Requirements specification for

Student Accommodation Matcher System (SAM)

Customer
Group10 AB

The delivery comprises

Software, operation, and maintenance for an accommodation system

Change log

2019-09-22 Release 1

2019-10-07 Release 2

- Changed the structure of the document:
    - The document is still based on the Requirements Template SL-07, but the chapters after chapter C are not named or used in the same way as in the template. The chapters are freely transformed into a structure that is more suitable for this project.
- Added customers' names and emails on front page
- Added How to read this document
- Chapter A:
    - Moved the stakeholder map from the appendix to chapter A. Added a description and changed the areas of the map
- Chapter B:
    - B2 updated. Added code column with identifiers
    - B3 updated. Added code column with identifiers
    - B4 updated. Deleted B.4.17 since it was not a functional requirement
    - B5 updated. Quality grid added.
    - B6 updated. Updated identifiers to current structure. Made description more clear by adding "...alloted to the suppliers"
    - B7 updated. Updated identifiers to current structure. Added explanation of highlighted rows.
    - B8 updated. Updated identifiers to current structure. Added explanation of highlighted rows. Updated priority (mainly for cells that had impact = 0 in release 1)
- Chapter C:
    - Added reference to chapter E for more information regarding data to record.
    - Work area 1 updated:
        - Changed environment for work area 1
        - C.1 updated. Marked variant tasks.
    - Work area 2 updated:
        - Added description, user profile and environment related to the work area.
        - C.2 updated. Added information and changed previous information. Changed from tenants to landlords under frequency. Changed description of subtask C.2.3
        - C.3 updated. Splitted subtask C.3.1 into two variants.
        - Added new task C.12 - pay for publishment of accommodation
        - Added new task C.13 - pay for upgrade of listing
    - Work area 3 updated:
        - Added description, user profile and environment related to the work area.
        - C.4 updated: added information and deleted the user field.
        - C.5 updated. Made changes for task C5 and added information.

        - C.6 updated. Added more clear description. Changed description of subtask C.6.2
        - C.11 updated. Added information. Changed description of start and end. Changed description of subtask C.11.1
    - Work area 4 updated:
        - Added user profile and environment related to the work area.
        - C7 updated. Added reference to C.4.
        - C8 updated. Deleted start and end.
    - Work area 5 updated:
        - Added description, user profile and environment related to the work area.
        - C.10 updated. Added information. Changed description of subtask C.10.1
    - Added new work area; work area 5 - advertisement.
    - Added new task C.14 - publish and pay for advertisement
- Added chapter D - Core functional requirements
- Added chapter E - Data to record
- Added chapter F - Other functional requirements
- Added chapter G - Non-functional requirements
- Added chapter H - Future extensions
- Added chapter I - Mockup
- Added glossary to the appendix
- Abbreviated the name of the system to “SAM” and made changes throughout the document replacing “system” with “SAM ”.
- Changed C.9 heading from”Update/Change profile settings” to “Manage profile settings”

2019-10-24 Release 3:

- Added references
- Made all sections and tables consistent by changing their IDs. Business goals Identifiers from “BG” to “B.2”
- Chapter A:
    - Updated the context diagram. Deleted system owner and supplier from the diagram. Changed the arrows. Changed the shape for the external systems.
    - Changed text in the first paragraph of A1.
- Chapter B:
    - B2 - business goals
        - Changed BG1.
        - Changed BG2.
        - Changed BG3.
        - Changed BG4
        - Changed BG5.
        - Changed BG6.
        - Changed BG7.
        - Deleted BG8.
        - Deleted the column related requirements

    - B3 - Early proof of concept:
        - Removed EPC1 “Efficient integration of external system”
        - Added EPC1 - “Validation of student status”
        - Added EPC2 - “Display search results according to preferences”
        - Changed identifier for EPC2 - “Posting an accommodation listing” to EPC3 - “Publish an accommodation listing.
        - Added EPC4 - “Apply for accommodation”.
    - Deleted B4 - “Functional requirements”:
    - Deleted B5 - Non-functional requirements
    - Moved quality grid from chapter B.5. to chapter G.
- Chapter C:
    - Added descriptions of how a variant is described.
- Chapter E:
    - Added “This is not a technical description of the data to record.” in chapter E.
- Chapter G:
    - Corrected typos.
    - Added G.6.4. Validate corporate identity number.
- Chapter F:
    - Removed Figure describing Integration with payments system.
- Chapter I:
    - Updated Mockup.
- Added chapter J - Inter-dependencies.

Table of contents

A. Background

A.1. Background and Vision

There is a demand for accommodation in Gothenburg, especially for students it is difficult to find one 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. International students have to look for accommodation in several places where other students already have been registered and collected queue points for a long time. Students want a system where they can find an accommodation as soon as they have been admitted to the university, without the need of being in a queue for a long time before beginning their studies.

Moreover, there are also agencies that require people to pay for the time they are waiting in the queue. Currently, there are several different systems where people can search for an accommodation and each system has different rules, queues and ways to apply.

The SAM-system aims to solve this issue in a way that all students have the possibility to get an accommodation on a first come – first served basis. SAM aims to connect landlords and students with specific needs efficiently. It also aims to decrease waiting time and increase the possibility for students to find an accommodation. SAM streamlines the contract signing process and rent payment and enables direct communication between the landlord and the student.

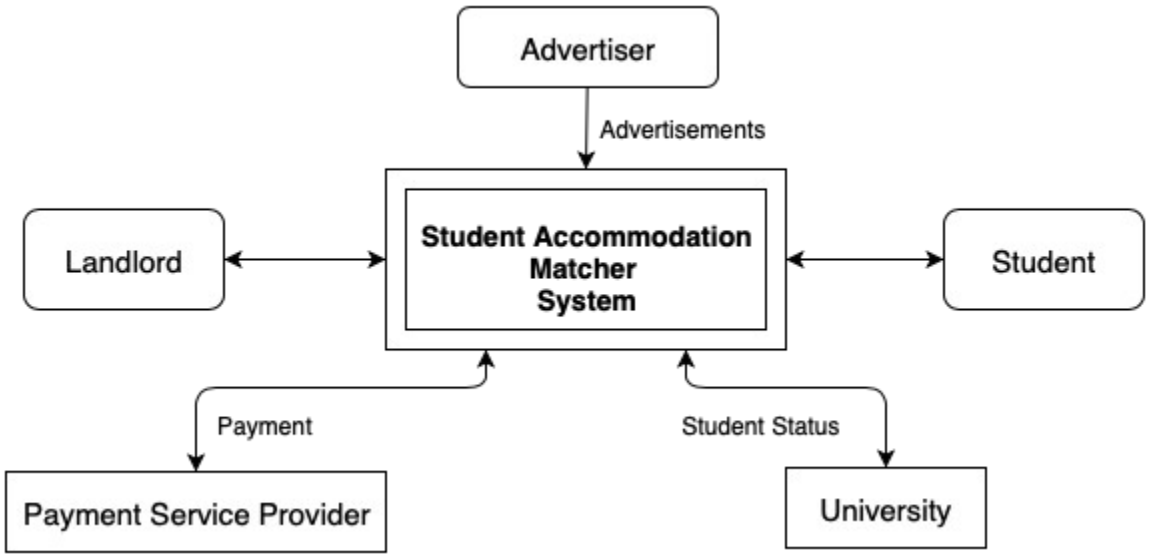

Fig 1. *Context Diagram*

The context diagram in Figure 1 above shows how various stakeholders and external systems interact with SAM.

The system, SAM, is depicted with a double-lined rectangle as it is the final product to be delivered. The arrows from the students and the landlords to the system indicate that they interact with the system to apply and provide accommodation respectively. These two stakeholders are matched with each other according to their preferences. SAM acts as a contract mediator between the student and the landlord once the contract is signed. Integration of SAM with the university ensures that only admitted students apply for accommodation. During the period of contract between the student and the landlord, SAM periodically checks the academic status of the student from the university, to validate their enrollment to a minimum number of credits. Although all payments are made through the SAM system, an integration with an external payment platform ensures secure and hassle free transactions, independent of the core functionality of the system. Advertisers submit advertisements that they want to be displayed in the SAM application through a form provided for the matter.

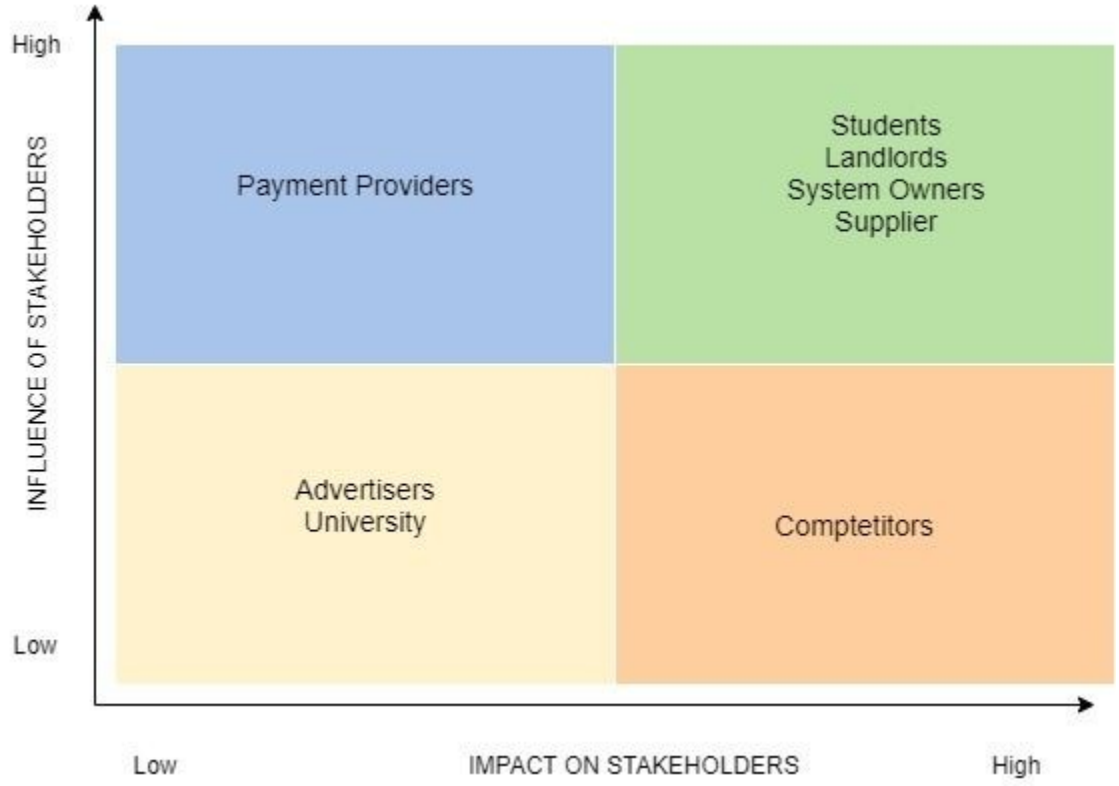

Fig 2. *Stakeholder Map*

The stakeholder map in Figure 2 shows the level of influence and impact that the SAM system has on the various stakeholders. The stakeholders, placed high in the graph, are highly influenced and impacted by

the system, while stakeholders placed below in the graph has less influence and are less impacted with the system.

Table 1 below describes how each stakeholder is impacted by SAM.

Table 1. *Stakeholder Impact and Influence on the SAM system and vice versa*

Stakeholder	**Influence of stakeholders**	**Impact on stakeholders**
Payment providers	The stakeholders have very less or no influence on the payment providers. The only way they will be affected is that they will be paid to be integrated with the system.	The payment provider have high impact on stakeholders. They provide the service of a payment platform through which all the stakeholders and the system will perform transactions.
Students	Students are the stakeholders who are affected the most by the stakeholders. All other stakeholders interact with the system to provide accommodation for the students.	Students are an important part of the system as they are the ones who the system is built for. They search for the accommodation which the landlords publish, and then they apply for them. Once an agreement is made, they sign a contract to become a tenant. They also provide reviews for accommodations to give the system owners and landlords feedback on the listings.
Landlords	The system provides the landlords with tenants and manages the contract between the tenants and the landlord. The payment platform integrated with the system aid in payment of rent and other transactions.	The landlords can be either a private owner or an agency. They are the providers of accommodation. The landlord decides on who the tenant should be; this way they impact the students. The number of listing and description of an accommodation published by the landlord affect the number of students willing to use the SAM system.
System owner	The system owner is affected by all the stakeholders interacting with the system. The stakeholders affect the market value of the system which in turn impacts the system owner.	The system owner sets the requirements and specifications for the system and also validates them. The system and its components decided by the system owner affects the users of the system directly.

Supplier	The supplier is mainly affected by the system owner. All requirements stated by the system owner is to be implemented and maintained by the supplier.	The supplier ensures the system is up and running. Any fault that occurs in the system is handled by the supplier. This way the supplier affects all the stakeholders interacting with the system.
Advertisers	Advertisers are not concerned or affected by any other stakeholders except the system owner who decides to have the advertisement or not.	The advertisers have very less impact on the stakeholders as the system runs independent of them. None of the stakeholders are dependant on the advertisers to use the system.
Universities	The university is mainly a provider to the system and is not affected by the system or any of the stakeholders.	The university validates a student's enrollment to it and also provides information on the number of credits passed by the student. This functionality does not affect the tasks performed by the system which are mainly applying for and providing accommodation.
Competitors	The competitors are greatly affected by the system as the system strives to become the top platform in the market without having the need for students to use multiple platforms. If the competitors are affected by the system, they are affected by the stakeholders that impact the system too.	None of the stakeholders are impacted directly by the competitors. Although the system owner may learn from the competitors to improve the system, the system is not influenced greatly by them.

B. High Level Demand

This chapter is about how to meet customer's business goals through the set of the requirements. The chapter also describes possible mitigation of high-risk requirement.

B.1. Flows

The system shall support two kinds of flow: renting accommodation and providing accommodation. The first one is from a student perspective and the second one from a landlord perspective. In the tables below, column 1 and 2 shows the steps in the flow and column 3 shows the related tasks and subtasks for each step in the flow. See chapter C for details.

B.1.1. Renting accommodation

The system shall support the flow of renting accommodation, which is depicted as shown in Table 2 below.

Table 2. *Flow for renting accommodation, a student perspective*

ID	Steps in renting accommodation	Tasks
B.1.1.1	Create account	C.7
B.1.1.2	Check whether admitted or not	C.7
B.1.1.3	Search an accommodation	C.1
B.1.1.4	Apply for accommodation	C.1
B.1.1.5	Accept or reject the offer	C.1
B.1.1.6	Sign the contract	C.1
B.1.1.7	Pay the rent	C.5
B.1.1.8	Quit contract	C.6

B.1.2. Providing accommodation

The system shall support the flow of providing accommodation, which is depicted in Table 3 shown below.

*T*able 3*. Flow for providing accommodation, a landlord perspective*

ID	Steps in providing accommodation	Tasks
B.1.2.1	Create account	C.7
B.1.2.2	Authenticate owner existence	C.7
B.1.2.3	Publish an accommodation	C.2
B.1.2.4	Offer accommodation	C.3
B.1.2.5	Sign the contract	C.3
B.1.2.6	Quit the contract	C.6

B.2. Business goals

Business goals are defined as the goals that are meant to meet the expected outcome from the system. They have to be pointed out at the initial stage of planning the system requirement. It is always better to deliver some of them in early proof of concept. The customer expects the following results that system could contribute to goals.

Table 4. *Business goals*

ID	Goals for the new system	Solution vision
B.2.1	Encourage and increase the number of paying users	System usage threshold will be lowered by allowing visitors to view the accommodation without having to actually register.
B.2.2	Increase the profit per listing	Offer upgrade for landlords on individual accommodations to increase a listing's placement in the search list, display the listing in the carousel.
B.2.3	Avoid paperwork	Automatic generation of contract agreement between two users helps reduce direct human involvement.
B.2.4	Avoid the need to develop new features from scratch	Integrate system with other external systems.
B.2.5	Provide a feeling of assurance to potential tenants	Review of the accommodation by previous tenants will give potential tenants a better overview of the accommodation.
B.2.6	No upfront costs for the students	Allowing the students to be able to view and apply for the accommodation without the need of an upfront payment and only make the payments once they have signed a contract, satisfies the SAM system vision.
B.2.7	Reduce response time for application acceptance / rejection	The system should notify landlords to respond to the applicants in queue. No response from landlords, removes the accommodation from the listing, notifying the students that the listing is no longer available for application.

B.3. Early proof of concept

Requirements described here are considered to be high level risks (refer to table 5). If the fundamental features are not completed in the given time that means the supplier has not attain customer's expectations. If the expectations are not met then Customer can terminate the contract. It is the core duty of the supplier to let the customer know when the early prototype will be delivered. Once, the prototype will be completed, It has to be accepted by the customer. The contract can be terminated by either party if the early proofs fail.

Table 5. *Early proof of concept*

Code	**Areas where an early proof of concept is required**	**Example of proof**
B.3.1	Validate student status	A university should return the correct student status when SAM system requests for one.
B.3.2	Display search results according to preferences	When a user inputs a query with filters, it should display appropriate results.
B.3.3	Publish an accommodation listing	A user should be able to input accommodation details and submit it for publication. The accommodation should appear in the listing.
B.3.4	Apply for an accommodation	A user should be able to view published listings and apply for them. The application should be reflected in the system.

B.4. Minimum requirements

The customer gives the minimum scores to each area that needs to be fulfilled by the supplier.

Scores: The customer gives each proposal scores for the requirement areas shown in Table 9 below. To provide better overview, the tables have space for several proposals alloted to the suppliers (columns A, B and C). The scores are represented by the following numbers :

- -2 : Cannot be supported by the system
- -1 : Inappropriate
- 0 : can be supported by the system
- 1 : Appropriate
- 2 : Efficient.

Minimum score: The requirement areas prove to be useful only when they satisfy a score above a certain minimum score.

Minimum requirements: All the requirement areas with minimum scores below must be satisfied by the system.

Table 6. *Minimum requirements*

Requirement area	**Reason**	**Minimum score**	**Score**		
			A	**B**	**C**
C.1. – C.2. Apply/Publish	The core of the system. Provides searching and publishing accommodation.	2			
C.3. – C.6. Offers/Contract	Ensuring that only applicants are students, contract for an accommodation is signed, or terminated.	1			
C.7. – C.10 Account management	Provides security for the system's users.	1			
E. Data		0			
F.1. System generated events		1			
F.2. Integration with other external systems	Increase quality of the system.	1			

G.1.Non-functional requirements (Security)	Importance of protecting data.	1			
G.2. – G.7. Non-functional requirements	Increases user experience .	0			
H. Future extensions	Potentially will increase profit value.	0			

B.5. Selection Criteria: MoSCoW

Priority of requirements are chosen based on MoSCoW. MoSCoW categorizes requirements according to the following order [2]:

Must - Requirements must be included in successful product delivery.
Should - Requirement which should be included in successful product delivery. In case of insufficient financial support, the core of the system can work without these requirements.
Could - Requirements which could be included if they could be financially supported.
Won't - Requirements which won't be included in successful product delivery. For the first product release, this category won't be included in this document.

Scores: Each requirement importance is decided according to the opinion of each stakeholder. Based on which importance is dominant for each requirement, this one is set as final importance.

Table 7. *MoSCoW requirement prioritization*

Requirement	**Rama**	**Pernilla**	**Teklit**	**Asad**	**Shameer**	**Sanja**	**Total**
C.1. – C.2. Apply/Publish	M	M	M	M	M	M	M
C.3. – C.6. Offers/Contract	M	M	M	M	M	M	M
C.7. – C.10. Account management	M	S	S	M	S	S	S
E. Data	M	S	S	M	S	S	S
F.1. System generated events	S	C	S	C	S	C	C
F.2. Integration with other systems	C	M	M	M	M	M	M

G.1. Non-functional requirements (Security)	S	M	M	M	M	M	M
G.2 - G.7. Non-functional requirements	M	S	S	S	M	S	S
H. Future extensions	C	C	C	S	C	C	C

Note - The highlighted rows correspond to the core functional requirements of the system as all o are a MUST-have requirements.

B.6. Selection criteria: Priority Scorecard

Features are categorised according to different criteria. Chosen categories are [3]:

User Experience: User experience when using the platform.
Revenue: Features which increases revenue.
Operation Efficiency: Features that impact profitability. For example: Rent payment and electronic contract signature increases profitability in that it reduces operational costs.
Security: This is about protecting users private data.
Maintenance: Outsource specific features to external partners such as payment system. Thereby reducing the need for internal maintenance.

Scores: Each requirement is assigned to a score from 0-100 for each category. 100 represents high impact on that category. 0 means no impact. Total score is calculated by multiplying the score by the weight.

Note - The highlighted rows show the top five requirements with highest priority. These requirements must be implemented as a part of the system.

Table 8. *Scorecard requirements prioritization*

Category	**User experience**	**Revenue**	**Operation Efficiency**	**Security**	**Maintenance**	**Total**
Weight	15%	30%	20%	25%	10%	100%
Requirements	**Score**					**Priority**
C.1. – C.2. Apply/Publish	70	70	20	20	30	43.5
C.3. – C.6. Offers/Contract	60	50	50	20	30	42

C.7. – C.10. Account management	5	30	20	100	10	39.75
D. Data	10	60	40	67	10	37.25
F.1. System generated events	40	14	20	10	20	18.70
F.2. Integration with other systems	40	40	30	40	10	35.00
G.1. Non-functional requirements (Security)	40	20	50	100	30	50
G.2. - G.7. Non-functional requirements	30	31	20	30	10	26.30
H. Future extensions	20	40	30	20	10	27.00

C. Tasks to support

In this chapter tasks that must be supported when users interact with the system are described. Even though all of the tasks are numbered, there are cases when some of the tasks are optional and when it is not necessary to perform them in the same order as they appear in the document. One task can contain several subtasks and it is possible to do one subtask several times during a session. For some of the sub-tasks there are alternative ways that they can be performed. These are called variants and are identified with letters such as a,b, c, etc.

A work area describes the tasks that need to be supported for a particular user and in a particular environment where the task is carried out. The tasks within each work area are related to each other in some way and the purpose of the grouping is to provide a better overview of the main parts of the system.

Data related to the tasks in this chapter is further described in chapter E.

Work area 1. Application for accommodation

This work area contains the various tasks involved in applying for an accommodation, from looking for an accommodation to moving out from one.

User profile: Student.
Environment: Smartphone app and web system

C.1. Apply for accommodation

This task creates the student's application for accommodation.

Users: Student.
Start: When a student starts to search for accommodation.
End: When a student signs the contract.
Frequency: Maximum of 5 students per single application and a limit of 5 applications per student.

Table 9. *Subtasks for C.1.*

ID	Subtask
C.1.1a	Search for accommodation without preferences
C.1.1b	Search for accommodation with chosen preferences
C.1.2	Display information about selected accommodation.
C.1.3	Apply for the selected accommodation
C.1.4a	Accept an offer
C.1.4b	Reject an offer
C.1.5	Sign the contract

Work area 2. Providing accommodation

This work area contains the various tasks involved in providing accommodation.
User profile: Landlord, student.
Environment: Smartphone app and web system

C.2. Publish accommodation

This task performs actions for landlords who want to put up a post of the accommodation(s) on the system for students to view. To ensure that only genuine landlords will offer accommodation, landlords will have to pay a fee when publishing their posts (see C.12). Once the students appear in the applicants queue, the landlord decides regarding acceptance or rejection of an applicant, as described in task C.3. The accommodation will not be published unless the landlord provides recently taken images of the accommodation. It is possible to save a post and publish it at a later stage.

Users: Landlord.
Start: A landlord wants to publish an accommodation.
End: Published accommodation accepted.
Frequency: 100,000 landlords can create an account and post accommodation.

Table 10. *Subtasks for C.2*

ID	Subtask
C.2.1	Enter detailed information about the accommodation.
C.2.2	Add images (mandatory).
C.2.3	Describe restrictions on tenant profile (e.g. non-smoking).
C.2.4	Pay for publishing accommodation
C.2.5a	Publish the post.
C.2.5b	Save the post for later.

C.3. Offer accommodation

This task provides an accommodation offer to eligible applicants. If the first student from the queue does not fulfil landlord's requirement, the second student in the queue will be offered to the landlord. If both parties agree on the terms and conditions, the digital contract is sent to them for finalizing agreement.

Users: Student, landlord.
Start: An accommodation has an applicant.
End: Sign the contract.
Frequency: 1,000 offers per day.

Table 11. *Subtasks for C.3*

ID	Subtask
C.3.1a	Accept applicants in queue.
C.3.1b	Reject applicants in queue
C.3.2	Sign the contract.

C.12. Pay for publication of accommodation

This task involves the handling of the payment for publishing accommodation. To ensure only genuine landlords, the system requires the landlord to pay a fee for publishing an accommodation. When the landlord has paid the fee, the system publishes the accommodation.

Users: Landlord.
Start: Landlord has completed the description of the accommodation and wants to publish it.
End: The landlord's accommodation is published.
Frequency: Once for every new published listing.

Table 12. *Subtasks for C.12*

ID	Subtask
C.12.1	Display cost and payment details.
C.12.2	Pay the fee.
C.12.3	Publish accommodation.

C.13. Pay for upgrade of listing

This task involves the handling of the payment for an upgrade of a listing. If the landlord wants a better chance of getting views for a specific listed accommodation, it is possible to pay for an upgrade of the listing. The upgrade will show the accommodation in a carousel and push the listing to a higher position on the accommodation list.

Users: Landlord.
Start: The landlord wants to upgrade a listing in order to get more views.
End: The landlord's accommodation is upgraded.
Frequency: Once for every new upgrade.

Table 13. *Subtasks for C.13*

ID	Subtask
C.13.1	Show cost and payment details.
C.13.2	Pay the fee.
C.13.3	Add listing to carousel.
C.13.4	Push listing to the top of the listings.

Work area 3. Contract maintenance

This work area contains the various tasks involved in contract maintenance.
User profile: Landlords, students.
Environment: Smartphone app and web system

C.4. Periodic check of student's academic status

This task checks whether the student is still an active student or not. Depending on the student's academic status, the system decides whether to continue or to terminate the contract. This task is executed by the system.

Start: Check student's academic status.
End: Continue or terminate contract.
Frequency: End of every semester.

Table 14. *Subtasks for C.4*

ID	Subtask
C.4.1	Check the student's academic status.
C.4.2a	Update the status as active in the system.
C.4.2b	Initiate contract termination.

C.5. Pay the rent

This task involves the handling of the payment for the rent. The landlord specifies required amount, the system adds a commision of xx percent of the rent and then sends the request to the student. The student pays the system and the system pays the landlord. When the student has paid the rent, the system generates a receipt containing information about the payment. If the student has not paid the rent on time, the system will still pay the landlord on time. The system will generate a reminder notifying the student that if the student will not pay the rent in xx days, there will be additional costs.

Users: Student, landlord.
Start: Landlord initiate the monthly rent process.

End The receipt of the payment.
Frequency Once every month.

Table 15. *Subtasks for C.5*

ID	Subtask
C.5.1	Require the payment of the rent.
C.5.2	Pay the rent.
C.5.3	Generate receipt.

C.6. Quit contract

This task terminates the contract.
In case of the end of a contract period, the system automatically initiates the termination of the contract. It could also happen because a student is no longer fulfilling the criteria for renting the accommodation. Furthermore, a breach of contract by either a landlord or a student could also lead to a termination of the contract. When termination process is initiated, the student has xx days to move out.

Users Student, Landlord.
Start Initiate the termination process.
End Termination of the contract.

Table 16. *Subtasks for C.6*

ID	Subtask
C.6.1	Initiate closing contract.
C.6.2	Acknowledge the end of the contract.

C.11. Accommodation fault reporting

This task provides the student with the possibility to report and describe any fault related to the accommodation. The student creates a ticket describing a fault that requires the landlord's attention. When the landlord views this ticket, fixes the fault or gives a feedback on the further process, he closes the ticket.

Users Landlord, student.
Start Student reports a fault by creating a ticket
End Landlord closes the ticket

Table 17. *Subtasks for C.11*

ID	Subtask
C.11.1	Create a ticket.
C.11.2	Close a ticket.

Work area 4: Account management

This work area allows the users to register, login and manage their account.

User profile: Landlord, student.
Environment: Smartphone app and web system

C.7. Register

This task creates an account for an admitted student or a landlord. A student is authenticated by checking the student's academic status (see C.4).

Users: Landlord, student.
Start: Enter the required information.
End: Verify registration.

Table 18. *Subtasks for C.7*

ID	Subtask
C.7.1.	Authenticate student
C.7.2	Complete registration
C.7.3	Reject registration

C.8. Login / Logout

This task allows users to login to an existing account, which enables more actions such as applications, publishing, etc.

Users: Landlord, student.

Table 19. *Subtasks for C.8*

ID	Subtask
C.8.1.	Login to the system
C.8.2	Request password reset
C.8.3.	Logout off the system

C.9. Manage profile settings

This task updates/changes a user's profile settings or deletes the account.

Users: Landlord, student.
Start: Navigate to settings.
End Save changes.

Table 20. *Subtasks for C.9*

ID	Subtask
C.9.1	Change personal information
C.9.2	Change notification settings
C.9.3	Delete account

Work area 5: Maintenance and technical support

This work area contains tasks related to maintenance and technical support of the SAM system.

User profile: Landlord, student.
Environment: Smartphone app and web system

C.10. System fault reporting

This task provides students and landlords with the possibility to report any fault related to the system. The student or the landlord creates a ticket describing the fault they encountered while using the system. This ticket is delivered to the system owner who handles it. Despite the status of the fault, the ticket can be closed by the system owner.

Users: Landlord, student.
Start: Raise a complaint ticket to report a fault
End The ticket is closed.

Table 21. *Subtasks for C.10*

ID	Subtask
C.10.1	Create a ticket.
C.10.2	Close a ticket.

Work area 6: Advertisement

This work area contains tasks related to advertisement.

User profile: Advertiser.
Environment: Smartphone app and web system

C.14. Publish and pay for advertisement

This task involves the handling of the payment for publishing an advertisement.
An advertiser can send a request for publishing an advertisement. If the advertisement request is accepted by the system, the system will give the advertiser an offer. If the advertiser accepts the offer, the advertiser has to pay a fee to the system in order to publish the advertisement. When the advertisement has been shown for the specified length given in the request and/or offer, the system deletes the advertisement.

Users: Advertiser.
Start: Advertiser wants to publish an advertisement.
End: Delete advertisement.
Frequency: Once for every new advertisement.

Table 22. *Subtasks for C.14*

ID	Subtask
C.14.1	Send advertisement request.
C.14.2a	Accept request and show offer.
C.14.2b	Reject request.
C.14.3a	Accept offer and pay the fee.
C.14.3b	Reject offer.
C.14.4	Publish advertisement.
C.14.5	Delete advertisement.

D. Core functional requirements

This chapter presents the core functional requirements with their associated detailed information. The chapter provides the supplier with an example of a technical level of detail.

D.1. Application for accommodation

This section describes the detailed logical flow of the applying for accommodation functional requirement. The system shall allow only eligible students to apply for accommodation. This section is related to section C.1. and provides a more technical level of abstraction.

D.1.1. Search accommodation

The system shall be able to allow visitors to search for accommodation using different preferred criteria with or without creating an account. The system shall allow any visitor to search an accommodation regardless of having an account, since visitors may be interested in searching before creating an account. The system shall also provide search with preferences and without preferences option. The search without preferences allows to simply search all accommodations available for rent, whereas search with preference allows the visitor to search an accommodation that the visitor is only interested in. The search with preferences option should allow the visitor to search for an accommodation using various criterias as given in Table 23 below.

Table 23. *Search preferences*

ID	Search preferences	Description
D.1.1.1	Size	The surface area of the accommodation
D.1.1.2	Location	The particular location (address) where the accommodation is found
D.1.1.3	Number of rooms	The number of rooms the accommodation contains
D.1.1.4	Maximum rent	The maximum amount that the visitor can afford to pay
D.1.1.5	Shared accommodation	The student prefers to share the accommodation they are applying to with another tenant.

Regardless of the search options used, the system shall show the visitor's search results with the information in Table 24 below.

Table 24. *Search results*

ID	Information	Description
D.1.1.6	Size	The surface area of the accommodation
D.1.1.7	Number of rooms	The number of rooms within the accommodation
D.1.1.8	Furnished or unfurnished	Whether the accommodation is furnished or unfurnished
D.1.1.9	Shared or unshared accommodation	Whether the accommodation has shared facilities or not
D.1.1.10	Location	The particular location of the accommodation
D.1.1.11	Apartment number	The apartment number of the accommodation
D.1.1.12	Floor number	The floor number
D.1.1.13	Rent	The amount of rent
D.1.1.14	Move in date	The date when the tenant can move in to the accommodation
D.1.1.15	Roommate allowed	Whether allowed to bring a roommate.
D.1.1.16	Existing tenants	The number of existing tenants, if any.
D.1.1.17	Elevator	Whether the apartment has an elevator or not

D.1.2. Apply for accommodation

The system shall be able to ask for login credentials when students want to apply for an accommodation (if they have not signed in yet). While students are applying for accommodation, the system shall notify them that no one can apply for more than five accommodations at the same time. The system shall allow only admitted students to apply for accommodation. When students apply for accommodation, the system shall ask them to agree with the terms of the housing landlord or agency. Application shall be possible from both the detailed information page for a certain accommodation and from the application page. When application is done from the application page, the student select what accommodation the application is for and the system fills out the rest of the details (see Table 25 below).

Table 25. *Application Form Details*

ID	Application form details
D.1.2.1	List of accommodations
D.1.2.2	Area
D.1.2.3	Rent amount
D.1.2.4	Address
D.1.2.5	Floor number
D.1.2.6	Apartment number

D.1.3. Accept or Reject an offer

The system shall allow the student to accept or reject the offered accommodation. This is because the student may not be interested in the offer. So, the student shall have an option to accept or reject the accommodation offered such as shown in Table 26 below.

Table 26. *Accept or Reject Option to an Offer*

ID	Option
D.1.3.1	Accept
D.1.3.2	Reject

D.1.4. Sign Contract

The system shall generate contract form details when the student accepts the offer. The contract form is an important one that binds the tenant and landlord to their agreement. The contract form may contain the details specified in Table 27 below. The contract form generated shall have an option to sign it physically or digitally.

Table 27. *Renting Contract Form Details*

ID	Contract form details
D.1.4.1	Full name of student
D.1.4.2	Full name to the landlord or agency
D.1.4.3	Passport ID or personal number
D.1.4.4	Phone number
D.1.4.5	Email
D.1.4.6	Address
D.1.4.7	Rent
D.1.4.8	Payment options
D.1.4.9	Duration of contract
D.1.4.10	Apartment number
D.1.4.11	Rent amount specification

D.2. Publish accommodation

Publishing listings of accommodations is one of the primary tasks of a landlord. A prerequisite for publishing a listing is that the user must have registered for an account in the system as a landlord.

D.2.1. Subtasks

D.2.1.1. Create a Listing

The steps involved in publishment of accommodation include the following:

Table 28. *Publishing accommodation subtasks*

ID	Steps
D.2.1.1.1	Navigate to “Create Listing” page
D.2.1.1.2	If the landlord wants to create a listing of an accommodation that is already in the system, select that accommodation
D.2.1.1.3	If the landlord wants to create a listing of an accommodation that is not in the system, navigate to “Add Accommodation” page
D.2.1.1.4	Provide details of the accommodation and save it
D.2.1.1.5	Provide details of the particular listing and save it for publishing later or publish it

D.2.1.2. View Applicants

Once the listing has been published, students can apply for the accommodation. Landlords should be able to view the first five applicants for their accommodation in chronological order. They should also see any messages sent to them by the applicants.

D.2.1.3. Accept / Reject Applicant

Prompt response is one of the key characteristics of the system. Landlords should accept or reject an applicant within x days from the receipt of the application. Such acceptance or rejection should be communicated back to the students in a timely fashion.

D.2.1.4. Sign Contract

Once accepting an applicant and the applicant has also accepted the offer, both parties should sign a contract. The system should generate such contract automatically from the data available to the system.

D.2.2. Flowchart

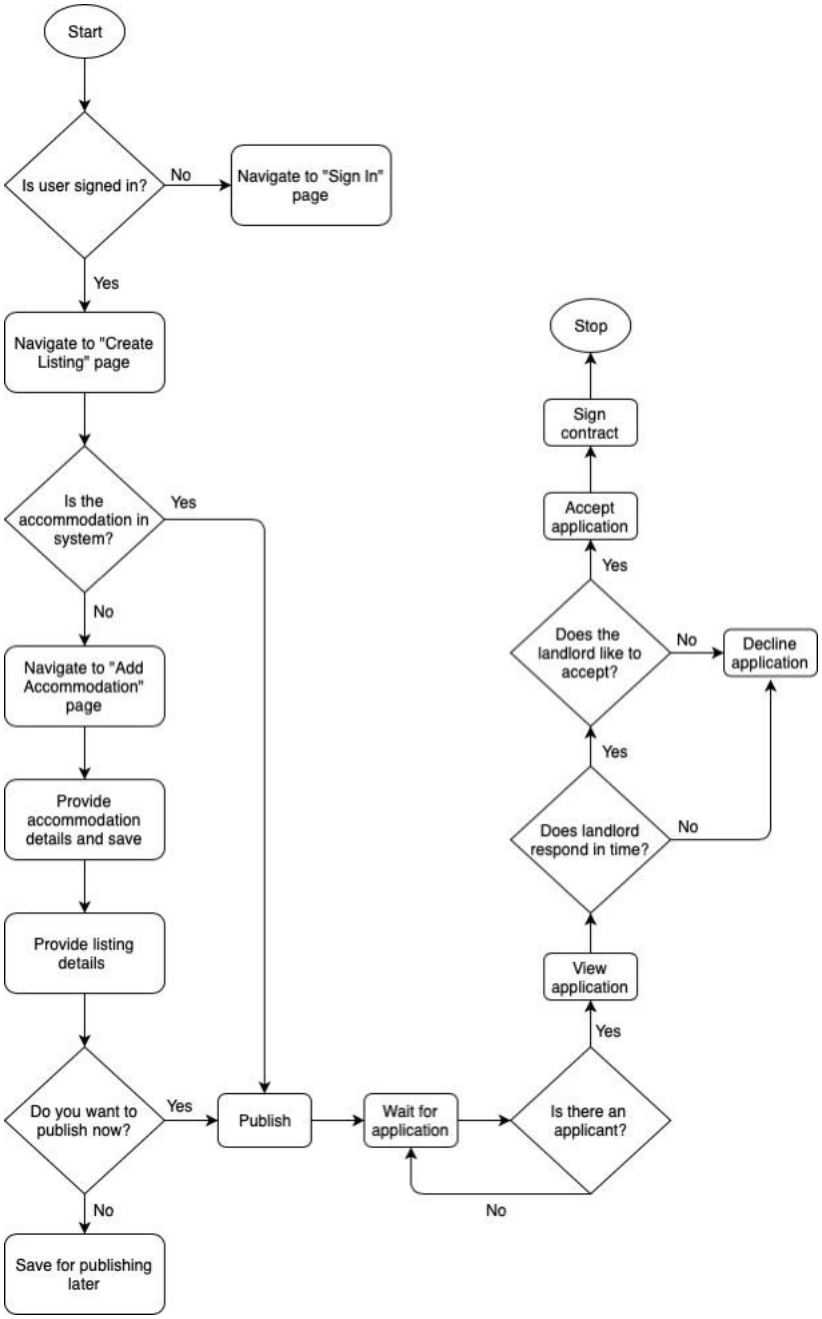

Fig 3. *Example flow chart for publishing accommodation*

D.2.3. Data Tables

D.2.3.1. Accommodation Table

Table 29. *Accommodation table fields*

ID	Field	Type	Other Info	Example
D.2.3.1.1	ID	Integer	Primary key, auto-incremented	1
D.2.3.1.2	Address	String		Exampelgatan 6, Exampelstad 123 45, Sweden
D.2.3.1.3	Size	Float		15.0
D.2.3.1.4	Number of rooms	Integer		2
D.2.3.1.5	Category	String		
D.2.3.1.6	Landlord ID	Integer	Foreign Key from *Landlord* table (not presented in this section)	1

D.2.3.2. Listing Table

Table 30. *Listing table fields*

ID	Field	Type	Other Info	Example
D.2.3.2.1	ID	Integer	Primary key, auto-incremented	1
D.2.3.2.2	Accommodation ID	Integer	Foreign Key	1
D.2.3.2.3	Rent	Float		4500.00
D.2.3.2.4	Application Deadline	Datetime		2019-12-01 12:00:00

D.2.3.3. Facility Table

Table 31. *Facility table fields*

ID	Field	Type	Other Info	Example

D.2.3.3.1	ID	Integer	Primary key, auto-incremented	1
D.2.3.3.2	Facility	String		Attached bathroom

D.2.3.4. Accommodation-Facility Table

Table 32. *Accommodation-facility table fields*

ID	Field	Type	Other Info	Example
D.2.3.4.1	ID	Integer	Primary key, auto-incremented	1
D.2.3.4.2	Accommodation ID	Integer	Foreign Key	1
D.2.3.4.3	Facility ID	Integer	Foreign Key	1

D.2.3.5. Image Table

Table 33. *Image table fields*

ID	Field	Type	Other Info	Example
D.2.3.5.1	ID	Integer	Primary key, auto-incremented	1
D.2.3.5.2	Image Link	String		https://cdn.example.com/..../acc1.jpg
D.2.3.5.3	Accommodation ID	Integer	Foreign Key	1

D.2.3.6. Restriction Table

Table 34. *Restrictions table fields*

ID	Field	Type	Other Info	Example
D.2.3.6.1	ID	Integer	Primary key, auto-incremented	1
D.2.3.6.2	Restriction	String		No smoking

D.2.3.7. Accommodation-Restriction Table

Table 35. *Accommodation - Restrictions table fields*

ID	Field	Type	Other Info	Example
D.2.3.7.1	ID	Integer	Primary key, auto-incremented	1
D.2.3.7.2	Accommodation ID	Integer	Foreign Key	1
D.2.3.7.3	Restriction ID	Integer	Foreign Key	1

D.2.3.8. Application Table

Table 36. *Application table fields*

ID	Field	Type	Other Info	Example
D.2.3.8.1	ID	Integer	Primary key, auto-incremented	1
D.2.3.8.2	Listing ID	Integer	Foreign Key	1
D.2.3.8.3	Student ID	Integer	Foreign Key from *Student* table (not presented in this section)	1
D.2.3.8.4	Application Date	Datetime		2019-10-05 07:52:27
D.2.3.8.5	Status	String		Accepted

D.2.4. Class Diagram

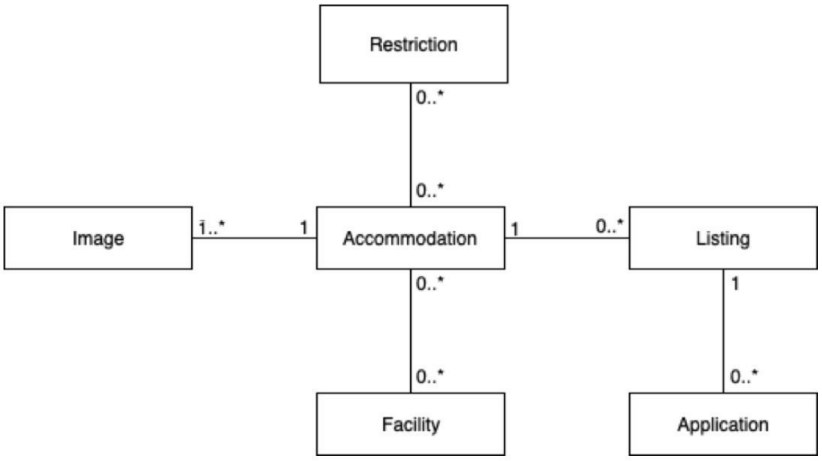

Fig 4. Class diagram for SAM

- An accommodation can have zero or more listings.
- A listing must be an accommodation.
- An accommodation can have one or more images.
- An image must be on an accommodation.
- An accommodation can have zero or more facilities.
- A facility can belong to zero or more accommodations.
- An accommodation can have zero or more restrictions.
- A restriction can be applied on zero or more accommodations.
- A listing can have zero or more applications.
- An application must be for a listing.

E. Data to Record

The SAM system shall record the data described in this chapter. This is not a technical description of the data to record.

E.1. User

The system shall record the data in this section.

A user is a physical or legal person who uses the system for one of the following reasons: searching for accommodation or providing accommodation. A person who has not signed in is referred to as a visitor and must sign in in order to apply for or to provide accommodation. Registered students can apply for accommodation. Landlords and agencies both are referred to as a landlord and must register in order to provide accommodation.

Examples of source and use: C.4, C.7, C.9
Data sources: User data is recorded during registration (C.7) and during updates/changes of profile settings (C.9). Students and landlords provide the system with most of the data, but authorities (external source) will verify if some of the data is correct (such as personal identification number and corporate identity number). The university will verify the student's academic status (C.4).
Data uses: User data is used for identification of users.
Data volumes: Around 200 000 registered users.

E.1.1. User

Table 37. *Data entered by the user that the system will record during C.7 and C.9.*

ID	Field	Description	Example Solution
E.1.1.1	Type of user	Student or landlord.	
E.1.1.2	Name	First name and last name.	
E.1.1.3	Personal identification number (PIN)	PIN is needed in order to verify the identity of users and can be used for exchanging information between external parties such as authorities and third party services.	
E.1.1.4	Date of birth	A person without any PIN can enter date of birth instead.	
E.1.1.5	Gender	Man, woman, non-binary. Not required for agencies.	
E.1.1.6	Address	Complete postal address for current accommodation.	

E.1.1.7	Phone number	The format of the phone number includes the country code.	
E.1.1.8	Email		The system shall verify that the email is correct.
E.1.1.9	Username		All usernames shall be unique.
E.1.1.10	Password		The system shall offer recommendations for choosing a strong password.
E.1.1.11	Profile picture		A user shall have the ability to add a profile picture to his/hers account.

E.1.2. Student

A student shall have all of the data fields stated in Table 37 in E1.1 and the Table 38 below.

Table 38. *Data related to a student.*

ID	Field	Description	Example Solution
E.1.2.1	University	The name of the university a student is admitted to.	
E.1.2.2	Academic status (admission status)	Active or non-active. A student has to be active in order to use the system. An active student is admitted to a university and passes a certain amount of credits each semester.	This data shall be provided by the university integration (see section E.6).
E.1.2.3	Duration of studies	Number of months that the student plans to study.	

E.1.3. Landlord

A landlord can be either a private person or an agency. All landlords should have the same data fields as stated for a user in section E1.1 (see exceptions for an agency in Table 42), and a type as stated in Table 39 below.

Table 39. *Data related to both of the types of a landlord.*

ID	Field	Description	Example Solution
E.1.3.1	Type of landlord	Private or agency.	The system shall record what type of landlord a certain user is.

An agency doesn't have a name or a PIN as a private landlord. An agency has a company name and corporate identity number instead of a name and a PIN. An agency should therefore in addition to the data fields in E.1.1 (excluding gender), have the following data fields as stated in Table 40 below.

Table 40. *Data related to an agency (landlord).*

ID	Field	Description	Example Solution
E.1.3.2	Company name	The agency's registered company name.	
E.1.3.3	Corporate identity number	The agency's corporate identity number is required to verify that it is a registered agency.	The system shall verify that the agency exists and matches the company name entered by the agency.
E.1.3.4	Website	A URL link to the agency's website.	

E.2. Accommodation

The system shall be able to distinguish each accommodation from another (e.g. by providing an accommodation ID) and to save accommodation information for future use (e.g. when publishing the accommodation as a new listing). An accommodation must have a landlord and only registered users shall be able to see who the landlord of an accommodation is.

Examples of source and use: C1, C2, C3, C11
Data source: Accommodation data are recorded during publishment of accommodation (C.2). Applicants and tenants are recorded during apply for accommodation (C.1) and offer accommodation (C.3).
Data user: Accommodation data are used for description of accommodations. When students search for accommodation, they can filter their results based on the data in this section. The data are shown for every listing. The data are used during accommodation fault reporting (C.11).

Data volume: The data shall be recorded for each accommodation and listing. There will be around 50 000 listings each year.

E.2.1. Accommodation information

A landlord has to enter information about the accommodation he/she is providing, and the system shall record this data (see Table 41 below).

Table 41. *Information that a landlord has to provide about the accommodation*

ID	Field	Description	Example Solution
E.2.1.1	Address	Complete postal address (street address, postal code, locality and county)	The system shall verify that the address exists in order to create and publish the accommodation as a listing.
E.2.1.2	Size	The size (in square meters) of the accommodation for rental	
E.2.1.3	Category	Room, apartment or house	
E.2.1.4	Number of rooms	Number of rooms included in the rental agreement for the accommodation.	
E.2.1.5	Maximal number of tenants	In case the accommodation is available for multiple tenants at the same time, maximum number of tenants have to be stated.	
E.2.1.6	Price (incl. rent)	The price includes the rent and any other costs that are related to the accommodation. It shall be stated what is included in the rent and what requires extra payment, e.g. internet, electricity or heating.	
E.2.1.7	Facilities	Description of any facilities related to the accommodation, e.g. laundry and gym.	
E.2.1.8	Restrictions	The landlord shall describe any restrictions related to the accommodation, e.g. if he/she requires the tenant to be smoke-free, animal-free etc.	
E.2.1.9	Inventory	The landlord shall describe if there are other things included in the rent, such as	

		furniture, porcelain or cutlery.	
E.2.1.10	Length of rental agreement	The landlord has to enter for how many months the accommodation is available.	
E.2.1.11	Application deadline	If there is a deadline for application, this shall be stated in the information about the accommodation.	
E.2.1.12	Available from	The landlord has to enter from what date the accommodation is available and ready for a student to move in.	
E.2.1.13	Images	The landlord has to attach at least one image of the accommodation.	
E.2.1.14	Additional information	The landlord shall be able to include additional information in text.	
E.2.1.15	Existing number of tenants	Number of tenants that are already living in the accommodation.	
E.2.1.16	Roommate allowed	The landlord shall enter whether it is allowed to bring a roommate.	

In order to keep track of previous applicants and tenants, the system shall record the data in Table 42 below.

Table 42*. Data that is related to an accommodation but that are not entered by the landlord.*

ID	Field	Description	Example Solution
E.2.1.17	Applicants	All the applicants that have applied for the accommodation. They might have applied for different listings of the accommodation.	The system shall record all the applicants for the accommodation.
E.2.1.18	Tenants	All the tenants that either have lived or are living in the accommodation.	The system shall record all the tenants of the accommodation.

E.2.2. Listing

The system shall record the data in Table 43 below.

A published accommodation is referred to as a listing.
A landlord shall be able to publish an accommodation multiple times without the need of specifying all of the accommodation information in E.2.1 every time. In other words, when the first listing for a certain accommodation is deleted, it shall be possible to create a new listing for the same accommodation.

Table 43. Data related to a listing.

ID	Data fields	Description	Example solution
E.2.2.1	Applicants	Applicants that have applied for the listing.	The system shall record which applicants that have applied for a listing.
E.2.2.2	Queue position of applicant	In case of several simultaneous applicants, a queue position is needed.	The system shall put applicants in a queue and show them in that order for the landlord. A listing shall be hidden for other students when the limit of five simultaneous applicants for that specific listing is reached.

E.3. Contract

A contract shall be generated automatically by the system. The contract shall include the data fields stated in Table 44 below.

Examples of source and use:
Data source: Contract data are recorded during the signing of a contract, which involves apply for accommodation (C.1) and offer accommodation (C.3). The most of the data will be imported from the system but start date, end date, and terms and conditions might be updated or added by the student or the landlord.
Data use: Automatic generation of a digital contract for rental agreement.
Data volume: One contract per listing.

Table 44*. Data related to a contract.*

ID	Field	Description	Example Solution
E.3.1	Landlord	The landlord of the accommodation.	
E.3.2	Student(s)	The student(s) who will be the tenant(s) of the	

		accommodation.	
E.3.3	Accommodation	The accommodation that the contract is generated for.	
E.3.4	Start date	The start date of the contract.	
E.3.5	End date	The end date of the contract.	
E.3.6	Price (incl. rent)		The contract shall include information about the rent and any additional costs.
E.3.7	Terms and conditions	Any restrictions should be stated in the terms and conditions.	The system shall generate a contract containing all terms and conditions for that accommodation. The system shall allow the tenant and the landlord to add additional information.

E.4. Communication

The system shall allow students and landlords to communicate with each other within the system, and thus record the data described in Tale 45 below.

Examples of source and use:
Data source: Recorded during communication between students (tenants) and landlords.
Data use: Communication between students and landlords.
Data volume:

Table 45. *Data related to communication.*

ID	Field	Description	Example solution
E.4.1	Sender		The system shall record who the sender of the message is.
E.4.2	Receiver		The system shall record who the receiver of the message is.
E.4.3	Date & time		The system shall record the date and time for a certain message.
E.4.4	Message		The system shall record the message and offer the ability to include any media in the message.

E.5. Fault report

The system shall record the data described in Table 46 below.

Examples of source and use:
Data source: Recorded during system fault reporting (C.10) and accommodation fault reporting (C.11).
Data use: Fault reporting of system or accommodation.
Data volume:

Table 46. *Data related to fault reporting.*

ID	Field	Description	Example solution
E.5.1	Type of ticket	The fault report either regards the accommodation or the system.	
E.5.2	Date & time		The system shall record the date and time for a certain fault report.
E.5.3	Problem/information		A user shall be able to describe the problem using both text and images.
E.5.4	Urgent		If the fault report regards an urgent issue, the system shall record this and send a special message to those who are affected.

E.6. University

The system shall retrieve and record the data described in Table 47 below.

Examples of source and use:
Data source: Imported from the university (external). Recorded when a student register (C.7) and during periodic check of student's academic status (C.4)
Data use: To ensure whether a student is fulfilling the criteria of using the system.
Data volume: One time per registered student and then one time per student per periodic check every semester.

Table 47. *Data imported from a university (external source).*

ID	Field	Description	Example solution
E.6.1	Academic status	Active or non-active. See section E1.2 for more	

		information.	
E.6.2	Number of passed credits		If a student has not passed enough credits, the system shall notify the student and initiate termination of contract (see C.6., F.1.2. and F.2.1.).
E.6.3	University	The name of the university.	

E.7. Payment

The system shall be integrated with a payment service provider in order to manage different kinds of payments within the system. The system shall identify the sender and the receiver of payment based on the reason for payment.

Examples: C.5, C.12, C.13, C.14

Data source: Payment status is imported from the payment service provider (external). Data are recorded during pay the rent (C.5), pay for publishing of accommodation (C.12), pay for upgrade of listing (C.13), and publish and pay for advertisement (C.14).

Data use: Billing for rent, publishment of accommodation, upgrade of listing and billing of advertisements.

Data volume:

Table 48. *Data related to a payment.*

ID	Field	Description	Example solution
E.7.1	Receiver of payment	Name and payment details of the receiver	
E.7.2	Sender of payment	Name and payment details of the sender	
E.7.3	Amount	The amount of payment in SEK	
E.7.4	Payment deadline	The last day for when the payment shall be paid	
E.7.5	Payment type	Swish, bank card, credit card, bank transfer	
E.7.6	Payment details	Details of payment type	The system shall allow students and landlords to save their payment details for future use.

E.7.7	Payment status	Whether the payment is paid	The system shall record the current status of the payment and notify the parties concerned when the payment is completed.
E.7.8	Reason for payment	See C.5, C.12, C.13, C.14 for information.	

E.8. Advertisement

When an advertiser wants to publish an advertisement, the data fields in Table 49 below is needed for the request. Depending on whether the request gets accepted, the data might be updated and the data fields in E.7 might be needed as well.

Examples of source and use:
Data source: Recorded during publish and pay for advertisement (C.14).
Data user: Publishment of advertisements.
Data volume:

Table 49. *Data related to advertisement.*

ID	Field	Description	Example Solution
E.8.1	Advertiser	Contact details for advertiser.	Company name, corporate identity number, complete postal address, phone, email, payment type
E.8.2	Advertisement	The content that the advertiser wants to publish.	Can be both media and text.
E.8.3	Start date	Start date of advertisement.	
E.8.4	End date	End date of advertisement.	

E.9. Statistical data

The system shall at least record the statistical data in Table 50 below.

Table 50. *Statistical data of special interest.*

ID	Field	Description	Example Solution
E.9.1	Total number of unique visitors per day		
E.9.2	Total number of unique signed in users per		

	day		
E.9.3	Total views per page		
E.9.4	Total views per listing		
E.9.5	Rate of use of filtering function	To what extent users use the filtering function.	
E.9.6	Most used filter factors	What factors users tend to use the most.	

F. Other functional requirements

The system must be able to perform the functions specified in this chapter.

F.1. System generated events

F.1.1. Notification about contract

Table 51. *Notifications about contract*

ID	The system must generate these reminders	Example solution
F.1.1.1	If the deadline for ending a certain contract is in 60 days, the system sends reminder to both parties about that. The system shall be able to calculate when the deadline is, and when to send the notification.	System sends an email to the landlord and the student to “warn” them that the contract ends in 60 days.
F.1.1.2	If the deadline for ending a certain contract is in 90 days, the system must offer extensions of the contract to both parties.	Systems sends an email to the landlord and the student offer to extend the period for ending the contract.
F.1.1.3.	The landlord gets notified when a student applied for the accommodation and waiting for the replay.	Systems sends an email to the landlord when the student is waiting for the answer for accommodation.

F.1.2. Notification when student is not student

Table 52. *Notifications when student is not student*

ID	The system must generate these reminders	Example solution
F.1.2.1	The system must notify both parties when student does not have student status anymore. The system fetches this data from universities.	When the student does not have student status, the system generates automatic email to both parties with information about the current student's status.
F.1.2.2	The system must send the notification to the student when the student lost the student's status.	System shall update the profile field about student status and send an email to the student to remind him that he/she does not have valid student status.

F.1.3. Notification when Data Integrity Policy is updated.

Table 53. *Notifications when Data Integrity Policy is updated*

ID	The system must generate these reminders	Example solution
F.1.3.1	The system must notify all users about new updates in Data Integrity Policy.	When a user logs into the account, a pop-up message appears to inform the user about changes in Data Integrity Policy.

F.2. Integration with other external systems

F.2.1. Integration with university system

As shown in figure 5 below, the system triggers a request to verify that a student is enrolled at the university. The university verifies the status of the student and generates the report including student status and number of credits. A check is performed every semester to ensure the student's status. Once the student's status is non-active (because not being enrolled or passing enough credits), the system will initiate the termination of the contract. When a termination of the contract is initiated, the system will notify the student that he/she has to move out within a specific period of time.

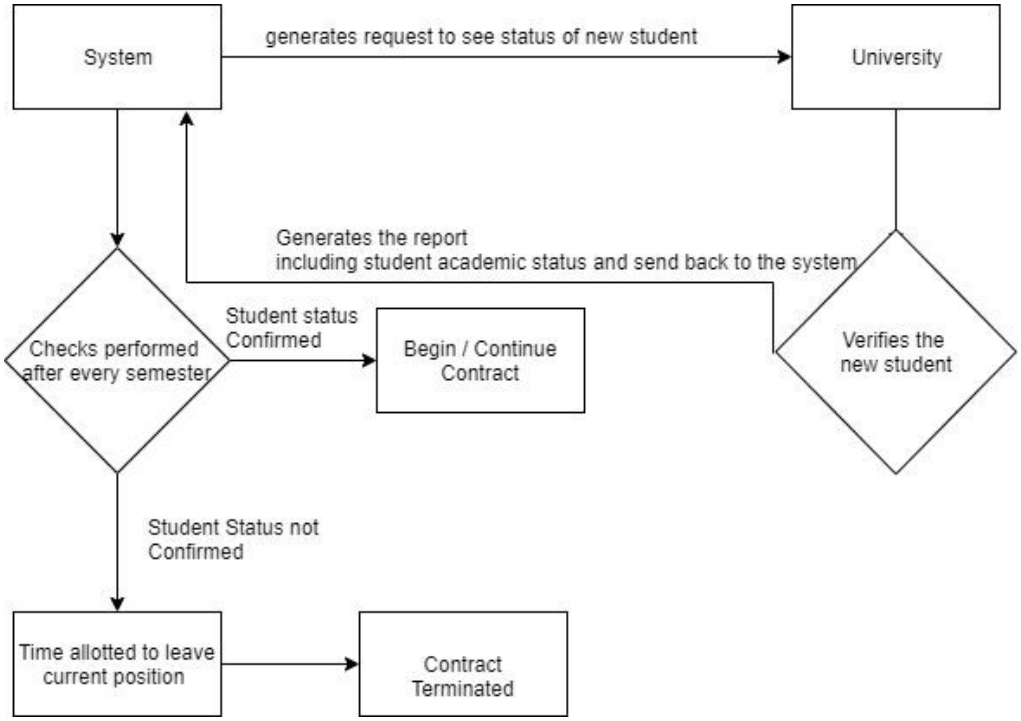

Fig 5. *Diagram explaining integration of the system with university systems*

F.2.2. Integration with payment system

SAM is designed to fully provide a flexible payment method to both students and landlords. The student receives automatic invoice and pays the rent on a specific day each month. The system will check if the amount deposited by the student is defined as in the invoice.

The student pays the rent to the system and the system in turn makes the payment to the landlord. The system shall pay the landlord on a regular basis regardless of the fact whether the student has made the payment on time. The student reimburses the system within a certain time. Failure to reimburse shall put the student on notice.

G. Non-functional requirements

The grid in Table 54 below has a row for each of the quality factors. Five most critical and important factors are identified and concerns about the same one are described below the table. They are stated as high-priority factors. Factors marked with numbers represent the most important factors for the SAM system and they are crucial. Factors marked with X has less impact on the crucial functionality for SAM.

Table 54. *Quality grid of non-functional requirements*

	Critical	**Important**	**As usual**	**Unimportant**	**Ignore**
Operation					
Integrity	**1**				
Correctness				**X**	
Reliability		**2**			
Usability	**3**				
Efficiency			**X**		
Revision					
Maintainability			**X**		
Testability			**X**		
Flexibility		**4**			
Transition					
Portability			**X**		

Interoperability		**5**			
Reusability					**X**

Concerns:

1. Having created an account on an application could make the personal data vulnerable to being hacked and misused. This negatively affects both the users and SAM owners.
2. SAM provides accommodation only for students. If the data about a student's status is wrongly presented, business plan as well as the service provided is negatively affected.
3. For the student/landlord who require the application in a language that they require, SAM supports multiple languages.
4. Owners always want to extend SAM with new functional requirements or tweak changes in the existing requirements. Software architecture is designed so that rapid changes in functionalities are supported.
5. User creates an account by providing his/her personal identification number. Personal identification number is automatically verified with the concerned authorities (external systems).

G.1. Security

G.1.1. Physical Security

Physical security of the system shall be secured. Only authorized personnel shall be allowed access to the system.

Source of stimulus System

Stimulus Physically secure the system

Artifact Security

Environment Normal usage mode.

Response Ensuring physical security will protect the data and application stored on the servers; ensuring the access by only authorized personnel will prevent data leak to outsiders.

Response measure Server condition, security alarms

G.1.2. Password Policy

The system shall enforce strong password rules and implement security policies such as two-factor authentication.

Source of stimulus	Users
Stimulus	Requires the users to create strong passwords for the accounts and implement two-factor authentication
Artifact	Security
Environment	Normal usage mode
Response	Strong passwords will make it less possible to access the system with techniques such as brute force. Two-factor authentication will make it nearly impossible to access the system without authenticating through another method such as sms.
Response measure	System security, System access logs

G.1.3. Waiting period policy

To prevent brute-force attack, the system can implement waiting period between successive login attempts or notifying the users in their email address or through SMS regarding the login attempts.

Source of stimulus	System, Users
Stimulus	Requires the users to wait for a certain period after a number of successive login attempts
Artifact	Security
Environment	Normal usage mode.
Response	Making the user wait a certain period would discourage automated bots to launch brute-force attack to the system; notifying the users regarding such attempts will make them knowledgeable to such attempts and try to make strong passwords.
Response measure	System security, System access logs, Change of password

G.1.4. Save password storage

The passwords are secured and not stored as plain text.

Source of stimulus	Security breach.
Stimulus	Password DB acquired by an unauthorized external party.
Artifact	System / Database
Environment	Normal usage.
Response	Password stored as hash values and not as plain text.
Response measure	Hash value for password.

G.2. Reliability

G.2.1. Probability of failure

The system’s probability of failure when student searches and apply for accommodation shall be less than 1 out of 100000 visit.

G.2.2. Validation of admitted student

The validation of whether a student is admitted at a university or not shall be 100 percent accurate.

G.2.3. Validation of active student

The semester wise validation whether the student is active or not shall be consistently work throughout the time duration for rent.

G.2.4. Roll back of registration process

The registration process shall roll back when validation of admission at a university fails to get access to the admission information.

G.3. Accessibility

Accessibility is making the system accessible for as many users as possible.

G.3.1. Multiple language support

The platform has options to be used in Swedish or English as these are requirements for students in the Universities.

Source of stimulus	End user
Stimulus	Tries to give options to use portal on more than one language
Artifact	User interface
Environment	Normal usage mode
Response	By selecting other language, entire platform data switches on selected language
Response measure	Task time, user satisfaction

G.3.2. Visually impaired support

The platform has options to be used by the visually impaired users.

Source of stimulus	End user.
Stimulus	Color blind; suffers from eye disease
Artifact	User interface.
Environment	Normal usage.
Response	Second version of UI visual design with more contrast between
Response measure	Task time, user satisfaction.

G.4. Compliance

G.4.1. Minimal square meters per person

The system should be compliant with national, provincial, and local laws regarding minimal accepted square meters per person and accommodation.

G.5. System compatibility

G.5.1. Different platform support

The user should be able to use the platform efficiently on multiple types of devices.

Source of stimulus	End user
Stimulus	Use the portal equally efficient on the tablet, laptop/PC or smartphone
Artifact	System
Environment	Normal usage
Response	Same presentation of platform on multiple types of devices
Response measure	User satisfaction

G.6. Interoperability

G.6.1. Exchange data with other system

SAM platform should be able to get student status from universities.

Source of stimulus	System
Stimulus	Check if student has valid student status
Artifact	System
Environment	Normal usage
Response	Student has student status or he/she does not have it

Response measure	Completion of processing time from request to response

G.6.2. Cooperation with bank system

SAM platform should provide payment possibility for student’s rent.

Source of stimulus	System
Stimulus	Student pays the rent each month
Artifact	System
Environment	Normal usage
Response	Payment received
Response measure	Completion of processing time from request to response

G.6.3. Validate personal identification number

When a private person is creating an account, personal identification number is validated.

Source of stimulus	System
Stimulus	A user enters personal identification number when creating an account
Artifact	System
Environment	Normal usage.
Response	Personal data validated.
Response measure	Completion of processing time from request to response.

G.6.4. Validate corporate identity number

When an agency is creating an account, corporate identity number is validated.

Source of stimulus	System
Stimulus	A user enters corporate identity number when creating an account

Artifact	System
Environment	Normal usage.
Response	Corporate data validated.
Response measure	Completion of processing time from request to response.

G.7. Extensibility

G.7.1. Extend system with new functionalities

SAM should be easy to extend in order to incorporate new functionalities.

Source of stimulus	System
Stimulus	New functional requirement on the system
Artifact	System
Environment	Normal usage
Response	Implement new requirements without negatively affecting existing functions
Response measure	Existing functionalities do not fail in task completion

H. Future extensions

This section defines what features or specifications could be added to the system at a later stage. Each of these specifications are allotted a level of importance. The levels of importance range from the lowest (1) to the highest (3). The features having the highest points will have highest priority during implementation and vice versa.

Table 55. Notifications before contract deadline

ID	Feature/ Specification	Description	Importance
H.1.	Insurance Package	An insurance package that the student can avail while renting the apartment. This package can include policies like protection of personal belongings or liability coverage.	3
H.2.	Temporary tenant swap	The system could allow and support students to let out their apartment temporarily to another student when they are away for summer work or thesis for a period of 1 year or less.	2
H.3.	Student Query Analysis	The system could analyse the preferences of students by drawing some statistics from their queries and searches. The results can then be used to suggest the landlord for possible changes or additions to their accommodation posting.	2
H.4.	Accessing data from external systems	To make the system more lightweight, most data could be accessed from external system. This could help scaling down storage of the system.	3
H.5.	Videos showing interviews of users	Once the system is in use, interviews of previous or	1

		current users can be posted on the homepage. This would be a	
H.6.	Friend and family locator	This feature of the system would help students track each other or their family members and stay in touch with instant messaging. They can also find and contact their friends who are close by.	1
H.7.	Watching Ads for discounts on services	Advertisers pay for using the system to post their Ads. The system uses this revenue to provide discounts on services for students who take time to watch these ads.The services on which discounts could be provided are dry cleaning, Taxi booking, restaurants, etc. Such a feature would benefit the students and also increase system ARPU.	1
H.8.	Like/Dislike view of listings	The accommodation listings could be displayed to the students in a Like/Dislike fashion. This would be a feature that would be easy to use and give the system better insights to analyse the trending accommodations.	2

I. Mockup

In this chapter, sample mockup what the proposed system should look like is shown and described. Mockup is an important technique that helps to realize what the system will look like. The designed mockup shows the homepage, registration, search, publish and apply for accommodation pages. The mock-up shows only the core functional requirements of the SAM system as a sample overview.

I.1. Homepage

epage of
ews about

Fig 6. *SAM homepage*

I.2. Registration Page

The registration page shows the page where students and landlords create account to apply for

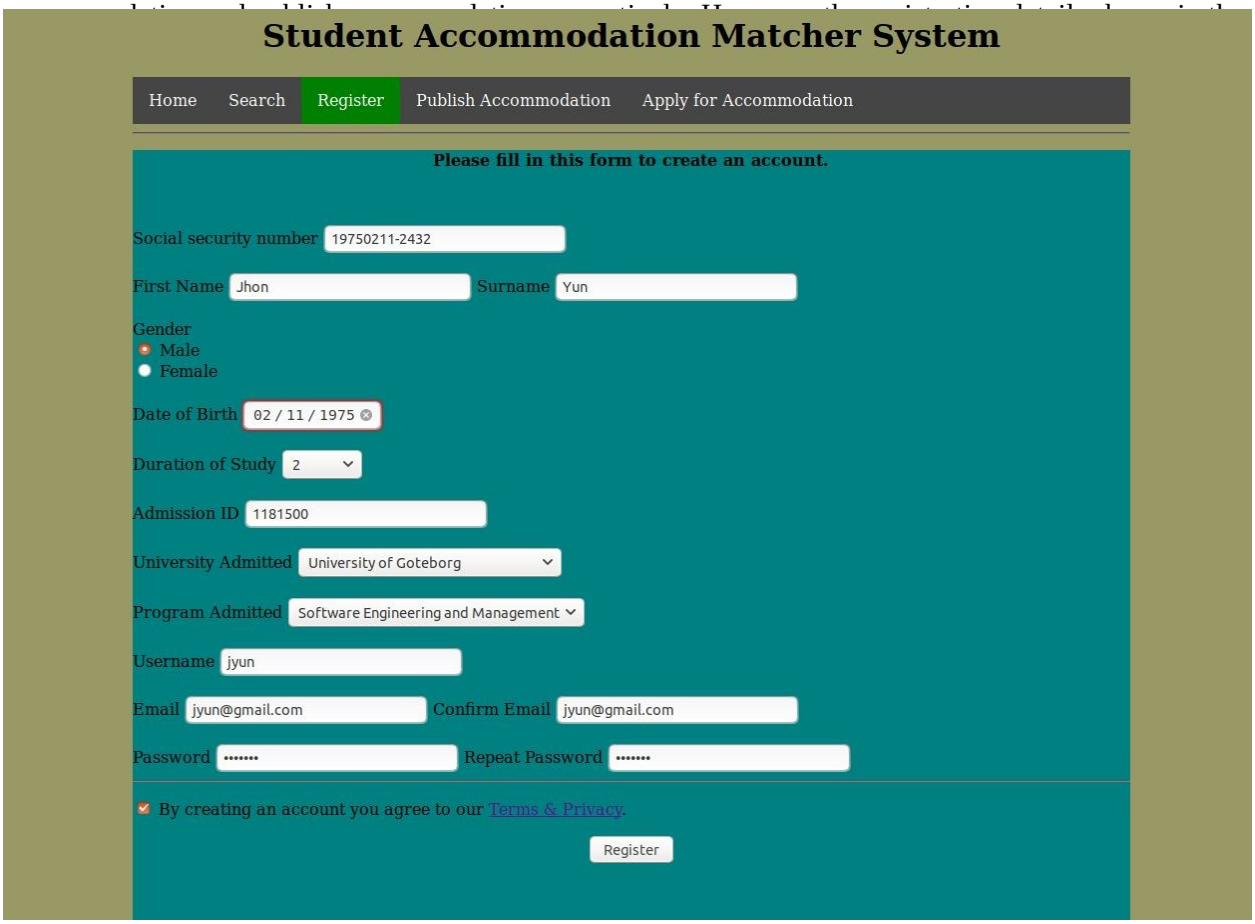

Fig 7. *SAM Registration page for Student*

I.3. Search Page

In the search page, visitors can search available accommodation without creating an account. They can

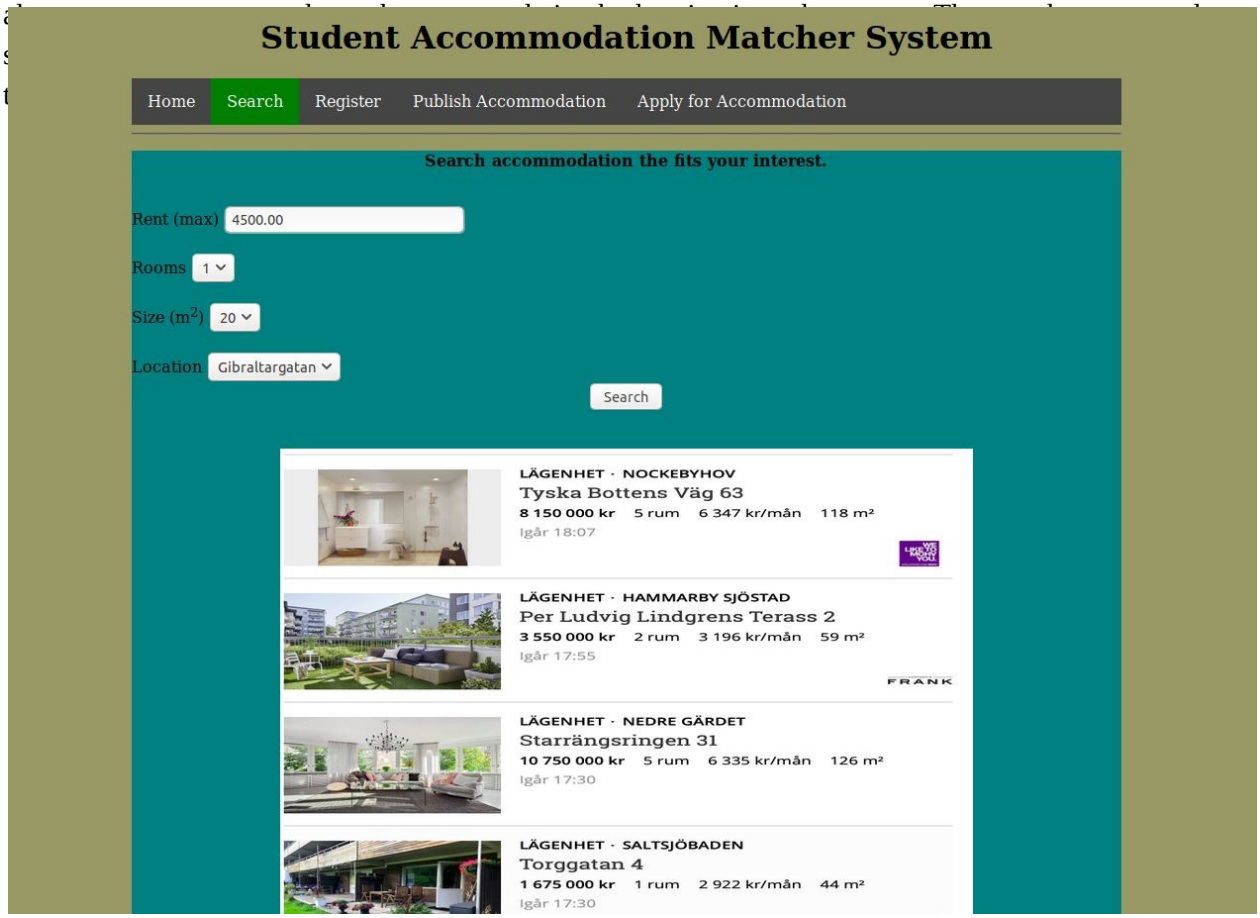

Fig 8. *SAM Search page*

I.4. Publish Accommodation

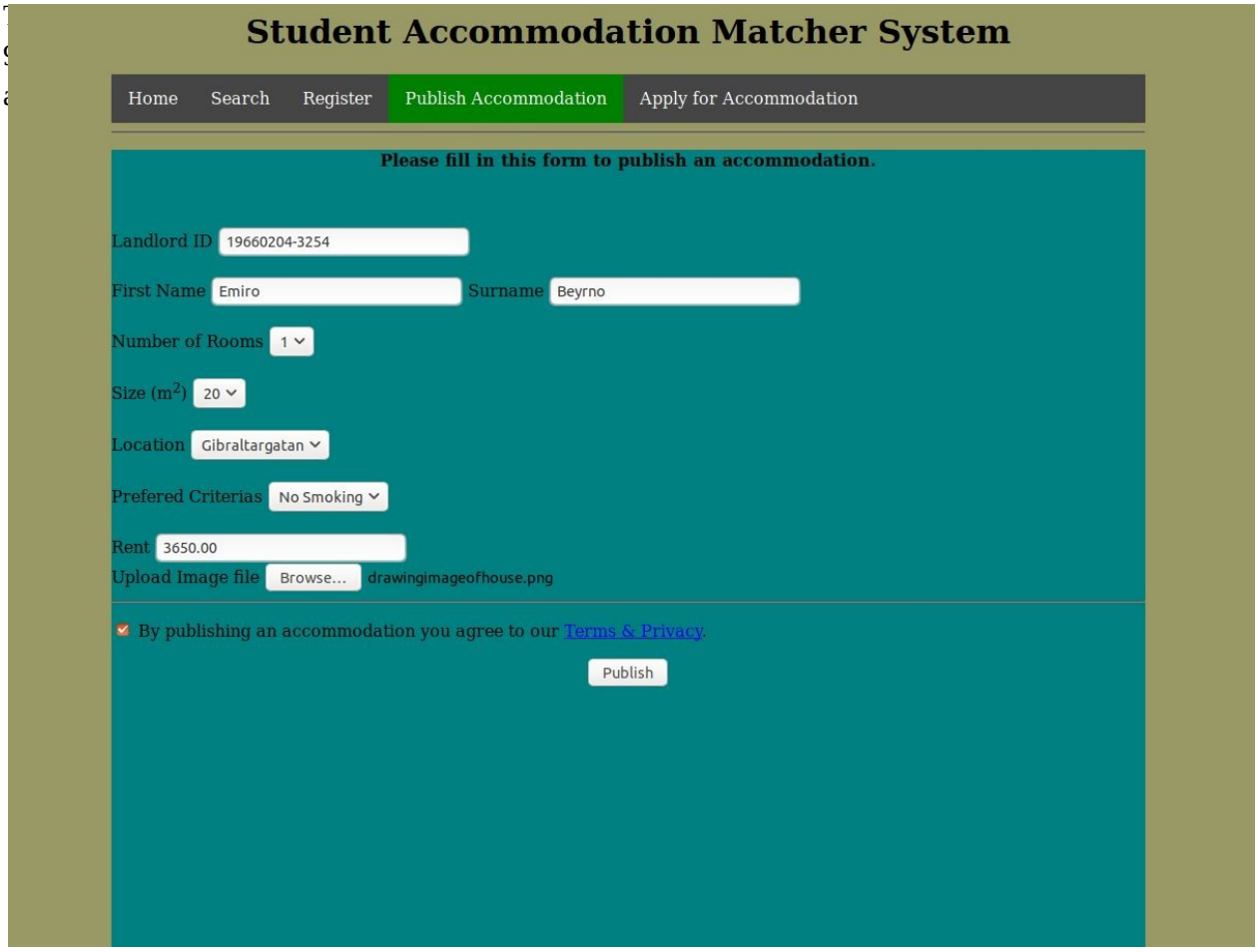

Fig 9. *SAM Publish Accommodation page*

I.5. Application for Accommodation

Application accommodation page should accept the detail information as shown in the figure below. The s
s
a

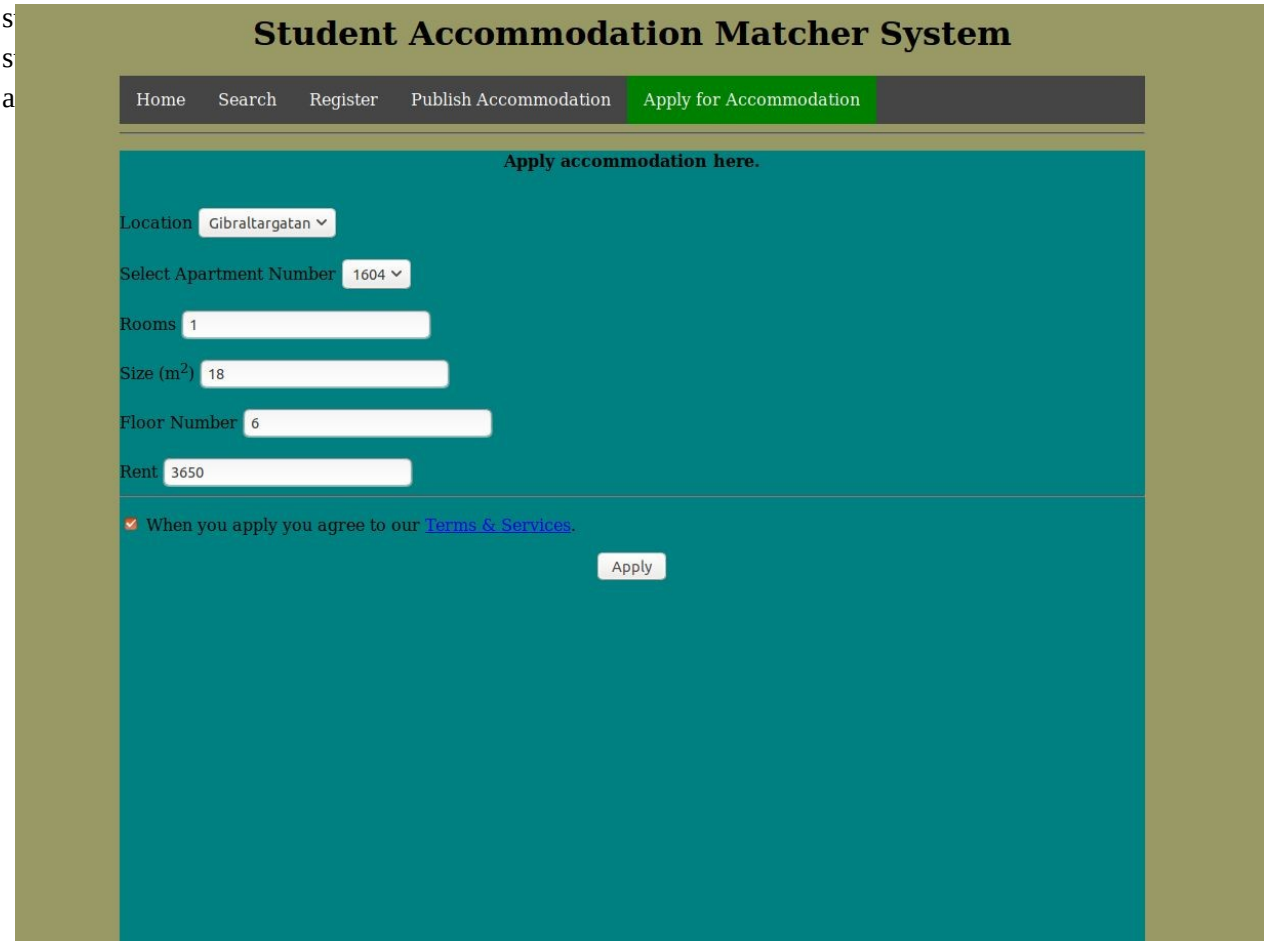

Fig 10. *SAM Apply for Accommodation page*

J. Inter-dependencies

Prioritization might be difficult to perform if there are a lot of interdependencies between requirements. If one requirement is really important but is dependent on another requirement, then both of them should have a high priority and might even have to be implemented in the same release because of the value and cost associated with it.

In order to provide an overview and explanation of how certain requirements might be related to each other, a description of a subset of requirements and their inter-dependencies is listed in Table 56 below. As seen in the table, some of the dependencies have branches of further dependencies (e.g. C.7. requires C.4. where C.4. requires F.2.1.). Since there might be alternative solutions for specific requirements though, some of the inter-dependencies stated below might change or even get eliminated because of another solution than the original.

The strategy of using this kind of relations worked as a base for the grouping of requirements during prioritization. C.1. requires C.2. shows e.g. that there might be good to prioritize them together since they are both needed to provide value to some of the stakeholders.

Table 56: *A subset of inter-dependencies*

Inter-dependency	**Description of interdependence**
C.1. requires C.2.	A student will never be able to apply for an accommodation unless a landlord has the ability to publish an accommodation.
C.3. requires C.1.	A landlord can't offer an accommodation to a student unless the student has an ability to apply for it.
C.3. requires C.2.	There must exist an accommodation before a landlord can offer it to a student.
C.4. requires F.2.1.	In order to perform an easy and fast check of the student's academic status, there must be a functioning integration with the external university system.
C.5. requires F.2.2.	In order to pay the rent using the SAM-system, there must be a functioning integration with a payment provider.
C.6. requires C.4.	In order to have the desired level of automation, terminating a contract because of a student no longer being an active student requires the check of a student's academic status to function.
C.7. requires C.4.	In order to only allowing real students to register, the check of a student's academic status must be functioning.

C.8. requires C.7.	In order to sign in to an account, there must be a way to register for one.
C.9. requires C.7.	In order to manage profile settings, there must be a way to first register for an account.

References

[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