

Group 6 Experience Report

Izabell Arvidsson (izabellarvidsson@live.se),
 Moa Berglund (moa.berglund00@hotmail.com),
 Erik Eriksson (erik.g.eriksson1@gmail.com),
 Oscar Forsyth (oscar.forsyth61@gmail.com),
 Simon Larsson (zico.xp.larsson@gmail.com),
 Thushini Ranasinghe (thushinirana1998@gmail.com)

Abstract—This document describes our experience of creating a requirements document for BarAdvisor, an app used for searching for the cheapest bars. BarAdvisor is targeted toward students, who often are on a tight budget, that still want to go out for something to drink. This experience report is mainly focused on two parts, the techniques used and what we learned, sections 2 and 5 respectively. Each part includes motivations and reflections for the techniques used and the learning outcomes.

1 INTRODUCTION

BarAdvisor is an application that seeks to assist students in navigating information about bars in their immediate vicinity. Because students frequently live on a tight budget, the program prioritizes costs to help students avoid overspending. The app should be able to filter on the precise drink in mind to deliver the most transparent bar recommendation possible. Similarly, the app should be dynamic, presenting bar offers and limited-time discounts. This report summarizes our experience while creating a requirements document for BarAdvisor, including a description, motivation, and commentary on the approaches we utilized, as well as thoughts on what we learned.

2 TECHNIQUES

In this section of the report, we specify, analyze and justify each approach employed during the project. The section begins with covering the techniques we used to elicit information about the domain, following with the techniques we used for specifying requirements based on this information, and ends with our approach to prioritize these requirements.

2.1 Elicitation

Before establishing the product's requirements, elicitation is a critical step that must be completed in order to thoroughly understand the demands of many stakeholders. Elicitation, which is a key aspect of the requirements process, may be done using a variety of ways. We used three elicitation techniques: brainstorming sessions, a questionnaire and lastly interviews with the stakeholders. In the sections below, these techniques are discussed and reflected on.

2.1.1 Brainstorming

Reasons why we chose brainstorming:

- Very easy and quick to organise
- Allowed us to put together a lot of ideas quickly

- Allowed for open discussion so anyone could share their opinions or ideas

The first strategy utilized in the project was brainstorming, which is a solid elicitation tool for generating diverse viewpoints and it was used to generate alternative project ideas. This technique came naturally to the project since there are six participants in the project, hence resulting in multiple perspectives. This method let us as a group be creative about what we could come up, with which led to plenty of different ideas. However, we noticed that once in a while we would get stuck on one specific idea and the specific details behind it. This led to unnecessary debate and a lot of wasted time. Still, when discussing these ideas we were able to clear up many conflicting assumptions different group members had.

2.1.2 Questionnaire

Reasons why we chose questionnaire:

- Quickly obtained quantitative data
- Requires less commitment from participants
- We wanted to get an idea as to if students would use our app and experience the problem we are trying to solve
- Possible to obtain a smart unique idea
- We wanted to see if the ideas we had were reasonable

The decision to use a questionnaire to the student target group was based on the fact that this technique collects quantitative data and offers different perspectives, even though questionnaires are not the most effective way to elicit requirements. The questions for the questionnaire differed from "are you a student?" to "Do you find it easy to find bars that suit your student's budget?", where the focus was on the usage of the app and its core functionality. This technique was also used for how simple it was to distribute. One difficult aspect of this technique was to write questions that were relevant and could help the process move along.

A challenge with questionnaires is that the one conducting the questionnaire (we, in this case) do not have

direct contact with the participants, which can lead to misinterpretations in both directions. The one answering the questionnaire might interpret the questions in another way than what was supposed by us, and we might interpret the answers in another way than what was supposed by the answerer. To reduce this risk, we tested the questionnaire before distributing it. We tested it by talking through the questions with a few persons in the target group, to make sure that they interpreted the questions in the way we wanted, and if not the question got reformulated. Of course, this does not eliminate all possibilities for misinterpretations. As a way to recover from this risk of misinterpretation, semi - structured interview with a representative from the target group answering the questionnaire was also conducted (explained in next section).

2.1.3 Semi-structured Interviews

Reasons why we chose interviews:

- We were able to directly communicate with important stakeholders in a controlled setting
- Establish a relationship with stakeholders
- It can possibly generate ideas about the future system
- Figure out where the conflicts lie
- Figure out what is realistic for the system
- Easy to expand if needed

Semi-structured interviews were performed with each stakeholder; a student, a Student Union representative, and a bar owner. These interviews were conducted in order to gain better knowledge from the three stakeholders, their problems experienced in the area, and their expectations from the system. Semi-structured interviews were held by the project to get the most out of the stakeholders. This means that the questions that were prepared were not finite questions. So when extra questions came up during the interview, they were addressed.

When interviewing the student stakeholder we noticed that they did not seem to be fully committed to bringing ideas to the table. Instead, they gave short answers to open ended questions. This could be due to their lack of knowledge of the system and the lack of preparation beforehand. As a result, the answers given were not incredibly useful. If we were to do this again or if we are to do interviews in the future, then we would make sure the stakeholder is committed and also prepared for the interview. A difficult aspect was coming up with questions that were both fascinating and relevant to the project's goals. However, because the interviews were semi-structured, the interviewers were free to ask questions on the spot.

When interviewing the student union representative, it was the other way around. He had many interesting thoughts and was engaged in the meeting, but unfortunately, we had to make the session short since he did not have more time. A learning outcome from this is to not schedule semi-structured interviews with open questions and passionate participants during a stressful time.

2.2 Specification

This section will cover the specification techniques used to specify the system. Specification is the process where all the

elicited ideas, problems and stakeholders wishes are stated as requirements or expressed in another descriptive way. Since requirements can be of different level of complexity and require different knowledge, we used various techniques to specify the requirements. The techniques applied is: Context Diagram, Task Description and Feature Requirements.

2.2.1 Context Diagram

Reasons why we chose Context Diagram:

- Overview of system interactions
- Outline the scope
- Relatively easy to read and interpret for everyone involved
- Summarize the key application features

The first specification technique utilized was drawing a context diagram, showing a general picture of the supposed interactions between the system and external actors. By doing so, the scope of the project in addition to the main functionality was clearly outlined. Not only did this help ensure a shared understanding within the group, but also for related parties, where the context diagram was used to convey the general idea during interviews. Further benefits of the context diagram include serving as a frame of reference when building data flow diagrams in the future. Thus the creation of the context diagram facilitated the work process by establishing a clear outline upon which to build.

2.2.2 Task Descriptions

Reasons why we chose Task Descriptions:

- A relatively easy way to list early functional requirements
- Easily understandable for different people involved
- Helps to get a feeling about how the system will be used in reality
- Helpful to explain a complex situation in the system
- Helpful to state and cement the main tasks

The second specification technique used in the RE process was task descriptions, which provide a structured overview of expected tasks to be performed in the app. The reason for using this specification technique early on is to derive functional requirements, which initially are prioritized to elicit over other requirements. Furthermore, due to the relative simplicity of task descriptions, they could favorably be presented to stakeholders and potential users of the system to inform of supported functionality and validate proper workflows respectively. Additionally, the task descriptions were grouped under suitable work areas, each providing a background of the environment where their subset of tasks is relevant. As a result, the domain is further specified which helps form a better understanding of the system's application in practice. Nevertheless, creating these work areas proved to be a difficult endeavor since there are several ways in which to divide the system, and plausible arguments could be made for any of these. Similar uncertainties were encountered with the task descriptions too, such as how many variants would be required before the task description is considered too extensive and in need

of being divided. In the end, though, the relatively troublesome process should be worthwhile as the task descriptions streamline future work.

2.2.3 Feature Requirements

Reasons why we chose Feature Requirements:

- Easy to formulate and interpret
- Precise (easy to verify)
- Not time consuming to write (if used for simple/basic features)

Feature requirements was chosen as the second technique to specify functional requirements, since we wanted a simpler way to express the requirements that are less complex and more self-explanatory (in comparison to the requirements described by task descriptions). However, since they are so easy to formulate, it can easily happen that one formulates more requirements than necessarily/aligned with the business goals. We tried to avoid this and focused on only the ones we find strict necessarily. It is though challenging to determine which requirements actually are strict necessarily and correlated to the business goals. For example, we have a feature requirement that covers to fulfill standards for disabilities. This is actually not directly correlated to any business goal, but in the meantime, the student union values this very high so that every student can use the app.

Another thing that was challenging for us was when talking about features, the discussion could go on forever (both with stakeholders and within our group), since the "nice-to-have" features tend to pop up at a high pace. We have balanced issues like the mentioned ones by always questioning if the features are critical, and ended up with a small list of important feature requirements.

2.2.4 Data Model

The specific data model created was an ER-model.

Reasons why we chose a Data Model:

- Gives a complete picture of all data and its relationships
- Prior knowledge of using it in the group (did not have to spend time learning a different approach in detail)

A data model is used to specify the data for a product. The way this is done in an ER-model is that we show the things we will have data about, the entities, as rectangles. Each entity have properties which corresponds to the actual data we will have for the entities, shown as text labels next to each entity in the model. The relationships between different entities are shown as diamonds or triangles. They show how entities are linked together, for instance how a rating must be connected to a single user and a single bar.

Specifying data using an ER-model gives a complete view at the data that shall be stored in the system. In our case, the ER-model can be directly translated to a relational schema for use in the relational database-part of our system. The only downside to using a data model for data specification is that it is hard to interpret and understand if you are not familiar with them. This became clear when we prioritized data requirements from the stakeholders perspective.

Because a, for instance, bar owner does not know how to interpret the ER-model they cannot prioritize it to the extent that they perhaps should have.

2.2.5 Quality Grid

Reasons why we chose to create a Quality Grid:

- To prioritize quality factors and discover possible concerns or conflicting assumptions
- Provide clear definitions for the groupings of QRs
- Provides a clear picture of prioritizations
- Relationship between business goals and quality factors

In order to prioritize the general quality factors, a Quality Grid was used. The quality factors included in the grid were based on McCull and Matsumoto's paper so as to capture the complete spectrum of possible qualities. However, we felt that *installability* was relevant for the context of BarAdvisor and therefore it was also added. These quality factors were then discussed based on the definitions provided in the paper, and finally assigned a priority on the scale: *Critical, Important, As usual, Unimportant and Ignore*. Although a straight forward process, it should be noted that quality factors ranked at *As usual* can still have QRs associated with them. *Ignore* on the other hand, meant that those quality factors were completely irrelevant for the project, and thus would not have any QRs. Any rank outside of *As usual* and *Ignore*, warranted a short explanation for why it diverged from the norm. By doing so, a clearer mapping was established between business goals and QRs.

2.2.6 Quality Model

Reasons why we chose to create Quality Models:

- Emphasize certain requirements
- Provide specifications for possibly ambiguous QRs
- Improve QRs (update after finding inconsistencies)
- Prior knowledge of using it in the group (did not have to spend time learning a different approach in detail)

Another technique related to QRs is to create quality models. This process entailed choosing a relevant QR and expand on the details concerning it, such as the overarching quality factor it was derived from, sub-category, attributes, metrics and an executable test plan. The resulting quality model would then provide a clear view of the requirement, including necessary details and its relating categories, to developers and system testers.

The experienced downside of creating quality models was the process in and of itself. Firstly, it was not a top-down process as was expected, starting with the quality factors from the Quality Grid and then gradually moving down in granularity, due to the difficulty in eliciting sub-categories and measurements while still unaware of the objective. Instead, QMs were directly elicited from the quality factors, to then move up in abstraction levels, only to go down into minute details afterward. Secondly, the existence of a quality model could imply two different things, either that the QR was of extra importance, or that it needed extra details for the sake of clarity. Thus, when multiple quality models exist, the reader may interpret the reason for their

existence differently, creating more confusion. Thirdly, there were a lot of steps in between the origin quality factor and the final QR that could be "wrong", especially in regard to sub-categories considering they are open for anything and any interpretation.

2.3 Prioritization

When all requirements are specified, the next step of the requirement process is to prioritize these. The goal of the prioritization phase is to get a better understanding of where the most important value lies in the application, as a base for planning the development process. It is important to involve all stakeholders in the prioritization process, since different aspects can be variously important for each stakeholder, and they all need to be satisfied in the end.

The techniques we used for prioritization was Grouping, The 100-Dollar Test and Top-Ten Requirements.

2.3.1 Grouping

Reasons why we chose grouping:

- Visual way to provide overview (we used sticky notes with color coding)
- Not that time consuming in comparison to giving each requirement a unique value

The first technique used for prioritization was Grouping. This was a way for us to sort all the ideas that have come up during the elicitation process. We placed each requirement in one of these four groups: must have, nice to have, could have, and out. Then, we kept the must have requirements.

One drawback in the way we conducted it was that we did not involve any stakeholders. This was due the fact that our stakeholders don't want to spend so much time on this fictional project, so we thought this first and rough prioritization could be done by ourselves. Of course, we thought about what we believe they would have strong opinions about based on the contact during elicitation. If we would do this in a "real project" we would make sure to include users/developers and financial representatives in this since their opinions are of high value.

2.3.2 The 100-Dollar Test

Reasons why we chose The 100-Dollar Test:

- Each stakeholder gets directly influence
- Easy for stakeholders to understand
- Gets a clear and precise overview of each stakeholders priorities

When executing this prioritization technique, each stakeholder representative got 100 dollar to distribute as they wanted through all the requirements. It can be difficult to compare different kind of requirements, like what is most important; security about passwords or being able to rate a bar? To reduce this problem, we did a 100 dollar test for each of the two major requirement categories we have; functional requirements and quality requirements.

One difficulty we encountered within this technique was that the requirements can be interpreted differently by different stakeholders. This can lead to that the stakeholders give values based on a wrong interpretation, and this will

of course bias the outcome. What we have learned from this, is how important it is to specify requirements in an transparent way that ideally only can be interpreted in one way. An approach we could have used to prevent this problem is to talk through the requirements with each stakeholder, making sure they interpret it in the desired way. This is though time consuming.

2.3.3 Top-Ten Requirements

Reasons why we chose Top-Ten Requirements:

- A way to gather only the most important requirements
- Clear what needs to be fulfilled to get each stakeholder satisfied about the solution

After receiving information about the requirements in relation to each other from the previous techniques, we wanted to elicit which of the requirements are the most critical to the stakeholders. Each stakeholder representative got to chose their top ten requirements, without any internal order or categorization. This was a good complement to the 100 dollar test since that was limited to prioritize inside the two categories Functional Requirements and Quality Requirements, which then does not display the overall most important requirements. Since the Top-Ten approach did not force any prioritization between the top ten, it was easier for the stakeholder to mix the categories here. In the same time, it was preferable even for us that the stakeholders did not prioritize here, since then it could be some dissatisfaction with the final product if for example some stakeholder got their number one, and the other got its number 8.

3 GROUP DYNAMICS

In this section we will briefly talk about how well the group worked together as a team and the different obstacles we had to overcome. As a team trying to achieve a goal we think we worked together quite well overall. We think this because we were able to separate the work load for each team member so that everyone had to do their fair share. Where one person thought they knew more on a subject, they said so and took more responsibility for that task. This led to very few instances were the work was not completed to a high enough standard or not completed at all. We think that everyone was able to share their ideas and opinions with the group due to the importance we placed on respecting each other. However, it is quite hard to tell how much a group member is holding back and even though we thought we had good discussions, it could have been even better if we got together outside of school to do some form of team building. This would strengthen our relationship to one another and most definitely lead to higher quality discussions.

Like any group there arose conflict of opinion. These conflicts were usually about how the application should work. To solve these issues we used, what we wrote in our team agreement, democracy. When one person had a strong opinion about a matter they would present their idea to the rest of the group and if the group vote favoured the new idea then we would switch. However, since we are six people there is a possibility that the vote will be even. This

happened only once and in this case we solved it by letting people express their thoughts about the conflicting ideas for about ten minutes. Another vote was held and this time it was not even resulting in a final decision.

4 ETHICS

Although the high-level aim of the project is to support students to make economically beneficial decisions when planning a bar visit, there is an important side effect our idea could trigger; that students visit more bars (and probably consume more alcohol) due to being framed with new bars or appealing advertisements/offers in the app. This is contradictory to the aim, since it could mean students actually spending more money at bars with the app than without. This would of course be beneficial to the bar owners. So if that would have been the primary intention with the project, some community of bar owners would probably be a suitable driving stakeholder, instead of the student union. Although, when conducting the project, we limited the scope to the situation when the student already has decided to visit a bar (without any influence from the app). Thereby the ambition to help students to make the best cost-effective decision can be reached with the app, without encouraging increased alcohol consumption.

5 LEARNING OUTCOMES

This section aims to summarize our general learning received during the project process. It will cover all the four central phases in the requirement process; Elicitation, Specification, Prioritization and Validation.

Common for all phases are that we have learned several techniques to approach the certain phases. We have also payed attention to advantages and disadvantages with the techniques to optimize the way they are used, and in which context they deliver at their best ability. Another gained knowledge applicable to all phases is that different techniques can be used to complement for each others drawbacks. For example, as mentioned, a questionnaire and a interview can compensate for each other by a interview covering the qualitative answers and a questionnaire delivers quantitative data.

For R2 we have started to think about some key points we have learned in each process, and they are stated below. For R3, we will extend this section by elaborating more on these.

5.1 Elicitation

- Importance of preparation before a interview (both estimate the needed time, and have questions that will engage the participant)
- The group is a great can be a great resource for ideas (not only stakeholders)
- The importance of be precise about both what is asked and what is answered to avoid misunderstandings
-

5.2 Specification

-
-
-
-

5.3 Prioritization

- Risk of stakeholders interpret requirements differently and thereby prioritize based on different assumptions
-
-
-

6 CONCLUSION

So far, it is still too early to write an appropriate conclusion for this experience report.