Experience report, group 3

1st Mola Ayenew
gusayenmo@student.gu.se

2nd Tommy Ernsund
gusernto@student.gu.se

3rd Pouya Faramarzi
pouyaf@student.chalmers.se

4th Wei Guo
gwei@student.chalmers.se

5th Sisi Lai
lsisi@student.chalmers.se

6th Abhinav Prasad
guspraab@student.gu.se

***Abstract*—This report contains the activities applied in the process of producing a requirement specification for a system for providing temporary housing to international and local students in Sweden. The report will also discuss the choices behind these activities while providing the group's experience and dynamics throughout the course of the project.**

***Index Terms*—requirement engineering, student accommodation, housing, ...**

I. Introduction

The first step for international students when they enter a new country is usually to find a place of residence. In Sweden, this process is significantly harder due to the existing student housing crisis and year-long queuing required to get a contract.The current housing systems for students, offer an edge to students who sign up ahead of their studies and acquire longer queue times, while students unaware of this are forced to look for temporary accommodations. This leads to students getting scammed, paying unreasonable amounts of rent or giving up altogether.

The system envisioned by the project team is to provide temporary means of accommodation to international students travelling to Sweden. In this report we discuss the techniques used and experiences gained while executing this project. Additionally, this report will discuss the challenges faced during the global spread of the covid-19 disease.

II. TECHNIQUES

In this section of the report we investigate and motivate each technique used during the project.

A. Elicitation

A key process to seeking and acquiring requirements for systems is elicitation through the stakeholders. In order to discover what expectations stakeholders put in the system, many different techniques may be applied to not only characterize the stakeholders, but also to understand and getting a deeper knowledge of their domain. This section will discuss the techniques that were applied throughout the project and their advantages as well as disadvantages.

1) Brainstorming: As a part of many creativity techniques, brainstorming is one of the most widely used ones in order to drum up new ideas. A brain storming session is very simple and concise, a clearly defined design problem is given and the group together generates ideas to solve them. As a general guideline for most creativity techniques, a 4-step process should be used.

First, the group devises the design problem and starts generating ideas. Secondly, the group allows for an incubation period where the generated ideas are allowed to grow and be criticized in the back of their heads while going about their daily lives for about a week. In the third step the group regroups and reviews the ideas and refines their choices. Finally, a user story is written and implemented into the project.

The brainstorming technique is a very easy applicable and efficient technique to use. However, it is very limited by the experience of the group and does not promote creativity like other used techniques. Therefore, the brainstorming technique should be used on design problems that are seemingly small or needs to have a solution promptly.

2) Hall of fame: Hall of fame is a creativity technique for unstructured problems with open questions and answers. The goal is assuming that group members are different type of famous people who will solve our challenge of finding new requirements and instigate creativity. It'll force us to think about what will be added and edited to our system from others' perspective.During the hall of fame workshop, our group confirmed positive outcomes that made us decide to add some extra functionality which will make system easy for different users to use.

One outcome was that our users might have different expectations of temporary housing. For instance, scientists might need a quiet and private space to study and work, while a soccer player Ronaldo might want a big house with a nice bathroom mirror. To achieve this, we need to add a filter when user searching for a room to cater to the user preferences. The filter is based on location, standard of accommodation, privacy, price and so on. Student can add different filter while searching in order to get their ideal temporary housing.

Another outcome is to modify our system to make it easier to communicate. Because the system might be hard for people which have limited social experience and English

literacy. Therefore, in order to simplify communicate and usage of the system, we can make some guidance for users and can different language packages for owners and users communicating with each other.

The creativity workshop also brought us some new ideas of our system. Main aim of our system is to provide housing for students in Sweden. However, once developed, we can image that the system can potentially become a social platform to build connection between home owners and students. Furthermore, it can also be expanded to students and home owners outside of Sweden.

3) Creativity triggers: Creativity triggers is another creativity technique which consist of activities and behaviors that stimulate the production of new and useful ideas. Creativity triggers vary widely. Our group held a creative discussion while being focused on the various creativity triggers. Each trigger has a sudden effect to produce a new, and possibly useful, idea.

Creativity session provided many suggestions on how to build our system more simple and useful by thinking using different triggers. The most important outcomes of the creativity trigger session are listed below.

- For Adaptable: Be able to use it across different platforms (OS, Android, Web) and Offer different language packs, integrate google translate.
- For Light: Less dependencies, no unnecessary framework Minimalist and responsive design.
- For Entertainment: Social platform, Seasonal themes, and add some Special emoji.
- For Economical: Don't store unnecessary data, Use cloud hosting - less maintenance and scales accordingly to needs.

There were also a lot of suggestions that were of no use, but for the exploration of the ideas it works very well.

4) Document Analysis: Document analysis is a technique used to describe the act of checking existing documents of comparable business systems to extract information related to the current project.

Our group chose this technique not only because it's time efficient and easy to apply, but compared with other elicitation techniques, document analysis is particularly valuable when there are already existing similar software and websites with a proven proof of concept. With the analysis of these similar software documents, our system can provide more functions and better overall user experience based on the existing software.

Through discussion, our group chose four applications and websites as objects of document analysis: Airbnb, Couchsurfing, Blocket Bostad, and HousingAnywhere. The main problems we decided to analysis were the way they work and their pros and cons during using.

The factors we found that could be used to improve our system were:

- Built-in one-key translation system.
- Have some lists based on the tenants' comments and the location and the purpose of the house.
- Have keywords about the environment. (e.g. Distance to the University, The number of supermarkets and stores nearby.)
- Different possibilities of signing a contract can encourage people to verify their identity

Things to avoid in our system:

- Keep it as a free platform as fees have the risk of turning away a portion of our user group.
- Avoid cluttered or complex design as it may also have the risk of turning away a portion of our user group.

There are still some outcomes that are not listed, but they are still very helpful to the requirement elicitation of our system.

5) Interview: An interview is a conversation for gathering information. A research interview involves an interviewer, who coordinates the process of the conversation and asks questions, an interviewee, who responds to those questions, and in our case we also used a note taker, who listens in during the interviews and takes notes on details that the interviewer might have missed.

The goal of this elicitation technique was to collect qualitative data from our main user group, namely students and home owners, on possible requirements for our system. In order to maximize the feedback gained in this elicitation technique from a few volunteers, a semi-structured interview strategy was chosen.

A semi-structured interview strategy, as opposed to a structured or unstructured one, allows for interviewers to ask questions on certain topics to get a conversation going and ask follow up questions in order to dig deeper into the root of problems to discover harder to gain data. Questionnaires, for example, lacks this ability and was therefore not used in this project. However, this is also troublesome since it means that the qualitative feedback gained through asking follow up questions is limited to the ability of the interviewer.

The structure of the interview included an introductory phase where the interviewer and note taker introduced themselves and explained the purpose of the interview. The interviewee was then notified of the approximate time duration of the interview, that his/hers anonymity is ensured, and that he/she has the right to withdraw from the research at any given time. After that the interview consists of a set of prepared questions that has a warm-up phase, a main body, and a cool off phase.

The warm-up phase includes a set of easy questions to get the conversation going and finding out basic information about the interviewee. We then proceed to the main body where heavier questions will be asked until half way through, the idea of the system is introduced and the interviewee is asked for his/hers response. The final phase includes a series of easy questions to round off and ends with the interviewer and note taker giving feedback on the data gained and thanks the interviewee for his/hers participation.

B. Prioritization

After a variety of techniques for requirement elicitation, we elicitated a lot of requirements. Because of budgetary constraints and tight deadlines, there is a need to do software requirements prioritization as it is impossible to do everything at once and maximizing the effort in all areas is considered very inefficient.

1) Quality grid: Quality requirements specify the degree to which the system must perform its functions. For measure quality requirements, there are many lists of quality factors. At first, we started with the McCall factors. But considering strict rules of Sweden housing, compliance could be an issue, so we added a compliance factor from ISO 9126. For finding the important factors in a systematic way, our group decided to use a quality grid to assess their degree of importance.

Quality grid is a good way to evaluate quality factors. During the process of selecting the factors, we checked all the possible quality requirements and found factors that could be ignored. Also in assessing the importance, we discussed a lot, because we need to keep balance but there are so many factors that seem important. So we have to set aside some quality requirements. In this process, we had to clarify the main goal of our system again.

2) Key attributes: Key attributes are properties of important requirements. For describing the important attributes in detail, our group decided to use Planguage. By using Planguage, we link attributes with quality requirements and stakeholder groups. At the same time, the standard of attributes becomes visualized.

Using Planguage brings us some benefits, one of those is that it gives us a clear view of core attributes. For instance, we all know that security is a very important attribute, but it sounds like very generic and lacks abstraction. What Planguage do is to break it down to our system. So by using Planguage, security has a more detailed meaning which is to ensure that the system does not have any security issues. Other key attributes such as usability, availability and response time are also using Planguage to make it clear.

3) 100-test and ranking: The 100-dollar test is a very straightforward prioritization technique where each stakeholder is given 100 imaginary units (money, hours, etc.) to distribute between the requirements. During the 100-test, we found out the most important requirement needs for developers, students, and home owners.

The reason why we used the $100-test technique is that this technique enables us to maximize stakeholder benefits and we can clear see the relationship between the which requirements been liked most and different stakeholders.

Another useful technique is ranking, When using the ranking technique, we get to know detailed information of per stakeholders. We choose this technique as it is very easy to get started with, allows us to combine different stakeholder views, and finally to complement more in-depth techniques such as the 100-test.

C. Specification

The specifications in a report offers a better understanding of the actualization of previously discovered requirements for a software system. The methods described in this section allows the for the problem (real world) domain to meet the solution (computer) domain.

1) Context diagram: Context diagrams are a very useful tool generally used to illustrate to the reader how the system interacts with external entities and their relationship in regards to its data flow. By visualizing the system as one single high-level process, and avoiding to overflow the reader with various technical details, the design of a system, along with its scope and boundaries, can easily be communicated to a wider audience, such as stakeholders, data analysts, etc.

If any negative aspects were to be mentioned, it would be that they are always vulnerable to errors and limited in term of how much knowledge that can be gained from the diagram alone.

2) Task description: Task descriptions is a way of specifying user requirements that describes what the user and system do to fulfill a necessary function. As they are focusing exclusively on the user's understanding of the domain, no thought had to be put into the product design.

We found task descriptions to be an easy way of describing user requirements in a way that would be easy to verify. Grouping the tasks under work areas also added an increased understanding of the interconnected parts of the system.

3) Stakeholder map and analysis: Through various elicitation techniques, an overview of the various primary and secondary stakeholder was established. In order to give a visual representation of which stakeholders influence the project and visually display how they all interact with each other, a stakeholder map was created.

This tool allows the project team to separate internal and external stakeholders to help prioritize and finding the right approach to managing them. Ultimately, based on the prioritization made, this map can allow to easily create an analysis chart that visually represents the stakeholders power and interest in a system.

The stakeholder analysis chart is a rating of stakeholders based on factors such as financial and emotional interest, but also how much influence one has over the system or the system has over them. When a rating has been reached, the shareholders can be categorized into four different categories, each corresponding to a quadrant in the Cartesian chart. These categories are shareholders we want to: Keep satisfied (high power/low interest), manage closely (high power/high interest), monitor (low power/low interest), and keep informed (low power/high interest), respectively. It is important to note that the existence of the project heavily relies on the manage closely category and the development team must ensure their

involvement.

4) Data model: We use an Entity-Relationship (ER) diagram to display the relationship between the entities and to give a high-level overview of how data is recorded by the system. Each relationship in the ER diagram is accompanied by their cardinality which specifies if the entities are necessary, optional or can be instantiated more than once during their lifetime in the system. These relationships with cardinalities are:

- One-to-One Relationship. For example, each accommodation can only have one related contract, and vice-versa.
- One-to-Many Relationship. For example, each Student can apply to many accommodations. They can also apply to 0 accommodations; they can use the system without applying at all. Similarly a Home Owner can post none or many accommodations.
- Many-to-Many Relationship. Multiple students can be interested in multiple Accommodations, This is a composite relationship, as its not directly evident from viewing the ER diagram.

ER diagrams are used as data models for designing relational databases. The diagram made here thus can be used in future work during technical implementation.

5) Data Dictionary: Each entity described by the ER diagram has its own accompanying attributes, which are completely listed in the data dictionary. The supplier can use the data dictionary to understand what data is recorded in the system, how the data is used and which task produces the data. Examples are provided for each data wherever it is relevant for clarity.

III. LEARNING OUTCOMES

1) Elicitation: In software systems development, elicitation is one of the very beginning stages that is used to collect requirements from different sources and with techniques to make the system more compatible and meet user requirements.

In our discussion, we learned how to establish a structure and process for gathering information through different techniques. Initially, we chose brainstorming and hall of fame creativity methods to collect requirements since as students, we are also the primary users for the proposed system.

Creativity techniques such as brainstorming, hall of fame, and creativity triggers are used to conduct a successful eliciting session very early in the project. We were able to identify and collect requirements which is a building block for the development of the system. We made a lot of fruitful discussions, through that we identified problems and met the requirements to avoid unwanted and/or unnecessary assumptions made by the developers. During the hall of fame workshop, our group came up with their famous person and what question should be asked and how we can apply our creative thinking to get the best out of them in a harmonious way. This would help us to transfer ideas, knowledge, experience, and integrate them into our proposed system. Although a full incubation period could not be used, mostly due to time constraints, smaller periods were used where we revisited our original ideas and verify.

The core of our elicitation techniques, however, were the document analysis and the interviews. Since popular housing web pages such as Airbnb and Couchsurfing already exists, many requirements could easily be elicitated by simply analyzing the competitors. This is largely due to the fact that these successful pages have a proof of concept that we can easily mimic or improve on. What distinguishes our system from the rest are the constraints placed. These constraints are: (1) the system has a main focus on housing for students, and (2) location of the system is limited to Gothenburg. Thus, it is important that we elicitate requirements from a user group that is a heavy minority on those previously mentioned housing pages. Interviews were therefore a great way to overcome this and elicitate as many requirements as possible from the very few volunteers that were available, a side effect of elicitating requirements during a pandemic.

The covid-19 virus outbreak impaired our ability to hold physical interviews and thus, by following regulations placed on our group, we had to conduct our interviews remotely. We believe this affected us very negatively as the selection for our sample size was not only very limited, but also subjected to selection bias. To recruit volunteers we had to reach out in various student and housing groups on Facebook, as well as through friends and family. Not only could this potentially misrepresent data and end up with us prioritizing our requirements wrong, but opinions of many stakeholders, especially older home owners who might not frequently use social media, are completely lost. With that being said, we still felt that the data gained through these interviews were irreplaceable and served as a foundation to many decisions throughout the project.

In total, we conducted interviews with two home owners and four students, all remotely. Through interviewing the homeowners, we were surprised to see that some of the user problems that were previously ignored or unconsidered, were very important to our stakeholders. This forced to take new positions and reconsider who our stakeholders really are. One of those aforementioned problems involve legal problems; home owners especially pay very much attention to this, because once something goes wrong, it will be troublesome in terms of both money and time. Which does not lie in the best interest of someone who wants to rent out an accommodation.

Additionally, we found that a reliable platform where you can find people they can trust would be highly valued by both students and home owners over the cost.

Finally, we found that the elicitation method plays a great role in any software development. We can easily track user requirements and easily be able to understand what the system will do and what problems should be solved to meet these requirements.

2) Specification: The specification's primary goal is to give an overview of the entire system and a description of the included requirements. To know what methods to use and how,

we had to learn about which were most suitable in our case and how to apply them.

As the specification changed as the project progressed, we also learned how it could be used as a tool for future changes to the included requirements.

3) Choice of template: During the project, our group decided to change from Lauesen's 2002 template used in the first release to the course template used in the second release in the requirements specification report. The thought process behind this decision was that the Lauesen template, although detailed and well adjusted to real world application with an actual supplier portion and sections that allow you to go into great detail about various system requirements, the course template was better adjusted towards what was required in this project.

The course template, with a few adjustments, could easily be represented within the Lauesen template, however the course template simplified the project work and made it seem very straight forward. Additionally, deviations from the original course template were much easier to make due to its simplistic nature, which is why we ultimately decided to switch for the second release.

4) Group dynamics: Group meetings were held frequently during weekdays with the timing communicated and organized in advance. All meetings were held remotely with project members attending from different corners of the world. The meetings were well structured and tasks were properly divided among the group members. The group always supported each other and strove to meet deadlines throughout the duration of the project.

Initially, there were problems with coordination; some of the group members were connecting from different time-zones. This was resolved by allocating a specific time-frame from the free time available specifically for the project. All the group members came from different backgrounds and were versatile in their methodology, but all approached the project objectively with the singular aim of making a great system. This approach enabled the team to focus on completing the project and allocating tasks.

The team got to know each other on a personal level by communicating outside of the project allocated time. As all meetings were online, there were multiple platforms of communication enabled in case a member wasn't reachable on a singular platform.

IV. CONCLUSION

Requirements engineering plays a great role in the development of software products. It is a very intensive and time-consuming process, but it provides many benefits. We have learned about different methods and techniques used to gather information, ideas, requirements, and collaborate them with our creative thinking to produce very constructing requirements that must be set as the preeminent stage of any software product. The elicitation approach is used to understand the stakeholders, the scope, and the participant of the product. We would apply the same method as we have done it before but we will add more techniques to gather information and to have the best out of them.