Requirements specification for

Student Accommodation Matcher System

Customer
Group10 AB

Supplier
...

The delivery comprises
Software, operation, and maintenance for an accommodation system

Table of contents

A. Background

A1. 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. Because of this international 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 coming here.

Moreover, there are also agencies that require you to pay for the time that you wait in the queue. Currently, there are several different systems you can search for an accommodation and each of them has different rules, queues and ways to apply.

Our product 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. The system aims to connect landlords and students with specific needs efficiently. The system also aims to decrease waiting time and increase the possibility for students to find an accommodation. The system streamlines the contract signing process, rent payment and enables direct communication between the landlord and the student.

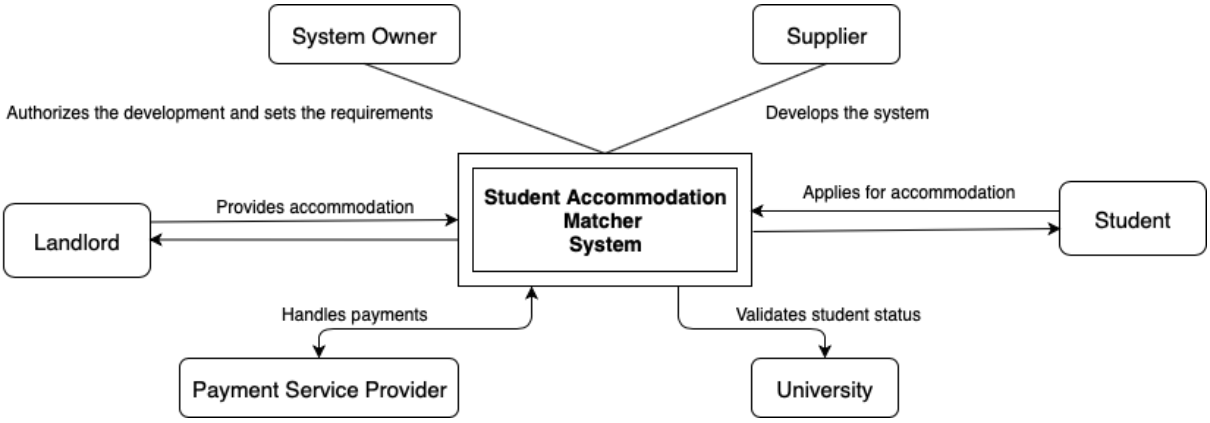

Fig 1. Context Diagram

The context diagram shows how various stakeholders and external systems interact with the system.

The system is depicted with a double-lined rectangle as it is the final product to be delivered. The arrows from the Student and the Landlords to the system indicate that they interact with system to apply and

provide accommodation respectively. These two stakeholders are matched with each other according to their preferences. The system acts as a contract mediator between the student and the landlord once the contract is signed.

Integration of the system with the University ensures that only admitted students apply to accommodation. During the period of contract between the student and the landlord, the system periodically checks the status of the student from the University, to validate their enrollment to a minimum number of credits.

Although all payments are made through the system, an integration with an external payment platform ensures secure and hassle free transactions, independent of the core functionality of the system.

The system owner sets the requirements and specifications of the system to the supplier who builds and maintains the system. The system owner also takes the responsibility of validating and authorizing the system delivered by the suppliers at all levels such as updates, security and validation.

B. High-level demand

This chapter explains how the customer's business goals are met through the requirements and how to mitigate high-risk requirements.

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 shows the steps in the flow and column 2 shows the related tasks and subtasks for each step in the flow. Chapter C shows the details.

B.1.1 Renting accommodation

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

Table 1. Flow for renting accommodation

	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 the table shown below.

Table 2. Flow for providing accommodation

	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

B2. Business goals

Business goals are set to describe main goals when providing the system. They are supposed to be done in the early phases of the system development, but some of them should be delivered already in the early proof concept. The customer expects that the system contributes to the goals as stated below.

Table 3. Business goals

Goals for the new system	**Solution vision**	**Related requirements**	**Deadlines**
Allow visitors to view the accommodation without having to actually register	Not having to register will lower the threshold for using the system. Registration of each potential user helps to calculate Average Revenue Per User (ARPU).	FR1	
Offer upgrade for landlords on individual accommodations	Increase a listing's placement in the search list, display the listing in the carousel. This will provide more profit which could be invested in the users.	FR4	
Automatically generate of the contract agreement between two users.	Get lower operational costs , avoiding paperwork and direct human involvement.	FR5	
Integrate system with other (external) IT systems.	Keep systems competitiveness by an easy adaptation to new environment if changes are required.	FR6	

Review of the rented accommodation by previous tenants.	This gives the potential tenants a feeling of assurance and comfort in giving preference to an accommodation and continuing to use the system.	FR7	
Allowing the students to be able to view and apply for the accommodation without the need of an upfront payment	This allows the students to make no payments when they are waiting to sign a contract; and only make the payments once they have signed one.	FR4, FR8	
Provide accommodation for students within a specified period of time.	The system should notify the landlord to respond to the applicants in queue within a week. No response from the landlord, removes the accommodation from the listing, notifying the students that the listing is no longer available for application, with reasons.	FR12, FR13	
Ensure that landlords and students comply with the contract terms.	The system, once provided notification regarding non-compliance of contract terms by either party, should penalize the non-compliant party in some manner.		

B3. Early proof of concept

According to the contract, both parties can terminate the contract if the early proof fails.

The following requirements are considered high-risks. If core features are not done in predicted time, the supplier won't fulfil customer's expectations. This possibly could terminate the contract from customer's side. The supplier is expected to deliver plan when the early prototype will be delivered. The delivery plan must be accepted by the customer.

Table 4. Early proof of concept

Areas where an early proof of concept is required	**Example of proof**	**Code**
Efficient integration of external systems.	- Check that the user who is registered for a student account actually is admitted at a University. Requirement for testing this feature must include a response from Admissions about the student's status. - Integrating a payment service with the system.	
Posting of the accommodation listing.	Without any accommodations there will not be any students applying for accommodation.	

B4. Functional Requirements

The system should support the following functional requirements.

Table 5. Functional requirements

Rqrt ID	**Requirement**
FR.1	Landlords and tenants shall be able to register for user accounts
FR.2	Landlords shall be able to perform tasks in Work Areas 2 and 4
FR.3	Registered tenants and unregistered visitors shall be able to view all of the listings
FR.4	The landlord shall be able to upgrade for an individual listing
FR.5	The system shall auto-generate tenancy contract to be digitally signed
FR.6	Tenants shall be able to write reviews for an accommodation
FR.7	The system shall display the final price for accommodation that includes commissions and taxes
FR.8	Students shall be able to perform tasks in work areas 1 and 4, and tasks C.5 and C.6 in work area 3.
FR.9	Landlords shall be able to perform tasks in work areas 2 and 4, and C.6 in work area.
FR.10	The system shall support task C.4 in work area 3.
FR.11	The system shall allocate a student a default accommodation before a deadline if they haven’t signed any contract.
FR.12	The system shall notify the landlord to respond to the applicants in queue within a week.
FR.13	The system shall provide landlords and students with the option of choosing preferences.
FR.14	The system shall integrate external systems for publishing/rent payment.
FR.15.	The system shall support advertisements from third parties.
FR.16.	The system shall store all users data.
FR.17.	The system shall be designed to be able to easily expand with future implementation.

B.5 Non-Functional Requirements

The system should support the following non-functional requirements.

Table 6. Non-functional requirements

Rqrt ID	**Requirement**	**NFR Category**
NFR.1	Standards such as WAI-ARIA should be incorporated in the system during development.	Accessibility
NFR.2	For people with hearing disability, there could be a functionality of text-to-speech.	Accessibility
NFR.3	Color scheme that is suitable for people with colorblindness should be used.	Accessibility
NFR.4	The system should be available throughout the clock and ensure the highest uptime.	Availability
NFR.5	The system should be compliant with national, provincial, and local laws.	Compliance
NFR.6	The system should support application and data replication to make it recoverable in case of a disaster.	Disaster Recovery
NFR.7	The system should be properly documented and use proper programming language conventions.	Documentation
NFR.8	It should be easy to extend the system to incorporate new functionalities.	Extensibility
NFR.9	The system should support major languages.	Internationalization and localization
NFR.10	The system should be available in multiple platforms.	System compatibility
NFR.11	The system should support display in multiple form-factors.	System compatibility
NFR.12	The system should provide privacy to the users and conform to privacy laws.	Privacy
NFR.13	The system should verify and validate access to the data.	Privacy

NFR.14	The system should be reliable. For instance, if the data is modified in one part of the system, that data should be modified everywhere in the system. There should not be more than one version of the same data.	Reliability
NFR.15	The system should conform to ACID (atomicity, consistency, isolation, and durability) standard.	Reliability
NFR.16	The system should be easily scalable. The system should have similar performance for a smaller number of users and a very large number of users.	Scalability
NFR.17	System server locations should be physically secure and only authorized personnel should be allowed access in those locations.	Security
NFR.18	The system should enforce strong password rules and implement two-factor authentication.	Security
NFR.19	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.	Security
NFR.20	The system should provide post-development support.	Supportability
NFR.21	The system should pass tests such as edge-case testing.	Testability
NFR.22	The text, links, and buttons in the system should make sense in any language that is supported by the system so that the users are not misled by the system.	Usability
NFR.23	The design should look ‘clean’ as much as it can to improve visual pleasure for the users.	Usability

B6. 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 the table below. To provide better overview, the tables have space for several proposals (columns A, B and C). The scores represents -2(not supported or very inconvenient), -1(inconvenient), 0(current or enough), 1(efficient), and 2(very efficient).

Minimum Score: All areas must meet the minimum score, a system that is below the minimum score in any area is useless in practice.

Minimum requirements: The system must meet the minimum scores below on all requirements areas.

Table 7. Minimum requirements

Requirement area	**Reason**	**Minimum score**	**Score**		
			A	**B**	**C**
C1-C2. Apply/Publish	The core of the system. Provides searching and publishing accommodation.	2			
C3 – C6. Offers/Contract	Ensuring that only applicants are students, contract for an accommodation is signed, or terminated.	1			
C7 – C10 / H1. Account management	Provides security for the system's users.	1			
D. Data. Assessed through the task support.		0			
E1.Advertising	Increases profit value.	0			

E2 . Expansion of the system	Potentially will increase profit value.	0			
F1. University system	Increase usability of the system.	1			
F2. Payment system	Increase usability of the system.	1			
H. Security		1			
L. Operation, support, maintenance		0			

B7. Selection Criteria: MoSCoW

Priority of requirements are chosen based on MoSCoW. MoSCoW categorizes requirements according the following order:

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 8. MoSCoW requirement prioritization

Requirement	Rama	Pernilla	Teklit	Asad	Shameer	Sanja	Total
C1-C2. Apply/Publish	M	M	M	M	M	M	M
C3 – C6. Offers/Contract	M	M	M	M	M	M	M

C7 – C10 / H1. Account management	M	S	S	M	S	S	S
D. Data. Assessed through the task support.	M	S	S	M	S	S	S
E1.Advertising	C	S	S	S	C	S	S
E2 . Expansion of the system	C	C	C	S	C	C	C
F1. University system	S	S	S	M	S	C	S
F2. Payment system	S	S	S	S	S	S	S
H. Security	M	M	M	M	M	M	M
L. Operation, support, maintenance	S	S	C	M	C	S	S

B8. Selection criteria: Priority Scorecard

Features are categorised according to different criteria. Chosen categories are:
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.

Table 9. Scorecard requirements prioritization

Category	**User experience**	**Revenue**	**Operation Efficiency**	**Security**	**Maintenance**	**Total**
Weight	15%	30%	20%	25%	10%	100%
Requirements	**Score**					**Priority**
C1-C2. Apply/Publish	70	30	20	20	30	31
C3 – C6. Offers/Contract	60	50	50	20	30	37
C7 – C10 / H1. Account management	0	0	0	100	10	26
D. Data. Assessed through the task support.	0	0	0	100	10	26
E1.Advertising	0	70	20	0	20	27
E2 . Expansion of the system	20	40	30	20	10	27
F1. University system	0	0	0	90	0	22,5
F2. Payment system	0	10	0	90	0	25,5
H. Security	0	0	0	100	30	28

C. Tasks to support

In this chapter tasks that must be supported when users interact with the system are described. Numbered tasks do not have to be carried out in a specific order and some of them are optional. There are also subtasks that might be repeated during one particular task and some of them can be performed in several ways.

The 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. Who does what, can depend on the solution for the area. The requirement is that the system shall support all of the described tasks in this chapter.

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: Students.
Environment: Web system (currently)

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 10. Subtasks for C.1

Subtask		**Example solution**
C.1.1	Search for accommodation without preferences	
C.1.1a	Search for accommodation with chosen preferences	
C.1.2	Display detailed information about selected accommodation.	
C.1.3	Apply for the selected accommodation	
C.1.4	Accept an offer	
C.1.5	Reject an offer	
C.1.6	Sign the contract	

Work area 2. Providing accommodation

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

Table 11. Subtasks for C.2

Subtask		**Example solution**
C.2.1	Enter detailed information about the accommodation.	
C.2.2	Add images (mandatory).	
C.2.3	Describe student profile criterias (non-smoking etc.).	
C.2.4.	Payment of service charge.	
C.2.5.	Publish the post.	
C.2.5.a	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: Students, Landlords.
Start: An accommodation has an applicant.
End: Sign the contract.
Frequency: 1,000 offers per day.

Table 12. Subtasks for C.3

Subtask		**Example solution**
C.3.1	Accept or reject applicants in queue.	
C.3.2	Sign the contract.	

Work area 3. Contract maintenance

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 terminate the contract.

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

Table 13. Subtasks for C.4

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

C.5 Pay the rent

This task involves the handling of the payment for the rent.

Users: Students, Landlords.
Start: Landlord initiate the monthly rent process.
End: The receipt of the payment.
Frequency: Once every month.

Table 14. Subtasks for C.5

Subtask		Example solution
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. Either it happens because of an end of a contract period or because a student is no longer fulfilling the criteria for renting the accommodation. It can also happen because of a breach of contract by either party. When termination process is initiated, the student has 90 days to move out.

User : Student, Landlord, System.
Start : Initiate the termination process.
End: Termination of the contract.

Table 15. Subtasks for C.6

Subtask		Example solution
C.6.1	Initiate closing contract.	
C.6.2	Accept the end of the contract.	

C.11 Accommodation fault reporting

This task provides the student with the possibility to report any fault related to the accommodation.

Users: Landlord, Student.
Start: Student reports fault about the accommodation.
End: Landlord notified.

Table 16. Subtasks for C.11

Subtask		Example solution
C.11.1	Open the ticket.	
C.11.2	Close the ticket.	

Work area 4: Account management

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

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.

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

Table 17. Subtasks for C.7

Subtask		**Example solution**
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 applications, publishing etc.

Users: Landlord, Student.
Start:
End:

Table 18. Subtasks for C.8

Subtask		**Example solution**
C.8.1.	Login to the system	
C.8.2	Request password reset	
C.8.3.	Logout off the system	

C.9 Update / Change 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 19. Subtasks for C.9

Subtask		**Example solution**
C.9.1.	Change personal information	
C.9.2	Change notification settings	
C.9.3	Delete account	

Work area 5: Maintenance and technical support

C.10 System fault reporting

This task provides students and landlords with the possibility to report any fault related to the system.

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

Table 20. Subtasks for C.10

Subtask		**Example solution**
C.10.1	Open the ticket.	
C.10.2	Close the ticket.	

Appendix 1

Stakeholder map

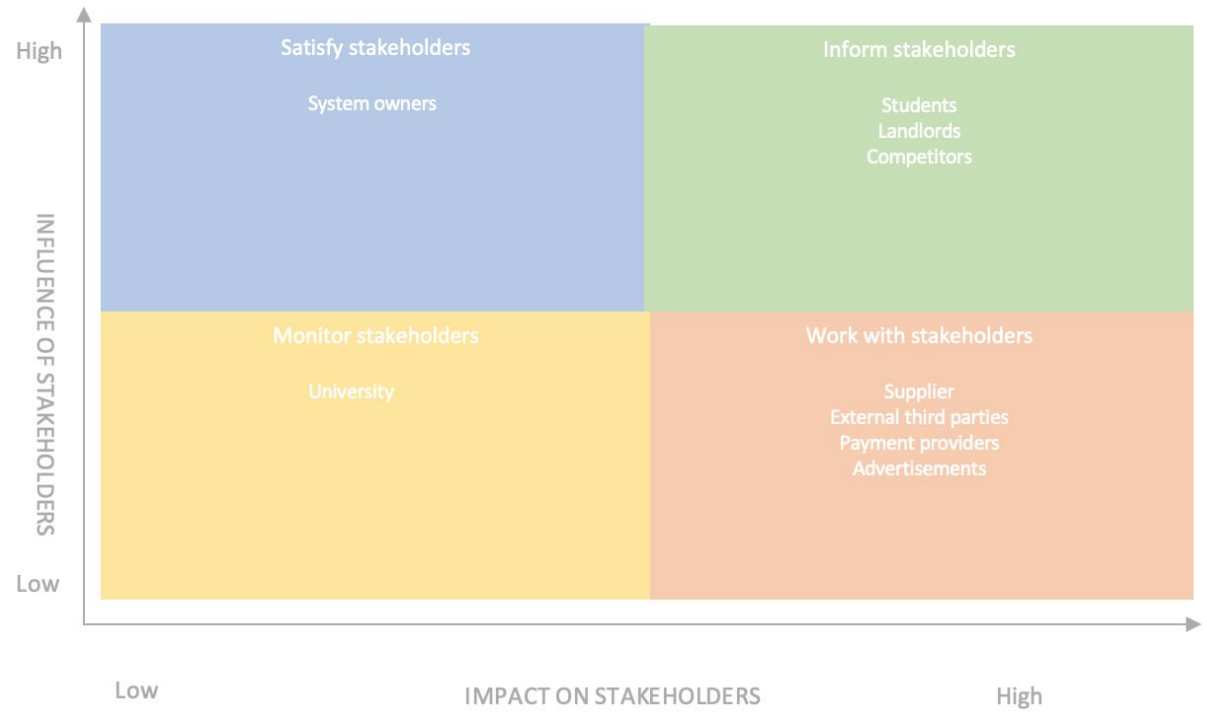

Fig 2. Stakeholder map: Stakeholders placed high in the graph are highly influenced and impacted by the system, while stakeholders placed below in the graph has less influence and they are less impacted with the system.