

Experience Report

Requirements specification for Student Accommodation Matcher System

Abstract -

Keywords -

I. INTRODUCTION

The high demand for accommodation is a worldwide issue at this time, and Gothenburg is not an exception. Gothenburg is the second biggest city in Sweden where students come to study every year. However, due to the high demand for accommodation students are facing difficulties to find accommodation for the period of their study. Existing agencies often use queue systems that are long and frustrating, which may lead to students not getting accommodation in time or not getting it at all. Students have to look for accommodation in several places where other people already have been registered and collected queue points for a long time. Because of this students want a system where they can find accommodation as soon as they have been admitted to the university without the need to be in a queue for a long time before coming here.

II. TECHNIQUES

The team used the following methods to document the requirements for the project idea selected.

A. Elicitation

Eliciting information from related stakeholders regarding their needs and expectations is one of the key initial tasks that need to be done. Two of the elicitation methods used are explained below.

a) Interviews

Lauesen [1] lists interviews as one of the methods to involve the stakeholders and suppliers in the requirements gathering process. Lauesen suggests interviewing both expert and ordinary users alike [1]. Students and landlords were selected as the primary stakeholders to interview. The team created a list of questions that it wanted to be answered. The team tried to make the questions as open-ended as possible. 21 people (16 students and 5 landlords) were interviewed in the process. The interviews

provided different perspectives regarding the needs of the stakeholders. Even the same type of stakeholders have different needs and expectations depending on their situation. For instance, the needs of an international student differed from those of a domestic student even though they are of the same stakeholder type. Most of the responses were as anticipated by the team. Since the team members had some prior knowledge of the problem domain, there was some bias regarding anticipation of the responses. Follow-up interviews are planned with the stakeholders with detailed questions to refine the requirements documentation prepared by the team.

b) Brainstorming

Since the team members have prior knowledge in addition to the knowledge they gathered through interviews, brainstorming is an appropriate technique to come up with new ideas. In this technique, the team worked in two phases; first a moderator gathered ideas from each team member, and then each idea was discussed and analysed.

The team then combined various responses from the interviews and brainstormed the possible requirements for the system. The team devised additional requirements besides the requirements obtained through interview. The members are motivated to stimulate new ideas with respect to the problems that they gathered from their interview sessions. Any new ideas raised by the team members were noted without any criticism. This motivated every team member to come up with his/her own new ideas without being shy or afraid.

The new ideas and requirements raised through brainstorming were discussed and analysed by the team members to filter the ideas and requirements that are appropriate to the project at hand. After thorough discussion and analysis the team members selected the most important ideas and requirements that pertain to the proposed idea.

B. Specification

a) Context diagram

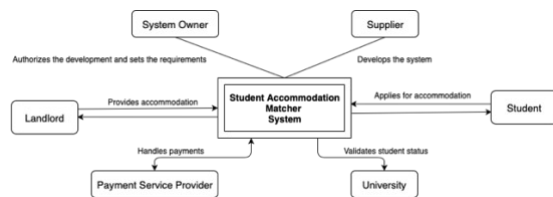


Fig. 1 Context diagram

b) Task description

The description of the tasks which needs to be implemented by the system are divided into work areas to complete the task description process. The work areas are further divided into subtasks to show the working and the whole process of specific work area. The work area includes the number of users using or performing a subtask along with the information of start and end of the task process. It also shows the frequency of a user performing a specific task and the frequency of task being used. The tasks that the system will perform are divided among five general work areas along with subtasks to show the working of the system.

C. Prioritization

There are two methods for requirement prioritization used. MoSCoW and Scorecard.

a) MoSCoW

MoSCoW, a numerical assignment technique [2], is used because it is easy to apply when managing requirements. It gives us a clear overview of what we as customers really want to have included in each software release. The team grouped the requirements in groups according to highest priority that the system must have in order to satisfy its core value. The complete requirement list is used and we grouped requirements according to four categories: Must, Should, Could, Won't. On every requirement listed, each stakeholder puts their own estimation of importance. Finally, requirement importance was set on based on what chosen level of importance dominates in the stakeholder selection process.

b) Priority Scorecard

To support the prioritization method mentioned above, Scorecard technique is used [3]. Instead of only choosing requirements randomly based on our opinion, this technique helped us to include various factors important for the system. By introducing various factors applied on the system, we get a deeper understanding of what requirements will need more financial support and what areas are affected with each requirement. The results of prioritization using this methodology indicates if requirements prioritized using MoSCoW actually meet the stakeholder's needs. When using this methodology,

the stakeholders decided what are the most important criteria that impact the cost-value relationship. The goal was to have the most beneficial requirements on top, followed by security as the next most important goal.

III. LEARNING OUTCOME

In this section, lessons learned from the project are presented.

The team experienced how difficult it is to write requirements specification. Lauesen describes that a good requirements specification, for example, is complete, unambiguous and verifiable [1]. These three examples are the ones that the team has discussed the most. The team found that it is difficult to know whether the level of detail is good enough to include all necessary requirements. It was hard to balance the level of detail. In other words, it is not easy to know when the specification is complete.

The team experienced how useful different specification techniques are. A context diagram gives a good overview of the system. It is helpful in the way that it describes the system on a high level. For the more detailed levels, task description serves as a good technique for specifying what to include. The team regarded a task as something that should be done without interruptions and that the end of it was regarded as an achievement. This made it easier to keep the tasks on a similar level of detail. Many of the requirements that the team had in mind were difficult to verify. This was handled by creating event and function lists, which were easier to use as checklists for verification.

During the prioritization of requirements, the team faced issues when deciding on what technique to use. With the fact that there are various techniques applicable, the team spent some time to understand which one will suits goals best. The team learned that using only one technique is not enough to prove if the prioritization done is really what the team wants. Including one more prioritization technique on the same requirements gave different point of view what requirements are actually important and it reflected prioritization more accurate.

The team realized the importance of good specification documentation when going through the identified requirements. When working with the requirements specification, the team realized that the requirements were ambiguous initially. This provided the insight that is important to work together with the specification.

IV. CONCLUSION

Requirement engineering is a difficult but an important part of software engineering. It is useful to know the customer's needs, and verify and validate the requirements gathered by the supplier to develop a software. In this requirements specification project, the team has learned different techniques and methods used in requirements specification. Requirements specification styles such as context

diagram and task description are a good way to present the high-level overview and the detailed description of the system respectively. Interview and brainstorming techniques are used to elicit the requirements of the system. Interview is a good way of gathering requirements and gain a thorough understanding of the domain knowledge. Moreover, brainstorming is also a useful elicitation technique when the analyzers have domain knowledge about the proposed system. In conclusion, the team understood that the choice of the different techniques of requirements specification depends on the nature of the project.

V. BIBLIOGRAPHY

- [1] Laussen, Guide to Requirements SL-07, 2017.
- [2] P. Achimugu, A. Selamat, R. Ibrahim and M. Naz'ri Mahrin, "Information and Software Technology," in Elsevier, Skudai, Johor, Malaysia, 2014.
- [3] . J. J. Phillips,, T. W. Bothell and G. . L. Snead, The project management scorecard, London/ New York: Routledge Taylor & Francis Group, 2011.