An Experience Report on the Use of Experience Maps and Sketches in a Database Course Project

Alexandra Martínez
Escuela de Ciencias de la Computación e Informática
Universidad de Costa Rica
San José, Costa Rica
alexandra.martinez@ecci.ucr.ac.cr

***Abstract*— This paper describes our experience introducing two user-centered design techniques in the context of an undergraduate database course project: experience maps and sketches. Experience maps are used to represent user requirements, while sketches are used to design the application's flow and interfaces. The goal of incorporating these techniques in the course was primarily to improve the quality of student projects. In previous semesters, the teacher noted problems related to requirements elicitation and interface design, mainly due to the students' lack of software engineering background. Experience maps and sketches were used for three semesters, and were evaluated through a student survey and a teacher's assessment of benefits and limitations. Results show that at least 85% of the students valued the creation of experience maps positively, and 100% valued the creation of sketches positively. Students commented that experience maps helped them the most to understand user's needs, each interface's purpose in the context of the entire solution, as well as the user's interaction with the system. The main limitation found by the teacher was the difficulty students had grasping and applying experience maps concepts. The main benefit was that in the end, project prototypes had more consistent, clear and usable interfaces.**

Keywords—experience maps, sketches, requirements elicitation, interface design, database course, project

I. Introduction

Database courses normally involve a large project where students put in practice the concepts learned. In our case, the course project is developed for a real end-user, and therefore, it is expected to have a friendly graphical interface by which the user can query and operate the database. However, due to the lack of software engineering background by the time students take our course, project products often miss relevant information (in the case of requirement specifications) or lack usability (in the case of the final prototype).

The main challenge we face is that students have to take the database course before the software engineering (SE) course, whereby they lack SE knowledge and skills when enrolling in our course. This has several implications. First, little time can be devoted in class to explain SE topics needed for the project such as software requirements elicitation and specification, and interface designing and prototyping. Second, students need some guidance on these SE topics in order to produce high quality prototypes. Avoiding the issue (leaving it up to the students to learn by themselves) does not help, since they can be either overwhelmed by the amount of techniques and approaches found on the web, not knowing which to choose, or mistaken by thinking that those processes are always performed in an empirical way. Third, we cannot and should not convert the database course in a software engineering course, meaning that our main focus should still be database design and implementation, while keeping the course project at a reasonable quality level. Removing the course project is not an option since both teachers and students consider it an invaluable motivational aid in the course.

In the past, most database teachers have attempted to address the problem by devoting a little time in class (typically less than an hour) to briefly explain to the students how to do an user interview, extract relevant information from it, and write down the requirements specification (emphasizing more on data requirements than on functional or non-functional requirements). To our knowledge, no guideline has been offered by the teachers on how to design application interfaces based on user experience (UX) principles. There is a lab practice where students learn to create a desktop application (with GUI components) that connects to a database, but its focus is on the programming aspects rather than the UX.

Such previous approaches to address the problem have come short of solving it satisfactorily. The author, who is one of the database teachers, has noticed that students often have difficulties in capturing user's needs and expressing them as system requirements. This is evidenced in that some teams: (1) usually leave the non-functional requirements section blank, (2) confuse functional requirements with non-functional ones, (3) present an incomplete requirements list, possibly derived from poorly-conducted user interviews, (4) focus heavily on data requirements (as the basis for database design), at the expense of functional requirements, (5) develop a system prototype that exhibited many usability problems, rendering it of little value for the client. In summary, while students were achieving the project goals from the database perspective, they failed to meet the client expectations from the usability and business point of view, therefore missing the opportunity to exploit the full potential of the project.

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

II. BACKGROUND

In our search for new ways to overcome the aforementioned problems, we came across *experience maps* (EMs) and *sketches*. Experience maps, also known as *customer journey maps*, are a visual representation of the users' experience while interacting with a system [1]. They use storytelling and visuals to illustrate the experience a customer goes through —across all touchpoints and channels— in order to accomplish a goal tied to a specific business or product [2]. The goal of user experience mapping is to understand and address customer needs and pain points [2,3].

One of the main draws of experience maps is the ability to bring a team together during its elaboration, promoting communication and establishing common objectives. The intention of producing an experience map is to understand the interactions between a user and a product or service from the standpoint of the former rather than the latter. Inviting the students to put themselves in the position of their prospective users helps them become more empathic towards the particular needs and desires of their end users.

On the other hand, sketching is an effective tool for creating a bridge between the design and implementation of a system, as well as among the individual ideas from each team member. Handmade drawings are inexpensive, quick to make, and easy to validate among collaborators. The most relevant idea for emphasizing this design methodology is the necessity to share ideas visually, and to avoid teams skipping design goals in favor of starting implementation right away, potentially ignoring any discoveries that were made during the experience map production stage.

III. RELATED WORK

Our literature review did not find any previous work where experience maps were used in an academic context for eliciting requirements. The closest study is the work by Ferreira et al. [4], where empathy maps are used to generate *personas*. An empathy map is a method that helps design business models according to the client's perspective, using guide questions [4]. The authors proposed a technique called PATHY, which integrates guide questions and the structure of the empathy maps with the idea of describing users through *personas*. PATHY borrows some steps from Design Thinking to understand who the user is, find her needs and problems, and generate innovative solutions. This technique helps identify features and characteristics for an application based on users' problems and needs. Our work differs from theirs in that we use experience maps –as opposed to empathy maps or personas– as a way to capture and represent user requirements. Another difference is that the students who participated in their study had already taken a Human Computer Interaction (HCI) course and an SE course, whereas our students had not. We also believe that experience maps are a step beyond *personas*, since they capture the experience of a user when interacting with the system. To design a good experience, the development team should understand the user's needs and problems.

The works of [5, 6, 7, 8] studied the relationship between user experience or HCI and requirements engineering in an industrial context.

Anithe PC et al. [5] proposed an integrated framework for requirements engineering (RE) and user experience design (UXD) in the product life cycle management process. The framework was not validated, but authors discussed some myths about RE and UXD that have become barriers for the integration of these disciplines. The authors state that UXD, unlike RE, has not yet found a formal space in traditional development processes, and point out that one of the reasons is the gap between academia theory and industry practices, which makes UXD still an alien concept to many organizations and development teams [5]. They also mention that user experience is not just about the interfaces of the product but also about defining the product itself and the total experience a user has with it. UXD professionals employ 'user-centered design' techniques to create products that are effective, usable, valuable, relevant and delightful [5]. Some of the UXD artifacts mentioned in this work are personas, scenarios, interaction concepts, wireframes and UI style guides [5]. Our experience maps are somewhat similar to scenarios, and our sketches and flows are a form of interaction concept, which can then be refined into wireframes. We believe that by teaching students the use of experience maps and sketches, we are contributing to integrate RE and UXD at an academic level, which can eventually permeate industrial practice.

Similarly, Heiskari et al. [6] conducted a case study in two software companies with the aim to understand the role of usability specialists in the requirements engineering process. They said that usability and RE both provide means for discovering, analyzing and fulfilling users' needs [6], which validates our premise. Their findings evidence that usability specialists are not integrated in RE activities, mainly due to the fact that usability is still seen as user interface design instead of a more comprehensive characteristic of a software product [6]. Their results also show that user interface design is a frequent task performed by usability specialists, while user studies is an infrequent task [6]. User studies gather data on users and their needs (typically though interviews or observation), with the aim of translating it into user requirements [6]. An interesting finding they report regarding UI design is that in order to design the user interface for an entire product, the big picture has to be understood early in the process (usability specialists have to be involved from the beginning –in elicitation phase-, since late participation as validators of ready-made designs had little effect on the design of the product) [6]. With our work, we are contributing to bridge RE and UXD at the academic level, by teaching students the importance of incorporating usability (experience maps) early in the design, and how this can shape the design of the entire system (forcing a traceability from EMs to sketches).

The work by Callele et al. [7] also explores the combination of UXD and RE, in the form of *experience requirements*, which are either descriptions of user experiences that must be met (functional experiences) or satisfaction goals (non-functional experiences) for products or services. The authors use a stimulus-perception-response model to guide the design of the user experience. Unlike their work, we represent the user experience with EMs rather than experience requirements, and no explicit model is followed.

Joshi et al. [8] proposed two metrics to measure the impact of integrating HCI activities in SE processes: (1) the usability goals achievement metric, a product metric that measures the extent to which the design of a product achieves its user-experience goals, and (2) index of integration, a process metric that measures the extent of integration of the HCI activities in the SE process. These metrics were evaluated both in an academic setting and an industry setting. Authors reported that the metrics were found to be useful, easy to use, and helpful in making the process more systematic.

Ardito et al. [9] analyze current software development practices in Italy, by means of an experimental study conducted with software companies. Their study confirmed that still many companies either neglect usability and user experience, or do not properly consider them [9].

Several tools exist to support sketching [10, 11, 12, 13]. In particular, Buchmann [10] presents an extension to Valkyrie, a UML-based modeling tool that allows free-hand diagram sketching. On the other hand, Wuest et al. [11] developed FlexiSketch, a mobile tool for model-based sketching of free-form diagrams, which allows defining and reusing diagramming notations on the fly. Later they extended their tool to support collaboration with multiple tablets and an electronic whiteboard, such that several users can work simultaneously on the same model sketch; they called this version FlexiSketch Team [12]. Motta et al. [13] present a sketching tool called Calico, which provides lightweight analysis and feedback about the developer's design. In our work, we decided not to use any sketching tool but rather use paper-based hand-made sketches, for ease and speed.

IV. Course Context

The Databases I course is a mandatory undergraduate course situated in the 3rd year of the Bachelor of Science's program, in the Department of Computer and Information Science at the University of Costa Rica. This 4-credit-hour course provides students with the necessary concepts to design, implement, and manipulate relational databases. The course is offered every semester, with an average enrollment of 20 students per section. The class meets twice a week for one hour and fifty minutes during 16 weeks, for a total of 64 hours of class in a semester.

A. Course Objectives

The main goal of the course is that students develop skills to design, implement, query and operate relational databases, through strategies that integrate theory and practice.

The specific objectives addressed through the project are:

1. To design relational databases which satisfy the data and operational requirements of a system.
2. To implement, query and operate relational databases, by using SQL, in order to build databases that operate correctly.
3. To build a software application that interacts with the implemented database.

B. Course Evaluation

The distribution of the final course grade was as follows: reading assignments (8%), quizzes (8%), lab reports (8%), two tests (20% each), and project (36%).

Before the introduction of EMs, there were two projects in the course: a large relational database project, weighting 30% of the course grade, and a smaller object-relational database project, weighting 6% of the final course grade (with small variations between semesters). The second project was developed during the last 3 or 4 weeks of the semester, and by that time students had already finished their first project. After some discussions with other database faculty, we decided to remove the second project (in fact, we removed the object-relational databases topic completely from the course). Therefore, we had at our disposal a little more time in the course and in the relational database project, too. We opted to spend more time in class studying topics that students normally struggled with, offering extra practice and more hands-on labs. Regarding the project, introducing EMs and sketches were two of the improvements made.

C. Course Project

The project consists in creating a prototype of a small database system for an actual end-user (client), in teams of 3 to 4 students. Students look for the client themselves, within their family or circle of friends, since it needs to be someone of trust in order to handle expectations properly (given that this is the first time students will develop a system prototype for someone other than the teacher). This is usually the first experience students have developing a system for a client.

Students usually develop the project outside of class, with sporadic input from the teacher when teams request it. The project is hence seen as a complement to the course, where students apply and deepen what they have learned in class: database conceptual design using the Entity-Relation model, database logic design using the Relational model, and database implementation using SQL, and desktop application building using Microsoft's C# and SQL Server. Since this is a database course project (not an SE course), students are expected to learn some things by themselves and investigate aspects that are not necessarily studied in class.

Before the introduction of EMs in the course, the project was divided in two phases only (given that students had less time to complete this project): *Requirements and design* and *Prototype implementation*. During the first phase, students performed user interviews to elicit the system requirements, wrote these requirements in functional, non-functional and data requirements, and finally proceeded to design the database based on the requirements.

D. The Teacher

The instructor of the course is an Associate Professor in the Department of Computer and Information Science at the University of Costa Rica. She has worked there for nine years, teaching both undergraduate and graduate courses in databases and software testing. She has a Ph.D. in Computer Engineering, and over three years of industry experience as a software tester.

We also had a collaborator with practical knowledge in UXD. He had recently graduated from our University, and was working on a research project with the course teacher. He had attended several workshops and trainings on entrepreneurship and UXD.

E. *The Students*

The students are usually in their third year of the B.S. program, and most of them were taking the course for the first time. By the time they enroll in this course, they have taken two programming courses, and one data structures and algorithms course, among others. However, they have not yet taken the software engineering course due to prerequisites.

V. DESIGN

A. *Design Goal*

The goal of incorporating EMs and sketches in the course is primarily to improve the quality of student projects. Clearly, in the process of using these techniques in their projects, students will learn about interface development and user-centered design, but we do not pretend this to be an HCI course.

B. *Type of Experience Maps Used*

User experience maps come in a variety of layouts, components and complexity, but their common objective is to depict a flow that will emerge between the system and a user trying to accomplish a task or goal. In order to help students focus on designing the user experience rather than worrying about which specific type of map to use, the teacher provided an EM template, so that all teams worked on an standard layout. This template was created by our UX collaborator (based on existing EM layouts) to meet the needs of the course, in coordination with the course teacher. Students were asked to create one EM per combination of user type and task or goal.

Figure 1 shows the EM template provided to the students, which was given in the form of a shared Google's drawing document that students could copy and customize as needed. This template favors a chronological depiction of the user's flow or experience, and it is composed of the following:

- *Experience stage*: discrete steps in the user flow towards completing a task or goal.
- *Context*: whether the stage involves interacting with the system (internal) or it happens in a separate environment (external).
- *Development over time*: whether the stage is an iterative process to be repeated a number of times (reiterative) or if it is a sequential chain of one-way events (linear).
- *Touchpoints*: specific customer interactions that take place in particular channels, during an experience stage. In our case, there is only one channel for all internal steps: a desktop application (this is a course restriction). However, there may be different touchpoints within this channel, and we instructed students to be rather specific when naming the touchpoint, by using descriptive interface names, such as 'Product search' or 'New product form' (acknowledging the fact that interfaces had not yet been designed).
- *Actions*: a brief listing of actions expected to occur during a given stage. This particular section more closely resembles the functional requirements, because each action implies a functionality the system is expected to fulfill.

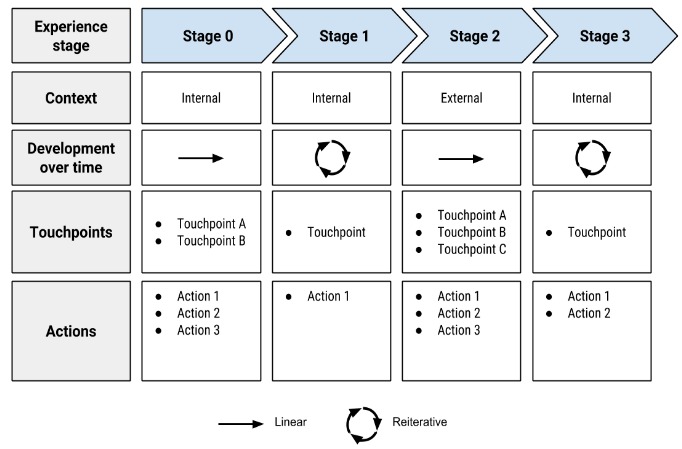

Fig. 1. EM template provided to the students.

C. *Type of Sketches Used*

In the same way as there are many types of EMs, there is also a wide variety of sketches. To prevent students from potentially spending too much time learning how to use a specific sketching/wireframe tool, the teacher suggested using paper-based sketches. The aim was for the teams to focus on the communicative effectiveness, usability considerations, and experience flow rather than on a high-fidelity wireframe or prototype design. A sketch flow was represented by arrows between sketches, and was meant to further detail the associated experience map. Both sketches and their flows served as an initial design of the interaction between the application and its users.

D. *Resources*

Given that it was not possible to devote much class time to explain the concepts of experience maps and sketches, the UX collaborator volunteered to give a talk (out of regular class time) about these topics. The talk had three parts. The first part was on user interviews: how to do an interview, common errors to avoid and good practices. The second part was on experience maps: the concept and aim of EMs, examples, and explanation of the template to be used in the course. Finally, the third part was on sketches. This last part also covered visual design principles and usability heuristics, both relevant aids in user experience sketching.

The visual design principles explained to students in the talk were loosely based on Gestalt principles of perception: contrast, alignment, repetition and proximity. For each principle or concept, a pictorial example of how it translates to a graphical user interface was given. Additionally, Nielsen's usability heuristics were also discussed. For each of the ten heuristics, an example of 'correct' use was presented, followed by an example of 'incorrect' use. To wrap up, a particular website was dissected using all of the heuristics, to show how these could be used to evaluate the usability of a system. We aimed for these principles and heuristics to guide students during the sketching process.

The first time the talk was given, it was recorded and the video was made available to students of subsequent semesters. The reason we included a part on user interviews was because we told students that experience maps were meant to encapsulate the information obtained from the user interview, but we felt students needed some guidance on how to make a good user interview.

E. Fitting Experience Maps and Sketches within the Project

The project was enlarged to encompass EMs and sketches. Consistent with the additional work required from students, the total weight of the project in the course grade increased by 6%. The major structural change in the project was division into three stages instead of two. The project phases were now as follows: (1) *Requirements and database design*, weighting 12%, (2) *Interface design*, weighting 8%, and (3) *Prototype implementation*, weighting 16%.

Project phases. Experience maps were added to the *Requirements and database design* phase, as a way to capture and represent the user requirements (in substitution of the previous textual representation of functional and non-functional requirements). Data requirements were kept since they are still the main source for database design. However, students were now required to verify consistency across experience maps, data requirements and database design.

On the other hand, sketches were added to the *Interface design* phase. Students were asked to verify consistency across experience maps and sketch flows.

Deliverables. At the end of the *Requirements and database design* phase, students had to turn in a "Requirements Specification and Database Design" document, and give an oral presentation about it. This document included: (*i*) an introduction encompassing the scope of the system and its main users, (*ii*) the requirements specification encompassing experience maps and data requirements, and (*iii*) the database design encompassing the conceptual and logic schemas. A template for the document and the EMs were provided.

Additionally, at the end of the *Interface design* phase, students had to turn in and explain the sketches and their flows, drawn on a large poster. Students had to verify consistency across experience maps and sketch flows.

Evaluation. A review session was planned with every student team at the end of the *Requirements and design* phase, where they had to present their experience maps and database design. In this session, both the teacher and the UX collaborator were present. Experience maps were evaluated in terms of completeness: Were important processes/flows missing? Were important steps in a process/flow missing? and correctness: Are the processes or experiences correctly depicted in the EM? Are the internal/external steps correctly identified? Are the touchpoints and actions correct and consistent? Clearly, consistency between EMs, data requirements and database design was evaluated as well. The feedback given to the students in this session had to be incorporated in the next phase of the project.

A second review session was planned at the end of the *Interface design* phase, where the teams had to explain their sketches and flows. In this session, again both the teacher and the UX collaborator were present. Sketches were evaluated based on consistency and adherence to visual design principles and usability heuristics. The consistency between sketch flows and experience maps was also evaluated. The feedback given to the students in this session had to be incorporated in the next phase of the project.

VI. IMPLEMENTATION

Experience maps and sketches were used during the fall semester of 2015 as well as during the spring and fall semesters of 2016. In 2015, 4 women and 15 men enrolled in the course. During the first semester of 2016, the class was composed of 2 women and 22 men. In the second semester of 2016, the class had 2 women and 18 men.

A. Sample Experience Maps Created by the Students

Figure 2 shows two examples of experience maps created by the students. The top image corresponds to the experience of an employee (user) when requesting vacations (task/goal), in the context of a company´s employee management system. The bottom image corresponds to the experience of an administrator (user) registering a payment to a customer's layaway (task/goal), in the context of a furniture store's sales and layaway system. In the first case (top EM), we see that students incorrectly classified the context of the last stage as 'external/internal' when it should be 'internal only'. Besides, it is not clear how the employee will get the vacation request response, since multiple touchpoints are specified: administrative employee, email, and database. ¿Does this mean the employee will receive the response through all of them, or any of them? In the latter case (bottom EM) we see that students missed some actions, for example: in the 'Data display' stage, there should be an action before 'Display data related to the selected layaway' where the user actually selects a layaway (the previous stages just searches but not selects). Also, in the 'Enter a payment' stage, there should be at least two actions before the 'Subtract' action: one where the user selects the 'Make payment' option, and another where the user enters the amount to pay.

It is evident from both EM examples that not all students conceptualize a touchpoint in the same way. Particularly, they tend to write generic touchpoints such as 'computer' or 'view in database', rather than specific ones like an interface or view name (as was instructed and expected). Two possible reasons for this are that they either confused touchpoints with channels or really had no idea of what interfaces they would come up with, thus, were not able to provide a descriptive name of the interface or view at this point. Still, many student teams used 'database' as a touchpoint, which is incorrect since the end user never really interacts directly with the database. On the other hand, students were able to correctly define the EM stages and development over time. They also identified most user actions and contexts adequately.

B. Sample Sketches Created by the Students

Figure 3 shows three examples of sketches created by the student teams. Most students used paper-based sketches. Many teams represented drop-down lists with sticky-notes containing sample data. The sketch flow was represented with arrows, which became messy and confusing when there were many sketches and many interactions among them. Students complained that they had to think carefully where to place each sketch in the paper poster to avoid too many crossed lines. Some even said they had to erase and redraw a few sketches to achieve the best possible position layout.

It can be observed from Figure 3 that a student team was using a tablet to show the EMs during the review session (middle image), while other team used a printed copy (bottom image). This is because students were asked to make the relationship between EMs and sketches explicit, i.e. for each EM, they had to describe which was the corresponding sketch sequence. This made them think about the interaction flow of the application early in the design process.

Furthermore, we noted that students were able to apply most of the visual design principles in their sketches. Yet some forgot to name and place buttons consistently across sketches.

C. *Experience Maps Feedback and Evaluation*

Several teams brought their initial experience maps to consultation with the teacher during office hours. The teacher

Experience stage	Define dates	Request dates	Get authorization or rejection
Context	External	Internal	External/Internal
Development over time	⟳	→	→
Touchpoints	• Personal calendar • Work calendar	• View in database	• Administrative employee • Email • Database
Actions	• Search for viable personal dates • Search for viable work dates • Decide on dates to request	• Choose dates • Send request	• Get request response (authorization or rejection) • Reschedule vacations (if needed)

Experience stage	Search for layaway	Data display	Enter a payment
Context	Internal	Internal	Internal
Development over time	⟳	→	⟳
Touchpoints	• Computer in store • Layaway database	• Computer in store	• Computer in store
Actions	• Search within a list of layaways, sorted by customer name • Search by invoice number	• Display data related to the selected layaway	• Subtract the amount paid to the layaway balance • If balance is zero, mark selected layaway as "finalized"

Fig. 2. Examples of EMs created by the students.

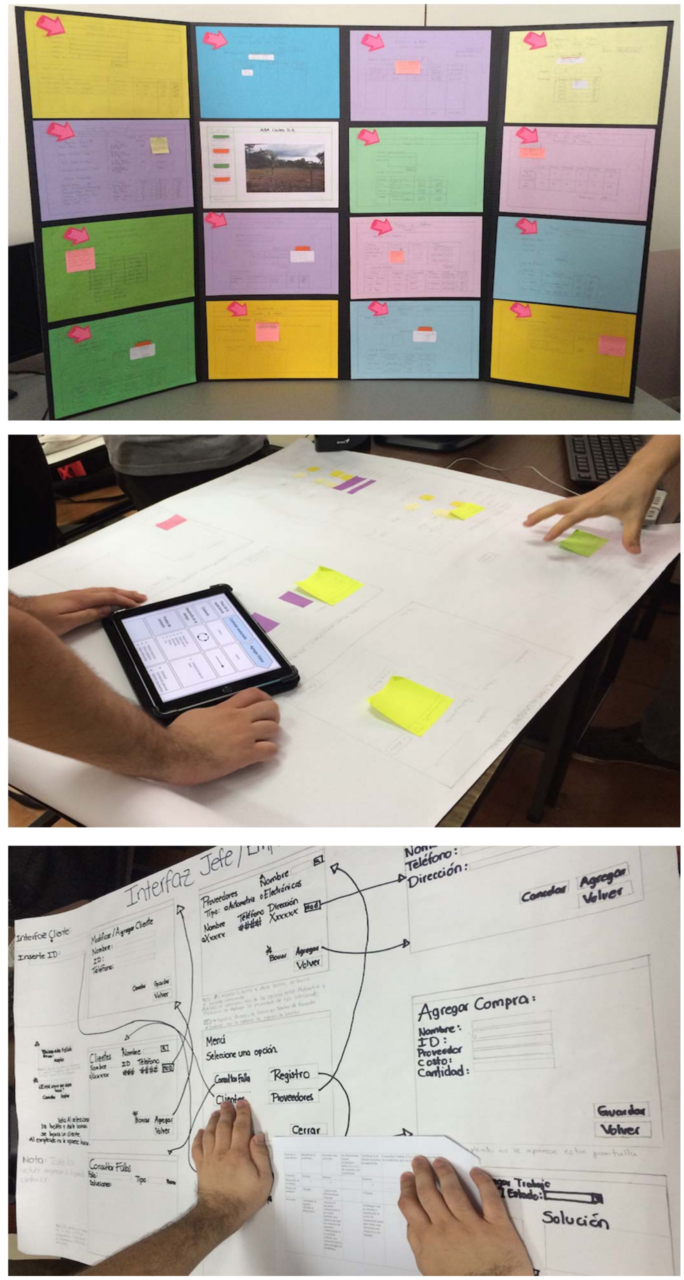

Fig. 3. Examples of sketches created by the students.

gave them feedback on things to improve or misconceptions. The teacher often had to reiterate the need to think about the system from the user's point of view rather than from the developer's viewpoint. Having the students change their mindset to worry about the user experience was a real challenge. The following questions helped students reflect about their user experience design: Why is the user performing this action in the context of the given flow? What triggered the user to start this flow? and If you were the user, would you like this kind of interaction with the system?

Afterwards, a formal review session was held at the end of the Requirements and design phase, where the experience maps were evaluated and graded by both teacher and UX collaborator.

D. Sketches Feedback and Evaluation

Visits to teacher's office hours were less frequent during the second phase of the project, but sporadic consultations were made. Some were about the sketch flows, particularly, how to draw the arrows among all the sketches without doing a mess. Some other were about the sketches themselves, essentially asking for a pre-review. Some teams had inconsistencies naming buttons that performed the same action in different interfaces. Others had inconsistencies among flows, e.g. one flow needed an extra click or an extra interface to accomplish the same action than other flow. Likewise, a formal review session was held at the end of the *Interface design* phase, where the sketches and their flows were evaluated and graded by both the teacher and the UX collaborator.

VII. Assessment and Findings

We evaluated the use of EMs and sketches through a student survey and a teacher's assessment of benefits and limitations. The student survey was designed by the teacher to assess the learning and teaching process, including the project. It contained a total of 35 questions, from which 3 were related to the use of experience maps and sketches in the project, and 4 were generic about the course project. Students were asked to complete this survey online by the end of the semester. Participation in the survey was voluntary and anonymous. In the first semester of 2016, 21 students participated in the survey, whereas in the second semester of 2016, only 12 students did. No data is available for the fall semester of 2015.

A. Findings from Student Survey

The first question related to EMs was "What do you think about creating *user experience maps* to specify the requirements of an information system?" Since this was an open question, we grouped student responses in three discrete categories: positive, neutral or negative. Categorized student responses are summarized in Table 1. We can observe from this table that at least 85% of the students valued the creation of EMs positively, while only 10% had a negative opinion. Within the positive category, we created two subcategories: 'Pure positive' and 'Positive with opportunities for improvement'. In the case of the first semester of 2016, 47% of the responses were pure positive while 38% were positive with opportunities for improvement. In the case of the second semester of 2016, 50% were pure positive responses and 50% were positive with opportunities for improvement. An example of an opportunity for improvement indicated by students was that EMs and sketches should be explained in class, instead of relying only on the video.

A similar question was posed in the first semester of 2016: "What do you think about creating *sketches* to design the interfaces of an information system?" Being an open question as well, student responses were grouped in three categories: positive, neutral or negative. Categorized student responses are shown in Table 2. This question was not included in the survey from the second semester of 2016, thus only data from the first semester is shown in the table. We can observe from this table that all students valued positively the creation of sketches for interface design.

The second question related to EMs was "In which of the following aspects helped the user experience maps and sketches?" This was a multiple-choice question with seven options to choose from. Figure 4 shows student responses to this question in percentage form (from the total number of students, how many chose each aspect). It can be inferred from this figure that EMs helped students mainly with understanding user's needs, understanding the purpose of each interface in the context of the entire solution, and having a clear idea of the interaction between the user and the system.

One of the generic questions regarding the project was "What effect did the development of a real project have on your motivation towards the course?" Table 3 shows the percentage of student responses per effect. It can be observed from this table that a vast majority of students were highly motivated with the development of a real project. In this case, students were assessing the project as whole, not specific parts like EMs or sketches, but we might assume some that if they were motivated is because they like all of it. To obtain further evidence about this, we analyzed two of the final questions of the survey, which asked students what they liked the most and the least in the course. Figure 5 shows the percentage of student responses that explicitly mentioned the project or the interfaces for each of these questions, by semester. We can see from this figure that during the first semester of 2016, 33% of the students singled out either the project or the interfaces as the aspect they liked the most in the course. During the second semester, 33% of the students said the project was what they liked the most. Other aspects students mentioned they liked the most were the labs, the teaching methodology, the classes, SQL queries and other SQL-related aspects. On the hand, in both semesters, about 16% of the students were not happy with the project or the EMs or the interfaces. Other aspects students mentioned they liked the least were quizzes, tests, and theoretical parts (normalization and physical organization of files and indexes on disk).

B. Findings from Teacher Assessment

From the teacher's perspective, the main benefit was that project prototypes had consistent, neat and usable interfaces, with very few minor usability problems. This was a major departure from previous semesters, and directly benefited the client. Another advantage of having students create experience maps and sketches for their projects was that it forced them to

TABLE I. STUDENT PERCEPTION OF EXPERIENCE MAPS

Response category	I semester of 2016	II semester of 2016
Positive	85%	100%
Neutral	5%	0%
Negative	10%	0%

TABLE II. STUDENT PERCEPTION OF SKETCHES

Response category	I semester of 2016
Positive	100%
Neutral	0%
Negative	0%

TABLE III. EFFECT OF A REAL PROJECT DEVELOPMENT ON STUDENTS MOTIVATION TOWARDS THE COURSE

Effect	I semester of 2016	II semester of 2016
It motivated me a lot	76%	92%
It motivated me a little	24%	8%
It did not motivate me	0%	0%
It demotivated me	0%	0%

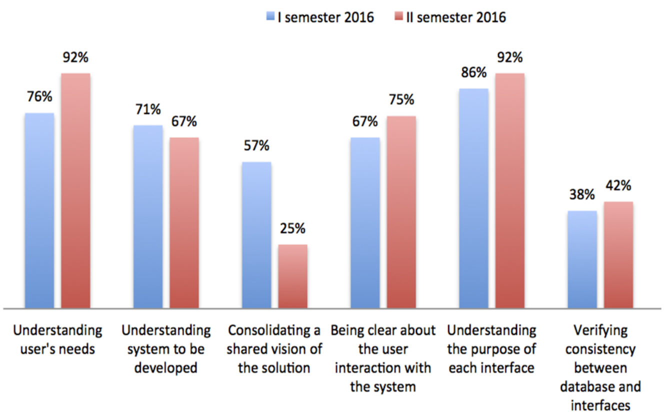

Fig. 4. Contribution of EMs and sketches across several aspects.

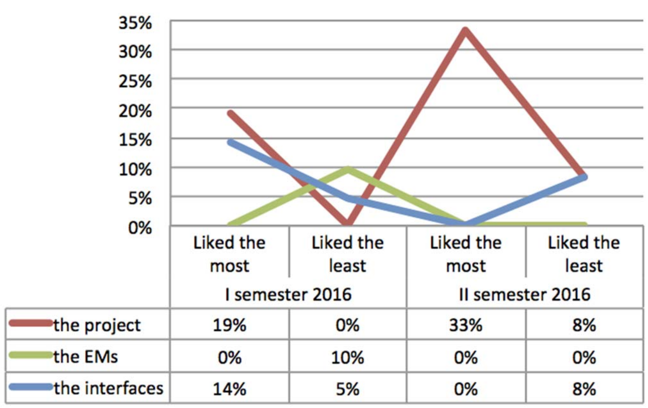

Fig. 5. Percentage of student responses that explicitly mentioned the project or the interfaces in each for these questions, by semester.

think about the user's needs and interaction with the system, in order to design a user good experience. In the process of doing that, they obtained a clearer picture of the system as a whole, and the role of each interface within the experience flow. In short, using EMs as a graphical representation of user requirements helped students better understand and visualize the operations and data needed at each stage, which in turn allowed them to design better interface flows that resulted in a good user experience. Moreover, the guidelines given to students on visual design principles and usability heuristics helped them to design aesthetically appealing and consistent interfaces.

The main limitation was the difficulty students had grasping the essence of EMs and applying it to their projects. The teacher had to explain what EMs were meant for, the need for external stages and other aspects, to different students who visited her during office hours requesting feedback on their EMs. It was very tiring to repeat these things to different teams of students since there was no spare time in class that could be spend on these. Beyond the extra burden on the teacher, what was most worrying was that watching the video was not enough for students to understand EMs completely (evidenced in the survey). Nonetheless, once students were given feedback on the EMs they created for their project, a significant improvement was observed, i.e. they were able to apply EMs concepts correctly.

Due to the aforementioned limitation, the teacher considered that the investment needed for students to correctly apply EMs was too high, and consequently stopped using this technique in the course. After some semesters trying to find alternative approaches, she introduced user stories —instead of EMs— to capture user requirements, and continued using sketches. This is a work in progress that started the first semester (spring) of 2019.

VIII. CONCLUSIONS

We presented an experience report on the use of *experience maps* and *sketches* in an undergraduate database course project, for three semesters. The main reason for introducing these techniques in the course was to improve the quality of the projects. The design and coupling of these user-centered design techniques in the course was explained in detail. In particular, we described what type of experience maps and sketches were used in our course, as well as the resources provided to the students. Additionally, we explained how the course project was adapted to accommodate EMs and sketches, in terms of project phases, deliverables and evaluation. Then we offered implementation details and sample works created by the students. Our experience was assessed from the students' perspective by means of survey, and from the teacher's perspective by an assessment of strengths and limitations.

Our results show that at least 85% of the students valued the creation of experience maps positively, while 100% valued the creation of sketches positively. Students reported that EMs helped them the most to understand user's needs, each interface's purpose in the context of the entire solution, a well as the user's interaction with the system. The main limitation found by the teacher was the difficulty students had grasping and applying EMs concepts. On the other hand, the main benefit reported by the teacher was that project prototypes had usable interfaces, with very few minor usability problems. Using EMs as a graphical representation of user requirements helped students better understand and visualize the operations and data needed at each stage, which in turn allowed them to design better interface flows that resulted in a good user experience. Furthermore, guidelines on visual design principles and usability heuristics helped students to design aesthetically appealing and consistent interfaces.

ACKNOWLEDGEMENTS

We thank Mr. Constantino Bolaños, our UX collaborator, for his ideas on designing and carrying out this work.

REFERENCES

[1] Blink. The Fine Art of Creating Experience Maps. URL: https://blinkux.com/blog/the-fine-art-of-creating-experience-maps/ Last accessed: 10/2/2018.

[2] Nielsen Norman Group. UX Mapping Methods Compared: A Cheat Sheet. URL: https://www.nngroup.com/articles/ux-mapping-cheat-sheet/ Last accessed: 10/2/2018.

[3] Visual Paradigm. What is User Experience Mapping? URL: https://www.visual-paradigm.com/guide/user-experience-mapping/what-is-user-experience-mapping/ Last accessed: 10/2/2018.

[4] Bruna Ferreira, Tayana Conte, and Simone Junquiera. Eliciting Requirements using Personas and Empathy Map to Enhance the User Experience. Brazilian Symposium on Software Engineering, Brazil (2015) DOI: 10.1109/SBES.2015.14.

[5] PC Anitha and Beena Prabhu. Integrating Requirements Engineering and User Experience Design in Product Life Cycle Management. International Workshop on Usability and Accessibility Focused Requirements Engineering, Zurich (2012) DOI: 10.1109/UsARE.2014.6890994.

[6] Juho Heiskari, Marjo Kauppinen, Mikael Runonen, and Tomi Männistö. Bridging the gap between usability and requirements engineering. Proceedings of the IEEE International Conference on Requirements Engineering, USA (2009) DOI: 10.1109/RE.2009.37.

[7] David Callele, Eric Neufeld, and Kevin Schneider. An Introduction To Experience Requirements. IEEE International Requirements Engineering Conference, Sydney (2010). DOI: 10.1109/RE.2010.55

[8] Anirudha Joshi, N.L. Sarda, and Sanjay Tripathi. Measuring Effectiveness of HCI Integration in Software Development Processes. Journal of Systems and Software, vol 83 (11), 2010. DOI: 10.1016/j.jss.2010.03.078

[9] Carmelo Ardito, Paolo Buono, Danilo Caivano, Maria Francesca Costabile, and Rosa Lanzilotti. Investigating and promoting UX practice in industry: An experimental study. International Journal of Human-Computer Studies, vol. 72 (6), 2014, pp. 542-551. http://dx.doi.org/10.1016/j.ijhcs.2013.10.004

[10] Thomas Buchmann. Towards Tool Support for Agile Modeling: Sketching Equals Modeling. Proceedings of the Extreme Modeling Workshop, Austria (2012). DOI: 10.1145/2467307.2467310

[11] Dustin Wüest, Norbert Seyff, and Martin Glinz. FlexiSketch: A Mobile Sketching Tool for Software Modeling. In: Uhler D., Mehta K., Wong J.L. (eds) Mobile Computing, Applications, and Services. MobiCASE 2012. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 110. Springer, Berlin, Heidelberg.

[12] Dustin Wüest, Norbert Seyff, and Martin Glinz. Sketching and notation creation with FlexiSketch Team: Evaluating a new means for collaborative requirements elicitation. IEEE International Requirements Engineering Conference Ottawa, ON (2015). DOI: 10.1109/RE.2015.7320421

[13] Alfredo Motta, Nicolas Mangano, and André van der Hoek. Lightweight analysis of software design models at the whiteboard. International Workshop on Modeling in Software Engineering, San Francisco, CA (2013). DOI: 10.1109/MiSE.2013.6595291