institutional context. In the current study, the students also answered a closed-ended questionnaire. Data collection included:

1. Observing the students working on their projects in the lab and talking with them freely; taking notes about the

students' work patterns and their spontaneous responses to questions.

2. Keeping records of the students' drawings and computer programs.

3. Interviewing a sample of 12 students, six from each school. The teachers selected the interviewees, to represent a sample of low, mid-level and high achieving students. The students were interviewed individually or in pairs twice – in the middle of the school year and upon completion of their projects. In these 15-20 minute interviews, the students were asked questions such as how they worked on their projects, what difficulties they encountered, who helped them, and what they thought about preparing a project in computer sciences.

4. Discussing issues with the teachers such as what are the objectives of integrating the project method into the school, what are the impacts on the students, and what the teachers felt their role was in the process.

5. Administering a closed-ended questionnaire in the classes, in which the participants marked their level of agreement for 23 statements related to issues such as independent learning and learning theory in project work. The students filled in this questionnaire twice: first, after about three weeks of working on their project, and then in the middle of the project work. More details on the content and development of the questionnaire are provided in the Findings section.

6. Administering an open-ended summative questionnaire to the students upon completion of their projects.

7. Collecting copies of 12 portfolios that the students had prepared on their projects at the end of the school year and analyzing their contents.

8. Observing the final examinations the students attended on their project (in one school), and holding discussions with the examiners and the students during and after the exam.

Both of the teachers were university graduates in computer sciences, with over three years of experience in guiding students in preparing graduating projects in computer sciences. One of the researchers, the co-author of this paper, was the teacher in one of the schools. In our opinion, the fact that the research took place during regular school hours was advantageous because it increased the likelihood that the findings would be authentic and reflected the reality of implementing the project method in teaching computer sciences in Israeli high schools.

## V. Findings

### *How the Students Selected Their Project Topics?*

At the beginning of the school year, the teachers explained the aims of the graduating project to the students and asked them to suggest their ideas for project topics. The teachers took notes about the way each student chose his project topic, and the students were later asked about this point in the interview. Of the 40 students, eight selected a topic they had already dealt with in the small assignments they had prepared in $10^{th}$ or $11^{th}$ grade, seven accepted the topic the teacher had suggested to them, and four said that they decided to work on a certain topic because it sounded interesting to them, without knowing too much initially about the subject matter. For example, a student decided to develop an information system for a travel agency because she thought it would be interesting to deal with flights, hotels and international tourist attractions. The remaining 21 students decided to develop an information system on a topic that was of personal interest to them or to their parents' vocation or business. For example,

- One student worked on an information system for a small garage where he was working part-time as an apprentice.
- Another student, who was taking driving lessons at the same time, decided to develop an information system for a driving school.
- A third student worked on an information system for her father's dental clinic.
- A fourth student developed an information system for a bank where his mother was working.
- Another student constructed an information system for a school library.
- A sixth student designed an information system for his family's clothing store.

The fact that the students worked on topics that interested them personally very positively influenced their motivation to cope with the task. In the interviews, the students made the following comments:

*"I made a great effort."*

*"I did not join my parents on vacation... instead I stayed to work on my project."*

*"I worked all the night on my project."*

At the year's-end, the teachers concluded that although the projects mentioned above were among the most successful in the class, these topics were often too complicated for the students to learn in-depth. Once the students realized this, they settled for making only a partial or superficial design of their system, as will be clarified in the following sections.

### *How the Students Addressed the Conceptual Design of Their Information System?*

The students were required to carry out an investigation of a customer's needs and prepare a document called a 'Project Charter,' which is based on a standard template consisting of the following elements: 1) background of the project; 2) customer's needs; 3) the changes or improvements required in the existing system; 4) aims and objectives, in terms of detailed measurable results; 5) criteria of success; 6) consequences of failure; 7) constraints and factors that might limit the planning and how to accommodate them; and 8) possible risks, their probabilities and how to overcome them. Some of the students were very serious about investigating a customer's needs. For example, a student who developed an information system for a travel agency visited several travel agents and interviewed local employees. The student learned about the types of information handled by the travel agent, such as flights, hotels and car rentals. He also studied the interface screens the users require in working with the system, and the kinds of reports and

International Journal of Innovation and Research in Educational Sciences
Volume 2, Issue 2, ISSN (Online): 2349–5219

printed documentation the system must provide. In contrast, other students only partially investigated the topic they were working on. Of the 40 students, 10 prepared a project charter that sufficiently covered all the required points, 16 prepared a document that only partly answered the demands and needed improvement, and 14 students prepared a very superficial or incorrect document. Since this stage of the project had only little to do with 'hands-on' work using computers, the students who majored in computer sciences regarded this stage of their project as a formal matter rather than as an essential part of the system development. In addition, many felt it was legitimate to estimate or visualize the data for their system, as they used to do in the 10th and 11th grades when solving small programming exercises. This point also appeared in the way the students sketched their systems' designs, as described in the following section.

### *How the Students Drew Diagrams and Graphs of Their Information System?*

The design of an information system takes place by drawing several diagrams and charts that present, for example, the architectural structure of the software system, its class inheritance, data flow, state machines, sequence diagrams, database tables and relationships. A systematic design of a software system is important not only for obtaining reliable and efficient software, but also for enabling communication between all individuals or teams involved in the system development, implementation or future upgrade**.** Yet, did the students in the current study internalize this view of a project in computer sciences?

Figure 1 illustrates an example of a diagram called a Data Flow Diagram (DFD), which is a graphical representation of the flow of data through an information system. A student whose project was about enhancing an existing information system in a small computer store prepared this chart and included it in the booklet he prepared of his project.

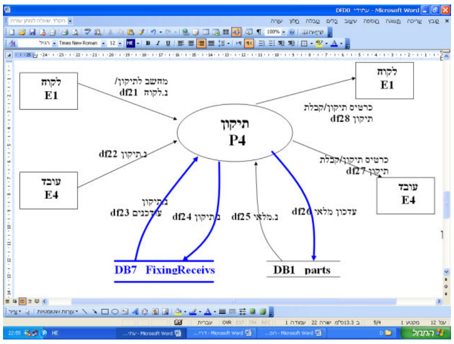
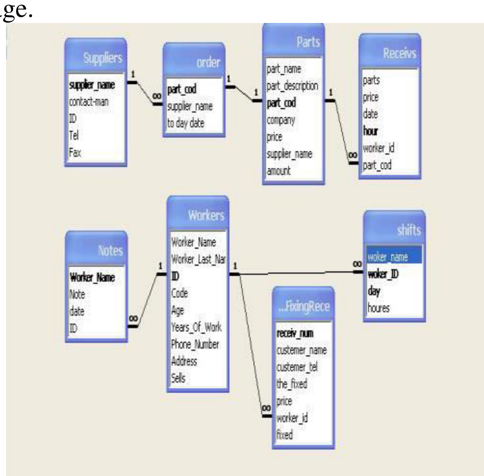



Fig.1. A Data Flow Diagram (DFD) sketched by a student as part of designing the information system.

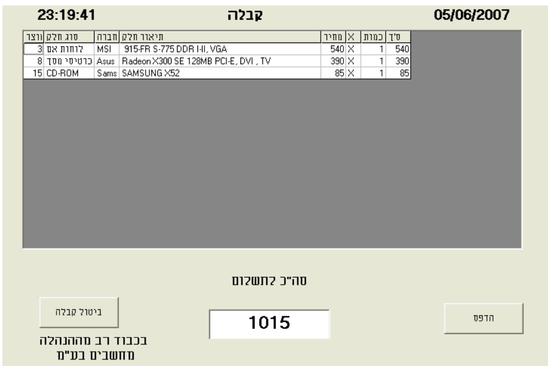
The diagram shown in Figure 1 relates to process P4 – receiving of a computer for repair in the store. This part of the system consists of two 'entities': E1 – the customer, and E4 – the store employee; the process is linked to two data files the student called DB1 – parts and DB7 – "FixingReceivs". The thin arrows in Figure 1 illustrate links that exist in the earlier system; the thick arrows indicate new functions the designer needs to add to it. It can be seen that the student sketched the chart in Figure 2 using a word processor; moreover, the student captured the entire screen using the PrtScr key and pasted it as a picture into his portfolio. In reality, today's designers use various sophisticated design tools called Computer Aided Software Engineering (CASE) and known as OOAD (Object-Oriented Analysis and Design) tools . This is another indication of the phenomenon that schools perceive that the essence of the projects was teaching programming, while the aims of fostering students' design skills and using advanced technological tools for design, gained less attention. Later in the paper, we will discuss the educational consequences of these findings.

### *How Students Programmed Their Information Systems?*

The major part of the students' work on their projects was programming the information system, which consisted of two stages: creating the databases for the system using Access (Fig. 2a), and programming all the logical functions, computations, users' interface screens and reports using Visual Basic (Fig. 2b). As previously mentioned, when the students started their projects in 12th grade, it was after they had studied programming in C, Visual Basic and using Access in 10th and 11th grades. However, the final project was the first time the students were required to integrate different programming tools and working environments in developing a system, rather than focusing each time on a specific method or programming language.




2a. Typical database structure created by Access




2b. A typical form created by Visual Basic

Fig.2. Examples of students' products in the implementation stage.

Despite their previous experience, the students often ignored or violated many of the good programming rules or guidelines previously described in the literature review, such as handling with variables, functions or procedures, and verifying the validity of data entered. One example is presented below:

- A student developed a data system in which the user is required to enter his ID number (9 digits) in order to log in. This student did not include a verification clause in the program as to whether the user enters only digits between 0-9, and if the entire number is a valid ID number, (there is procedure for this type of verification).
- After the teacher advised the student to include such a test in his system, he wrote a very cumbersome sequence and copied it into all of the forms that dealt with ID numbers, instead of building an efficient procedure for this purpose.

In the above example, the students did not follow two important programming rules: first, to check the data a user inputs before proceeding with any action; and second, the need to use modular programming for repetitive logical computational operations. Other frequent errors were the use of local versus global variables and variable type. In the interviews, students had the following comments:

*"I am used to working this way."*
*"I didn't feel it was important."*
*"I didn't know it was required that I use procedures."*
*"The teacher didn't ask me to do that."*

It is worth noting that although the Visual Basic working environment includes tools for debugging the kinds of errors shown above, the students often ignored these error messages in their first programming steps and opted to resolve them by using the trial and error method. In fact, reading English messages is not easy for many Israeli students; for Israeli Arab-speaking students in particular, as in the current case; English is their third language after Arabic and Hebrew.

During this 4-6 week period, the students progressed very slowly, and many felt very frustrated. Below are some examples of comments by students during this period:

*"The computer went completely crazy."*
*"I don't know what the problem is."*
*"The program worked in theory but did not run properly on the computer."*

### *Students' progress in programming*

So far, we have seen that when the students started programming their system, the vast majority encountered many problems and proved to be dependent on their teachers. However, after the initial 'shock' many students experienced when starting to program their system, things improved gradually, as demonstrated below.

- Upon the teachers' strong recommendation, and with their help, the students gradually moved to modular programming methods rather than repeatedly using specific sections of code in different parts of their program.
- The students improved their handling of Visual Basic error messages. For example, some students prepared a list of error messages they were receiving and translated them into Arabic or Hebrew.
- The students recognized that many of their mistakes were about defining and using variables, transferring parameters to procedures, etc. Consequently, many students included in their portfolio a table of all the variables they were using, including the name, type and function of each.
- Some students improved or re-wrote parts of programs they had prepared earlier to make them more efficient or elegant.

In discussions with students who had nearly completed their projects, the students had the following reflection on their work:

"*I now really understand the error."*
*"I made the same error again because I was in a hurry."*
*"I am writing down every error massage… I am taking notes… the next time it appears I will solve it faster…"*

### *Examiner's Viewpoint*

The examiner in one of the classes was experienced in computer engineering and information system design, and in teaching these subjects at the engineering level. The researcher attended the examinations in a class of 20 students for about four hours, and talked informally with the examiner and the students. The examiner asked the students to demonstrate the system they had developed, explain specific parts of the programs they wrote, and describe the entire system development process. Sometimes he asked the student to make some changes or improvements in the system and gave them 15-30 minutes to do so. In a discussion with the examiner, he had the following comments:

*"I am looking to see if a student understands the structure of the information system he or she constructed and the process of storing and updating data in the system."*
*"High school students are not professional programmers… it is enough that they acquire basic programming skills. It is more important to see if the students understand the process of information system development, are able to explain what they are doing and be aware of the advantages and limits of the system they constructed."*
*"One cannot compare what the students learn from working on their projects to that learned in a conventional course."*
*"Every year I am amazed about the knowledge level many students reveal. Their motivation is very high."*

### *Final Scores*

The students' final scores on their projects (on a scale of 0-100) are marked in the formal high school matriculation certificate each student receives from the Ministry of Education. These scores comprise an average of scores granted to each student from the teacher and the external examiner. In the current case, while the external examiners in the two schools based their marks mainly on the oral exam and the booklet each student prepared on his/her project, the teachers also took into account the student's efforts and progress throughout the project work. Of the 20 students in one school, eight were granted a score ranging from 90-100 (excellent), three from 80-89 (very good), five from 70-79 (good), and four from 60-69 (poor). The distribution of scores in these examinations in the other school was quite similar, and both schools present a common picture of students' achievements in the final projects on information system design in Israeli high schools. These findings indicate that the graduating project on information system design is a demanding task, and not all the students excel in it. Yet, students' reflection on the projects is quite positive, as described below.

*Outcomes of Attitude Questionnaire Administrated in the Class*

In addition to collecting qualitative data in the classes through observations and interviews, we administrated an attitude close-ended questionnaire in the class. The students filled in this questionnaire twice: first, about three weeks after they had started working on the projects, and then in the middle of the school year when the participants were at the peak of their project preparation process (upon completing their projects, the students answered another open-ended questionnaire; see the following section). The open-ended questionnaire was designed to probe students' viewpoints on issues such as the extent of independent learning in class, collaboration among the students, and the integration of theory and practice into the projects. The full text of the questionnaire is presented in Appendix 1. The students marked their answers to 23 items on a four-level scale: 1. Very low; 2. Low; 3. High; and 4. Very high. To obviate the problem of acquiescence bias, namely respondents' tendency to agree with all of the questions, 13 items expressed a positive opinion and the remaining 10 items a negative view. A preliminary version of the questionnaire was given to two pilot classes of computer science students (n=38) who prepared similar final projects in a different schools not participating in the current study. The researcher also talked with these students about the questionnaire content in order to identify problematic or unclear questions. An amended version of the questionnaire was administered in an additional pilot class (n=25) before formulating the final version that was given to the students in the current study. Statistical analysis (t-test) showed that there were no significant differences in the results (mean scores) either between the two schools participating in the study or between the results in the first and second runs in each school. Therefore, all of the results (class 1, 2 and the dates 1, 2) were combined into one file (n=80). We would like to provide a general picture of the findings and understand the 'spirit' of students' views about the different aspects of the project work, and not just look separately at each question. Therefore, we present the mean scores obtained in all 23 items in ascending order, in Appendix 1 and Figure 3. In Figure 3, the bars relating to 'negative' items in the questionnaire are filled in, while those relating to 'positive' items are empty. It may be seen that the students gave low scores (between 1 = Very low and 2 = Low) to the majority of 'negative' items, such as "I had no right to decide in my project work" (item 23) and "The project did not relate to the theories we had learned" (item 15). On the other hand, the students gave high scores (between 3 = High and 4 = Very high) to the 'positive' items, such as "The students helped each other out in the project work" (item 6) and "The project helped me understand the theory" (item 2). There were some exceptional cases, for example, 'positive' item 14 "I did not depend on the teacher in my project" was graded an average score of 2.0417 (Low), or 'negative' item 18 "The project caused competition and tension among the students," which received an average score of 3.0694 (High). However, Figure 3 illustrates that the students consistently expressed positive views about the project work, especially in the following aspects: learning computer sciences, independent learning and collaboration in the classroom.

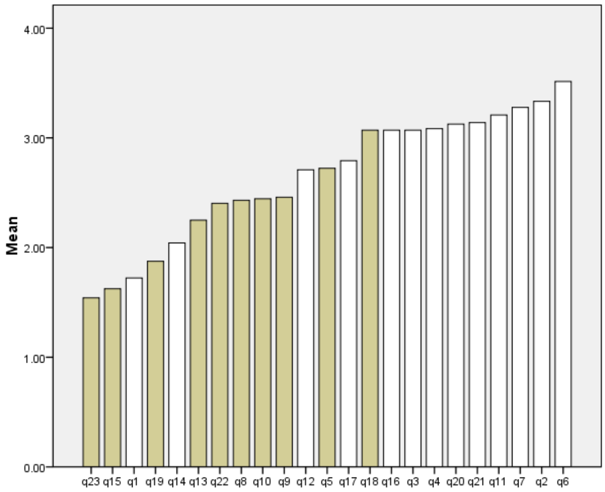



Fig.3. Students' answers to the attitude questionnaire twice during the project work

(1 = Very low; 2 = Low; 3 = High; 4 = Very high; Filled bars indicate a negative expression)

*Students' Views about Using the Internet in Working on Their Projects*

Since the participants in the current study were high-achievers who majored in computer sciences and had almost unlimited access to the Internet either in school or at home, we were interested in the question as to what extent and how the students used effectively the Internet in working on their project. Let us now see how the 40 students marked this question in the open-ended questionnaire they filled in at the end of the school year:

- Nine students wrote that they had used the Internet mainly for learning Visual Basic or for finding solutions to programming problems.
- Seven students reported that they had used the Internet for communicating with their friends, exchanging codes or other software tools.
- Twenty-four students stated that they had not used the Internet at all for their projects.

In the closed-ended questionnaire discussed in the previous section, the mean score for item 20 "The Internet helped me enrich my project" was 3.12 on a scale of 1 (Very low) to 4 (Very high). How can we explain the fact that the students reported only moderate use of Internet for their projects? Further discussions with the students revealed that many of them use the Internet extensively daily. However, they considered using Internet for school

International Journal of Innovation and Research in Educational Sciences
Volume 2, Issue 2, ISSN (Online): 2349–5219

purposes as being less legitimate, since they regarded the project as a kind of personal task or exam they had to accomplish on their own.

## VI. Discussion

This study presents an example of creating a student-centered learning environment in learning information system design in high school. The current case deferred from the conventional approach of teaching information system design in two main points. Firstly, the students learned the new subject by working on a graduating project throughout the entire academic year as an alternative to learning a conventional teacher-directed course. Secondly, the assessment was not based on a final pen-and pencil exam, as is common in Israeli schools, but on the teachers' and an external examiners' evaluations, taking into account students' efforts and progress while developing their projects, the final products they produced, the portfolio they prepared, the knowledge and skills in information system design they demonstrated, and their ability to reflect on their project work. Perhaps this change in the evaluation approach indicates more than anything the schools' shift from what [27] called the 'teaching paradigm' to the `learning paradigm' in information system education. Yet, the research also noted some difficulties and issues that educators should consider in introducing the constructivist-guided learning into teaching computer science in high school, as discussed below.

Projects in information system design put into practice the notion of contextual learning, which has to do with educators' beliefs that meaningful learning takes place when students learn things in school that relate to real-life contexts, issues that interest them or affect their lives. In the current study, we have seen that giving students the opportunity to prepare a project on developing an information system in domains that are personally meaningful to them or to their family could very positively affect their motivation to accomplish the task. In these projects, students have the opportunity to deal with real needs and construct their knowledge by creating products that could be useful to others. This is the central idea behind the constructivist learning theory [17], [11], which suggests that learning can occur most effectively when people are actively engaged in making things that are meaningful to them and that they can share with others.

The current study revealed, however, that the idea of authentic projects must be carefully considered. Students, especially in high school, have limited knowledge and experience that are required to fully develop a real-life information system. Since students tend to choose projects that are too complicated for them, teachers should guide each student in selecting a project appropriate for his/her abilities. Learning information system design through projects is closely related to the notion of knowledge construction, which is in the heart of the constructivist learning theory. According to this view, learners construct their own knowledge by incorporating new experiences into already an existing knowledge and conceptual framework [26]. Exploration, feedback and reflection are important to this process. In the current study, we observed this process through the way students worked on the actual construction of their system, namely programming. Although many students initially faced difficulties in building their system, the majority made marked progress in completing their project. Let us examine more closely, what happened in the classes. When the participants started working on their graduating projects, they had already studied two years of programming in the environments of C, Visual Basic and Access. Despite this experience, the participants often made errors common to novice programmers. The explanation for this was that in 10$^{th}$ and 11$^{th}$ grades, the students had learned programming as a series of separate issues, and completed most of their practical assignments by solving specific exercises either in the class or at home. Moreover, as [15] pointed out, in conventional computer science courses, the teachers mostly ask the students to write software programs of relatively limited scope and do not engage them in extensive and complex tasks. In the graduating project, the students had to deal with an open and extensive task that required them to integrate different domains of knowledge, learn new subjects independently, and be responsible for their project. It took time for the students to grasp this change, but they gradually changed their working habits and improved their programming skills. For example, the students significantly improved their abilities to handle variables, procedures and functions, or to debug their programs. Two factors were crucial to this process: one – the situation whereby the student already had basic knowledge and experience in programming; two – the fact that programming is an interactive process that enables students to learn from their mistakes and get immediate feedback about their work.

A different picture was found concerning the students' work on the conceptual design of their system. As we have seen in the Findings section, many students only partially completed their work on the system's conceptual design, such as learning about the customer's needs, preparing a detailed project charter for their design, or drawing Data Flow Diagrams (DFD) for the system they were developing. We can attribute this to three factors. Firstly, the project was the first time that the students dealt with these subjects and they were inexperienced in learning new subject matter on their own. Actually, the students commented that it is essential that the teacher give basic explanations and show examples to the entire class to save them time and effort in working on their individual projects. Secondly, the students had misconceptions about the process of system development. This occurred because while studying programming in 10$^{th}$ and 11$^{th}$ grades, they had received artificial or imaginary data from the teacher to practice programming, while the question as to the source of these data or their validity was considered marginal. Later, it was difficult to eliminate this incorrect conception from the students' minds and have them seriously research the problem they were tackling. Thirdly, the students performed all the design processes manually (for example, making design sketches and graphs using the

IJIRES Source of Knowledge
International Journal of Innovation and Research in Educational Sciences
Volume 2, Issue 2, ISSN (Online): 2349–5219

word processor drawing tool), and were primarily dependent on the teacher's help and feedback. Therefore, this part of the project departs from the spirit of learning computer science in an advanced interactive technological environment. As noted earlier by [2], [3], [23], computers could serve as 'cognitive tools' since a computerized learning environment can help learners in accessing information, representing ideas, developing ideas, and generating products. Moreover, computer and information technologies are also regarded as 'metacognitive tools' since they can guide students in making decisions regarding their learning goals or in receiving and using feedback from their tutors, peers or technological means. In the schools observed in this research, the technological environment contributed to learning through the students' work on programming but not to their work on the general design of their systems.

## VII. Concluding Remarks

Information system design is likely to provide a good platform for creating a constructivist technology-based learning environment in school. However, the design of a real-life system is often a complicated task for high school students because it requires the integration of conceptual knowledge, mainly in the phase of defining a system's objectives and general structure, and procedural knowledge, for example, in the detailed design, implementation and testing phase.

The common situation in schools is that students learn and practice mainly procedural knowledge, whereas accumulating conceptual knowledge is a long-term process. Indeed, the current research highlighted that the conceptual design phase might appear to be one of the most challenging parts for students inexperienced in carrying out a design project, because this task has to do mainly with exploring peoples' needs, desires and expectations. As students proceed to detailed design phases, including implementing and programming, the task becomes more concrete to them since it is essentially about procedures and instrumentation.

To encourage learning and knowledge construction, it is important to merge the teaching of design concepts with the teaching of programming by engaging students in open-ended design tasks at the very early stages of learning computer sciences. It is recommended that educators give students small, open-ended design tasks and gradually increase the scope and complexity of these assignments. After all, the student's designing skills is to being developed after cycles of experimentation, interaction with instrumentation, collaboration with peers and reflection.

**Appendix 1: Students' Answers to the Attitude Questionnaire Administrated in the Class at the Beginning and Middle of the Project Work (N=40)**

(Items arranged in ascending order of mean score)

| No | Item | Mean | Std. Dev. |
|---|---|---|---|
| 23 | I had no right to decide in my project. * | 1.5417 | .69073 |
| 15 | The project did not relate to the theories we had learned. | 1.6250 | .63772 |
| 1 | I got help from a professional out of school. | 1.7222 | .84290 |
| 19 | I was able to prepare my project without learning the theory. | 1.8750 | .90285 |
| 14 | I did not depend on the teacher in my project. | 2.0417 | .84649 |
| 13 | Other students could not help me in designing the system for my project. | 2.2500 | .80053 |
| 22 | There was no need for much theoretical knowledge in my project. | 2.4028 | .68505 |
| 8 | The theory we learned is at a higher level than my project. | 2.4306 | .64625 |
| 10 | I made progress at each stage of the project only after receiving the teacher's approval. | 2.4444 | .90209 |
| 9 | I helped out other people even though it delayed progress in my project. | 2.4583 | .76798 |
| 12 | I made use of examples of projects from previous years. | 2.7083 | .95589 |
| 5 | Without the teacher's help, it was difficult for me to make progress. | 2.7222 | .90728 |
| 17 | I made use of books for my project. | 2.7917 | .85477 |
| 18 | The project caused competition and tension among the students. | 3.0694 | .69862 |
| 16 | I got ideas from friends while working on the project. | 3.0694 | .67816 |
| 3 | I helped my friends during the project work. | 3.0694 | .69862 |
| 4 | The project integrated the theoretical knowledge we had learned in various computer science courses. | 3.0833 | .74588 |
| 20 | The Internet helped me enrich my project. | 3.1250 | .78610 |
| 21 | I willingly helped other students. | 3.1389 | .61221 |
| 11 | The project work enabled me to practice the theory I had learned. | 3.2083 | .78610 |

| | | | |
|---|---|---|---|
| 7 | I took control and made decisions on my own in my project. | 3.2778 | .63295 |
| 2 | The project helped me understand the theory. | 3.3333 | .69201 |
| 6 | The students helped each other out in the project work. | 3.5139 | .55647 |

(Scale: 1 = Very low; 2 = Low; 3 = High; 4 = Very high)
* Marked items indicate a negative expression

## REFERENCES

[1] Avison, D. E., & Fitzgerald, G. (2002). *Information systems* development: Methodologies, techniques and tools. London: McGraw-Hill.
[2] Azevedo, R. (2005). Computer environments as metacognitive tools for enhancing learning. Educational Psychologist, 40(4), 193-197.
[3] Azevedo, R., & Hadwin, A. F. (2005). Scaffolding self-regulated learning and metacognition – implications for the design of computer-based scaffolds. *Instructional Science, 33,* 367-379.
[4] Barak, M. (2005). From order to disorder: The role of computer-based electronics projects on fostering of higher-order cognitive skills. *Computers and Education, 45*(2), 231-243.
[5] Barak, M. & Shachar, A. (2008). Project in technology and fostering learning skills: The potential and its realization. *Journal of Science Education and Technology, 17*(3), 285-296.
[6] Barron, B. J. S., Schwartz, D. L., Vye, N. J., Moore, A., Petrosino, A., Zech, L., & Bransford, J. D. (1998). Doing with understanding: Lessons from research on problem- and project-based learning. *Journal of the Learning Sciences, 7*(3), 271-311.
[7] Bennett, F. L. (2003). *The management of construction: A project life cycle approach*. Amsterdam: Butterworth/ Heinemann.
[8] Brown, J. S., Collins, A. & Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher,18*(1),32-42.
[9] Bruner, J. S., and Bruner, J. S. (1977). *The process of education*. London: Harvard University Press.
[10] Cherubini, M., Venolia, G., DeLine, R. and Ko, A. (2007). Let's Go to the Whiteboard: How and Why Software Developers Use Drawing. *Chi Conference, 1,* 557-566.
[11] Connelly, T. M. & Begg, C. E (2006). A constructivist-based approach to teaching database analysis and design. *Journal of Information Systems Education, 17*(1), 43-53.
[12] Dewey, J. (1963). *How we think: A restatement of the relation of reflective thinking to the educative process*. Heath.
[13] Dolmans, D. H. J. M., De, G. W., Wolfhagen, I. H. A. P., and Van, D. V. C. P. M. (2005). Problem-based learning: future challenges for educational practice and research. *Medical Education, 39*(7), 732-741.
[14] Fincher, S., & Petre, M. (1998). Project-based learning practices in computer science education. *Proceedings - Frontiers in Education Conference, 3*, 1185.
[15] Gal-Ezer, J. & Zeldes, A. (2000). Teaching software designing skills. *Computer Science Education, 10*(1), 25–38.
[16] Hamalainen, W. (2004). Problem-based learning of theoretical computer science. *Proceedings - Frontiers in Education Conference, 3.*
[17] Harel, I., Papert, S., & Massachusetts Institute of Technology. (1991). *Constructionism: Research reports and essays, 1985-1990*. Norwood, N.J: Ablex Pub. Corp.
[18] Hazzan, O., Dubinsky, Y., Eidelman, L., Sakhnini, V., & Teif, M. (2006). Qualitative research in computer science education. *Acm Sigcse Bulletin, 38*, 1.
[19] Jonassen, D., & Reeves, T. (1996). Learning with technology: Using computers as cognitive tools. In D. Jonassen (Ed.), *Handbook of research for educational communications and technology* (pp. 694–719). New York:Macmillan
[20] Karweit, D. (1993). *Contextual learning: a review and synthesis*. Baltimore, MD: Center for the Social Organization of Schools, Johns Hopkins University.
[21] Kernighan, B. W., & Plauger, P. J. (1978). *The elements of programming style*. New York: McGraw-Hill.
[22] Kilpatrick, J., Swafford, J., Findell, B., & National Research Council (U.S.). (2001). *Adding it up: Helping children learn mathematics*. Washington, DC: National Academy Press.
[23] Lajoie, S. P. (2000). *Computers as cognitive tools, volume two: No more walls: theory change, paradigm shifts, and their influence on the use of computers for instructional purposes.* Mahwah, N.J: L. Erlbaum.
[24] Myers, M. D., & Avison, D. E. (2002). *Qualitative research in information systems: A reader*. London: SAGE.
[25] Nuutila, E., Törmä, T. and Malmi, L. (2005). PBL and computer programming - the seven steps method with adaptations. *Computer Science Education, 15*(2), 123-142.
[26] Piaget, J. (1950). *The psychology of intelligence.* London: Routledge & Paul.
[27] Saulnier, B. M., Landry, J. P., Longenecker, H. E., & Wagner, T. A. (2008). From Teaching to Learning: Learner-Centered Teaching and Assessment in Information Systems Education. *Journal of Information Systems Education, 19*(2), 169-174.
[28] Savery, J. R. (2006). Overview of problem-based learning: definitions and distinctions. Interdisciplinary. *The Interdisciplinary Journal of Problem-Based Learning, 1*(1), 9-20.
[29] Scherz, Z., & Polak, S. (1999). An organizer for project-based learning and instruction in computer science. *Sigcse Bulletin, 31*, 88-90.
[30] Schunn, C.D., & Silk, E. M. (2011). Learning theories for engineering and technology education. In M. Barak & M. Hacker (Eds.). *Fostering Human Development through Engineering and Technology Education* (pp. 55-74). Netherlands: Sense Publishers.
[31] Thomas, H. W., Mergendoller, J. R. & Michaleson, A. (1999). *Project-Based Learning: A Handbook for Middle and High School Teachers*. Novato, CA: The Buck Institute for Education.
[32] Tversky, B. (1999). What does drawing reveal about thinking. In J. S. Gero, & B. Tversky (Eds.). *Visual and Spatial Reasoning in Design* (pp. 93-101). Sydney: Key Centre of Design Computing and Cognition.
[33] Yourdon, E. (2007). *Structured Analysis*, Retrieved January 2009, from http://www.yourdon.com/jesa/jesa.php.
[34] Wix, J., & Price, C. (1999). *Describe and operate a personal computer system (Version 3).* Auckland [N.Z.: Software Educational Resources.

## AUTHOR'S PROFILE

**Mr. Nayif Awad**
is a Ph.D. student in science and technology education at Ben Gurion University of the Negev. His research deals with promoting an integrated learning of STEM subjects at rich computer – based learning environments.
He received his B.Sc. degree in computer engineering from the Technion - Israel Institute of Technology, and his M.A. degree in science and technology from Ben Gurion University with acknowledgement of excellence in his final paper. Currently he is a lecturer of mathematics and computer science at the Academic College for qualifying teachers in Sakhnin and a coordinator of ICT specialization. He teaches a wide range of courses such as: digital systems, computer architecture, science programs in junior high schools, and computer image processing (with STEM interdisciplinary approach).
Mr Awad has an experience in teaching software engineering at the high school level for more than 10 years. He has led many projects aimed at reducing the digital gaps between students in schools and in college, and taught many professional development courses for teachers in subjects of science and computers.

**Prof. Moshe Barak**
is the head of the department of science and technology education at Ben Gurion University of the Negev, Israel.
He received his PhD, M.Sc. and B.Sc. degrees from the Technion – Israel Institute of Technology. His professional background research areas are in electrical engineering and include subjects such as technology, engineering and science education, project-based learning, problem solving and creativity.
Prof. Barak has over 30 years of experience in curriculum development and evaluation in areas such as control systems, robotics, image processing and electronics, including teachers' pre-service and in-service training. He has published over 45 papers in professional journals.