

DAT231/DIT284

REQUIREMENTS SPECIFICATION
RELEASE 3

**System for providing temporary housing to
students in Sweden**

Mola Ayenew
(*gusayenmo@student.gu.se*)
Tommy Ernsund
(*gusernto@student.gu.se*)
Pouya Faramarzi
(*pouyaf@student.chalmers.se*)
Wei Guo
(*gwei@student.chalmers.se*)
Sisi Lai
(*lsisi@student.chalmers.se*)
Abhinav Prasad
(*guspraab@student.gu.se*)



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020

Change log

Release 1.0 2020-09-21

- Chapter A
 - First draft of sections A1., A2., A3..
 - First draft of figures A.1, A.2, A.3.
- Chapter B
 - First draft of section B1..
 - First draft of tables A.2, A.3, A.1.

Release 2.0 2020-10-05

- Chapter A
 - Second draft of sections A1., A2., A3..
 - First draft of sections A4., A5..
 - Second draft of figures A.1, A.2, A.3.
 - First draft of figure A.4.
 - First draft of tables A.5.
 - Second draft of tables A.2, A.3, A.1.
- Chapter B
 - Second draft of section B1..
 - First draft of sections Supplemental user stories(*), B3., B4. and Functional requirements(*).
 - First draft of figures B.1.
 - First draft of tables User story table(*), Functional requirements table(*), B.2, B.3.
- Chapter C
 - First draft of sections C3., Detailed data requirements(*), C4..
 - First draft of figures C.3, C.4, Confirm and pay(*).

(*): Section/figure/table was removed or replaced in the following release.

Release 3.0 2020-10-23

- Chapter A
 - Third draft of sections A2.,
 - Second draft of sections A4., A5..
 - First draft of sections A3.3., A5.2..
 - Third draft of figures A.1, A.3.
 - Second draft of figures A.4.
 - Third draft of tables A.2, A.1.
 - Second draft of tables A.5.
- Chapter B
 - Third draft of sections B1..
 - Second draft of sections B3., B4..
 - First draft of sections B2., B4.3..
 - Second draft of figure B.1.
 - Second draft of tables B.2, B.3.
 - First draft of tables A.4, B.4.
- Chapter C
 - Second draft of sections C3., C4..
 - First draft of sections C1., C2..
 - Second draft of figures C.3, C.4.
 - First draft of figures C.1, C.2.
 - First draft of tables C.1, C.2, C.3, C.4, C.5.

(*): Section/figure/table was removed or replaced in the following release.

Contents

A High-level description	1
A1. Goal and scope	1
A2. Stakeholders	2
A3. Business case	4
A3.1. Business goals	4
A3.2. Flows	5
A3.3. Business Constraints	6
A4. Core functionality	7
A5. Critical attributes and constraints	9
A5.1. Key attributes	10
A5.2. Technical Constraints	11
B User requirements specification	12
B1. Tasks to support	12
Work area 1: Account management	12
W1.1 Create an account	12
W1.2 Verify student	13
W1.3 Verify home owner	13
Work area 2: Accommodation management	14
W2.1 Search for available accommodation	14
W2.2 Browse list of accommodations	14
W2.3 Browse map of accommodation	15
W2.4 Post accommodation(s) with accompanying details .	15
Work area 3: Communication	16
W3.1 Contact home owner	16
W3.2 Make payment	17
W3.3 Select student for communication	17
B2. Feature requirements	19
B3. Data requirements	19
B3.1. Entity Relationship Diagram	19
B3.2. Data Dictionary	21
B3.2.1. User	21
B3.2.2. Student	21
B3.2.3. Home Owner	22
B3.2.4. Accommodation	22
B3.2.5. Contract	23

B4.	Proposed prioritization	24
B4.1.	\$100-test	24
B4.2.	Ranking (sorting)	24
B4.3.	Final prioritization	25
B4.4.	Next release	26
B4.5.	Second release	26
B4.6.	Future releases	26
C	System Requirements	27
C1.	Other system requirements	27
C1.1.	Integrity/Security	27
C1.2.	Usability	28
C1.3.	System integration	29
C2.	Traceability matrix	31
C3.	UI prototype	34
C4.	Acceptance tests	35

List of Figures

A.1	Context diagram of suggested system	2
A.2	Stakeholder map	2
A.3	Stakeholder power and sentiment chart	3
A.4	Use-case diagram	7
B.1	Entity-Relationship Diagram	20
C.1	External System Context Diagram	30
C.2	Traceability matrix for the proposed system (link)	33
C.3	User flow - The steps that need to be taken for a student to sign up and verify, with the home page as the entry point	34

C.4 User flow - Starting at the search page (left), a user can either click on a listing or on the map to be taken to a page showing more information about the selected accommodation (right)	35
--	----

List of Tables

A.1 Business Goals	5
A.2 Logical Flow For Students	6
A.3 Logical Flow for Home Owners	6
A.4 Core Requirements	8
A.5 Quality grid for the proposed system.	9
B.1 Feature requirements	19
B.2 Each stakeholder's distribution in the \$100 test	24
B.3 Each stakeholder's distribution in the \$100 test (ranked)	24
B.4 Final prioritization based on the combined score of the \$100 test and ranking	25
C.1 Security requirements	28
C.2 Usability requirements	29
C.3 Interoperability requirements	29
C.4 Traceability ID and goal and requirement descriptions (1)	31
C.5 Traceability ID and goal and requirement descriptions (2)	32

A

High-level description

A1. Goal and scope

The first step for international students when they enter a new country is usually to find a place of residence. In Sweden, this process is significantly harder due to the existing student housing crisis and year-long queuing required to get a contract. The current housing systems for students such as Boplats, SGS, and Chalmers studentbostäder, offer an edge to students who sign up ahead of their studies and acquire longer queue times. On the other hand, students unaware of this system are led into tough situations as they are being forced to look for temporary solutions. These tough situations include signing second hand contracts written by unreasonable landlords, getting into scams, and paying a premium for hotels/Airbnb.

The goal of this system is to alleviate the aforementioned tough situations by offering both incoming international and native students a portal for finding affordable temporary and acute housing. Subsequently, the system also provides a platform for existing home owners to offer hospitality and lodging services to students.

In the context diagram, found in Figure A.1, an illustration is made to represent the different actors and components of the system and display how they interact with one another.

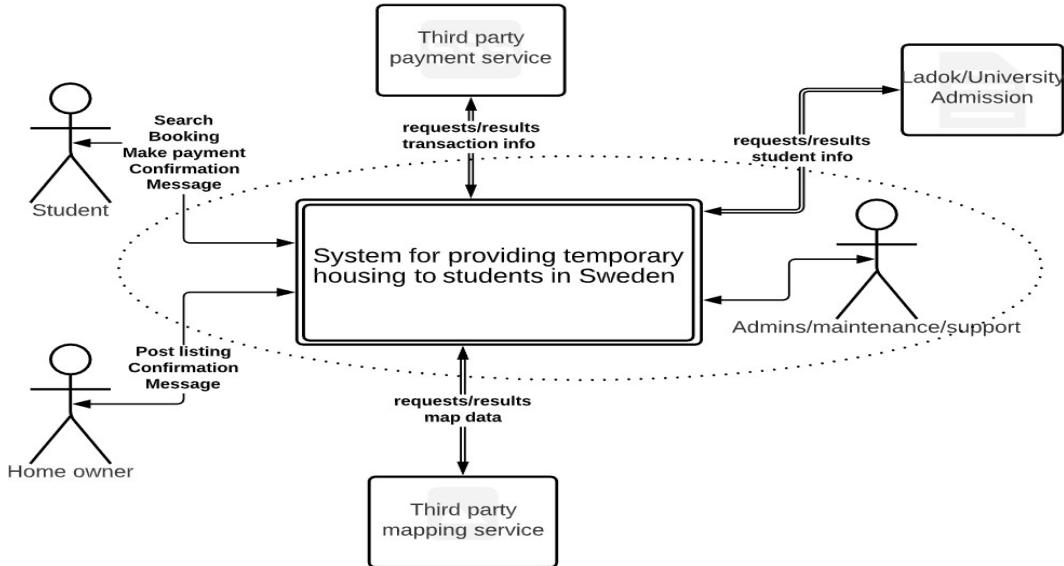


Figure A.1: Context diagram of suggested system

A2. Stakeholders

Primary stakeholders for the system are **Students** and **Home Owners**. Secondary Stakeholders are the parties which indirectly get affected by the system, such as competitors and business partners. Figure A.2 maps primary and secondary stakeholders with the system.

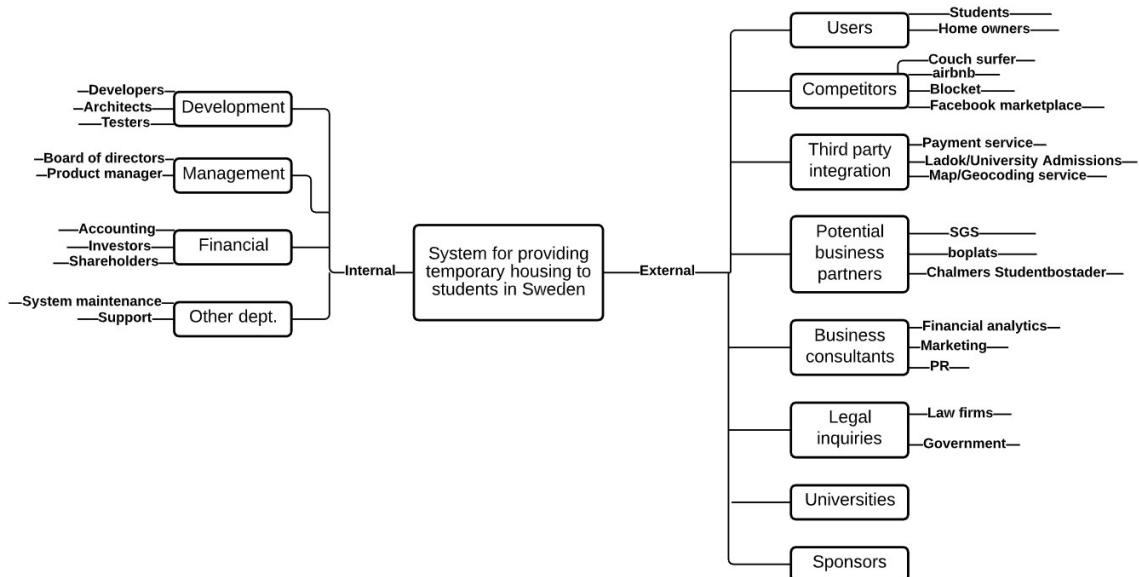


Figure A.2: Stakeholder map

A rating, or prioritization, of stakeholders was also made based on several factors such as financial and emotional interest, but also how much influence one has over the system and/or the system has over them. The users, for example, has an indirect influence on the system to potentially change and add many features while

holding a high interest in its success, as they benefit from its existence. On the other hand, competitors hold very little power over the system, however, their existence may influence decisions regarding the system in order to stay ahead of them in the market and is therefore rated low but still needs to be monitored.

Additionally, it is important to note that the highest prioritized stakeholders are essential for the survival of the system and should therefore be managed very closely and have an high involvement in the system. An illustration of these rankings can be found in the stakeholder analysis chart, Figure A.3.

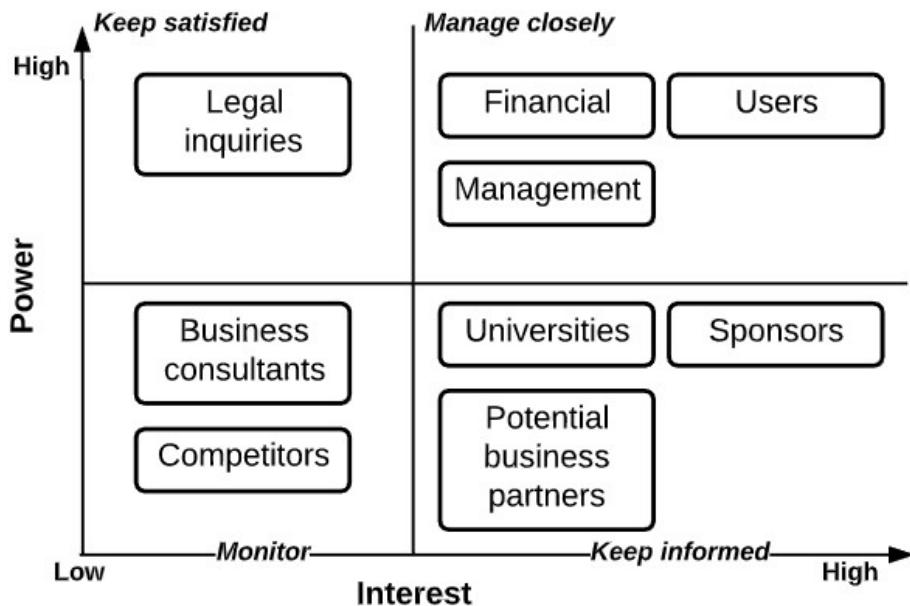


Figure A.3: Stakeholder power and sentiment chart

The majority of the internal stakeholders hold a high power and high interest in the system as they have both a high influence over the system and benefit from its success. For this reason, only a few were listed in the chart. As for the external stakeholders, all but the third party integration are listed in the chart. This was mostly due to fact that there are always substitutes for different integration systems which means that there are any number of replacement systems or other methods that can be implemented, thus the overall power and interest of individual third party systems are almost non-existent.

A3. Business case

This section states the system's business goals and discusses intended workflows and its associated tasks.

A3.1. Business goals

To help students find affordable temporary and acute housing easily, there are three main problems that our system aims to solve. The first problem is creating an easy to use application for home owners and both international and local students. The second one is the core of the system, which is the problem of creating a platform for students and home owners to connect and giving them the opportunity to rent and rent out accommodation, as such platform does not currently exist. And finally, the third one is making sure that the users of the system can offer and accept their lodging services safely while avoiding to run into legal issues and lower their risk of running into a potential fraud.

To provide an overview, the system's specific business goals are shown in Table A.1. A priority score out of 100 was also assigned to represent the importance of each business goals to the primary stakeholders.

The priorities were carefully assigned after successful elicitation with the user groups. While user satisfaction and usability, components of the first business goal, are quality factors that should be strived for in most web applications, the user groups argued the feeling of being safe, both legally and emotionally, in their paid-for accommodation takes precedence.

Goals for the system	Solution vision	Related requirement	Priority
(BG1) Expand user market by providing a multilingual system that is usable by the people who are less tech-oriented, such as the elderly.	Optimize the way of interaction, reduce unnecessary scene(page) conversion. Support multiple languages and allow users to easily change languages. The system should be easy to use and adaptable to its user while providing explanations and guidelines to parts that may be found confusing.	UR.01, UR.02, UR.03, UR.04, IO.04	20/100
(BG2) Enable communication between students and homeowners and connect them on a common platform to be able to rent and rent out accommodation.	Provide communication channels with an easy point of access. Allow home owners to post and edit accommodation while students to search, browse, and apply. Present static information about the accommodation before communication as to meet expectations.	CR.01, CR.03, CR.04, CR.05, CR.06	45/100
(BG3) Reduce risk of fraud, protect integrity of users, and ensure payment security	Verify home owners and students to keep them both safe and mitigate potential frauds. Home owners and students have their identities, status, and accommodations verified. Secure payment transactions through trusted third party services are offered between home owners and students to avoid fraud.	CR.02, CR.07, CR.08, CR.09, SR.01, SR.02, SR.03, SR.04, SR.05, SR.06	35/100

Table A.1: Business Goals

A3.2. Flows

The system shall support two flows: Finding and offering accommodation. Finding accommodation caters to the needs of the Students while the Homeowner offers accommodation via the system. For both the flows, some steps can be omitted or offered alternatives, while others can be repeated several times.

The logical flow is carried out entirely digitally. Table A.2 showcases the process of finding accommodation from a student perspective, with Column 1 detailing the logical steps for the same. Column 2 shows the related tasks and subtasks for each step in the flow. Table A.3 details the steps undertaken by the homeowner to display their accommodation in the system. Chapter B shows the details.

Steps in finding accommodation for the student	Tasks and Subtasks
1. Create a Student account	W1.1
2. Student verification	W1.3
3. Search for available room(s)	W2.1
4. Browse through the list of room(s)	W2.2, W2.3
5. Contact the owner(s) of the room(s) the student is interested in	W3.1
6. Make payment	W3.2

Table A.2: Logical Flow For Students

Steps in offering accommodation for the home owner	Tasks and Subtasks
1. Create an account as an Owner	W1.1
2. Post a room with detail information about the room condition	W2.4
3. Home owner verification	W1.3
4. Select a student to offer the room to based on home owner preference	W3.3

Table A.3: Logical Flow for Home Owners

A3.3. Business Constraints

Business constraints are any factor that affect the profitability of a company or enterprise. Increasing profit margins requires eliminating or reducing business restrictions. In the system, we consider the following five constraints:

Location constraints: Due to limited sampling for elicitation techniques, the sampling will be limited to the city of Gothenburg and therefore so will the system.

Time constraints: The pandemic is not over yet, and it is foreseeable that there will not be much progress in the coming period. People's willingness to travel and short-term lease will greatly decrease, which will affect the implementation of the system.

Cooperation constraints: It is difficult to obtain third-party cooperation, especially the government agencies, like Ladok and student admissions.

Financial constraints: Not enough money to hire the senior staffs and purchase servers that support high concurrency.

Regulations: (1) the Swedish Tenancy Act (Hyreslagen, Chapter 12, Land Code of Sweden, 1970:994) and (2) the Swedish Law on Private Subletting (Privatuthyrningslagen, 2012:978).

A4. Core functionality

The system has two main user stakeholders, both with a separate objective. While the student wants to rent an accommodation, the home owner wants to post the accommodation and have it rented out. This creates two separate logic flows described in section A3.2.. In the use case diagram, Figure A.4, the logic flows are represented along with the addition of a few supplementary extended and included use case scenarios as well as minor tasks to completely represent the ideal core functionality of the system.

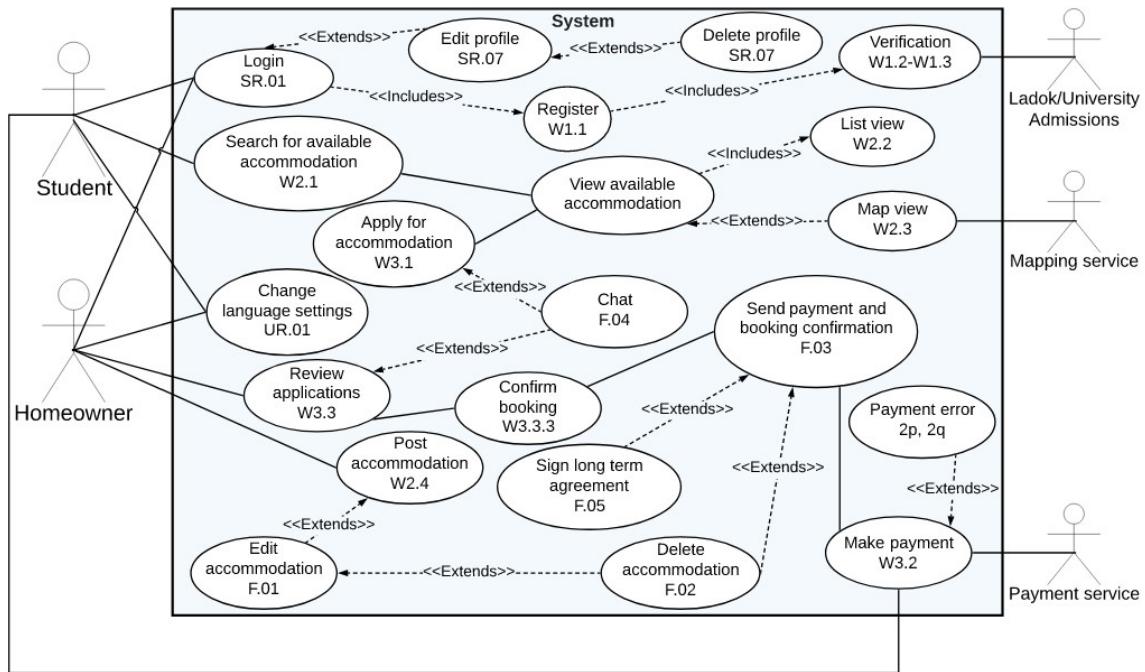


Figure A.4: Use-case diagram

The use case diagram, Figure A.4, shows several UML use cases and visualizes the previous flows described in section A3.2.. The box represents the proposed system and the corresponding tasks and user stories are represented in the use case bubbles.

As a result of the use-case diagram, Table A.4, describes the core requirements of the system. The ID of the core functionality is given in the first column, while the second and third describe its category and description. Finally, the fourth and last column contains ID references to the related tasks and/or other requirements that must be supported to verify the core functionality in question.

ID	Category	Description	Related requirement
CR.01	Account management	The system must allow for users to sign up, edit, and delete accounts. This would allow users to access user-oriented functions while also complying to GDPR laws.	W1.1, SR.07
CR.02	User verification	The system must be able to verify identities, student status, and accommodations. This is done to guarantee a certain sense of security when renting or renting out accommodations.	W1.2, W1.3
CR.03	Accommodation management	The system must allow users to post, view, and delete accommodation listings.	W2.4, F.01, F.02
CR.04	Searching	The system must allow users to search with filters and browse through accommodation listings.	W2.1, W2.2
CR.05	Location browsing	The system must allow users to browse accommodation listings by location on a map through a third party service. This would allow users with little prior geographical knowledge of Gothenburg to visualize the distance to important locations such as school.	W2.3
CR.06	Communication	The system must allow for home owners and students to contact each other through a provided chat system. This is not only done to allow users to discuss rental agreements, but also provide a sense of comfort and security for the users.	W3.1, F.04
CR.07	Confirmation	The system must send out booking and payment confirmations.	F.03
CR.08	Contract	The system must allow for home owners and students to reach either a short term agreement or sign a long term agreement with an underlying contract.	W3.3, F.05, F.06
CR.09	Payment	The system must allow users to pay and receive payment using a secure and trusted third party service provider.	W3.2

Table A.4: Core Requirements

A5. Critical attributes and constraints

The importance of quality factors varies in many projects, while some are more essential, others may be deemed low-pressure or unnecessary. Therefore, in order to be time and funding efficient, the quality factors must be evaluated and their degree of importance assessed. The output of this process may be found in the quality grid represented in Figure A.5 where a combination of McCall & Matsumoto's and the standard ISO9126 lists of quality factors were used.

	Critical	Important	As usual	Unimportant	Ignore
Operation	-	-	-	-	-
Integrity/security	1				
Correctness			X		
Compliance			X		
Reliability/availability		4			
Usability	2				
Efficiency			X		
Revision	-	-	-	-	-
Maintainability			X		
Test-ability			X		
Flex-ability			X		
Transition	-	-	-	-	-
Portability			X		
Interoperability	3				
Re-usability			X		
Install-ability				X	

Table A.5: Quality grid for the proposed system.

Concerns:

1. Integrity/Security: The service involves not only payment transactions, but also the handling of sensitive data for all involved parties. The system needs to provide assurance that money won't be lost due to some physical disturbances and/or malicious access attempts, etc. Additionally, the system process a lot of important personal information on students and home owners such as personal identification documents and student info. The system must provide adequate security in accordance with applicable data protection laws and regulation. See section C1.1. for more information on security requirements.

2. Usability: Our customers include many incoming international students and also elderly home owners. They may not know the process of renting or renting out an accommodation. Simple and convenient operation can keep both parties from getting into trouble or giving rise to confusion. Additionally, adding several popular language options, based on where the majority of international students who attend universities in Gothenburg come from, allows the system to remain accessible to a larger amount of students. See section C1.2. for more information on usability requirements.

3. Interoperability: Our system has an internal payment function. Support more third-party payment platforms, users may give priority to using our products. Additionally, users have multiple account systems. They prioritize smooth integration with existing products. Some aspects of interoperability such as integration with spreadsheet, etc. is considered unimportant as a built-in information form is used. See section C1.3. for more information on security requirements.

4. Reliability/availability: We aim to help students find affordable temporary and acute housing. Hard to meet user needs if the system were to fail. And as a system involved in money transactions, if there are problems, it is easy to cause economic losses for users. Too frequent, we may even face commercial litigation.

A5.1. Key attributes

In this section, in order to tackle the poor clarity problem, we use Planguage which was developed to describe the important attributes in detail.

TAG: Security

GIST: Security of the payment transactions between students, homeowners and third parties.

SCALE: The payment information should be encrypted by using the current encryption algorithm for insurance.

METER: Attempts to get payment information through obtained messages on 1000 payment transactions section.

MUST: 100% of the time should be encrypted.

TAG: Usability

GIST: Usability for students and home owners to use the system by following the fewer steps.

METER: Find 5 users to use our system and calculate how many steps are needed to meet their own needs.

MUST: No more than 6 steps. Those 6 steps are Login/Sign up, Verification, Searching, Viewing an accommodation, Making an agreement and Confirmation.

WISH: No more than 5 steps.

TAG: ResponseTime

GIST: The time for system to display the search results when students search for a room.

SCALE: The Response time of a search.

METER: Measurement obtained from 1000 searches during testing.

MUST: No more than 2 seconds 100% of the time

WISH: No more than 1 second 100% of the time

TAG: Interoperability

GIST: The interoperability about supporting third-party payment platforms

SCALE: The success rate of redirecting third-party payment software

METER: 100,000 times of jumping to third-party payment software.

MUST: Success more than 95% of the times.

PLAN: Success more than 99% of the times.

WISH: Success more than 99.99% of the times.

TAG: Reliability/availability

GIST: The availability of the system, i.e., uptime.

SCALE: The average system availability.

METER: Measurements obtained from 100 hours of usage during testing.

MUST: More than 95% of the time.

PLAN: More than 99% of the time.

WISH: 100% of the time.

A5.2. Technical Constraints

It is difficult to adapt to the terminals of all users. For example, mobile phones have Android and Apple systems. Among them, the scale of the screen size of mobile phones using the Android system varies greatly. It is a very challenging task to achieve good display effects on each mobile phone. In addition, on the computer, there are Windows, Apple and Linux. Different users prefer different browsers, and the browser version of users may not be up to date. Therefore, it is also hard to adapt the system on the website.

It is difficult to provide services to some remote places because there is no good network signal and GPS positioning.

B

User requirements specification

B1. Tasks to support

This chapter describes all the user tasks that must be supported to some degree, and all problems that need to be mitigated by the system. For convenience, each task is grouped under different work areas where the user profile, purpose, and environment of performing the task may vary.

A few tasks are supplemented as product features as they were too trivial or too product sided to be listed as a task. The relating or supporting feature requirements can be found referenced under the tasks or in section B2. Supplemental requirements and prerequisite requirements, if any, that are needed to accomplish the tasks are either mentioned in the example solution column of the related sub-task or mentioned in the task overview.

Work area 1: Account management

This work area comprises of every task involving any form of account creation and editing.

User profile: Students. Varying levels of IT experience, but expected to have basic experience in browsing simple web pages. Might have limited knowledge of the local housing markets.

User profile: Home owners. Expected to have basic experience in browsing simple web pages.

Environment: Account

W1.1 Create an account

This task shows the process of creating an account in our system. The user profile along with its data can be edited and deleted after creation, for more information see integrity requirement SR.07 in section C1.1..

Users: Students or Home owners.

Start: User initiates registration process.

End: When user has received confirmation of registration.

Frequency: Medium, once per student.

Difficult: Never.

Subtasks and variants:	Example solutions:	Data:
1. Select account type		
2. Record personal information		B3.2.1.
3. Confirm Registration		

W1.2 Verify student

This task will verify that a registered student is enrolled as a student and confirm information filled in previous task.

Users: Student.

Start: User accesses the verification page.

End: The student's status is updated to verified.

Frequency: Medium, at least once per student account per semester.

Difficult: Unofficial documentation increases complexity in verification.

Subtasks and variants:	Example solutions:	Data:
1. Open verification page	Open up a new page where student can sign in through Ladok/Universityadmissions.se. See integration requirement IO.03 in section C1.3..	
2. Sign in with student account acquired from university		
2p. Problem: Could not verify details	Show an error message and where to get further help. See integration requirement IO.05 in section C1.3..	
3. Receive confirmation of verification	System update the account's status to verified and set study location	B3.2.2.

W1.3 Verify home owner

This task verifies that the accommodation offered by the home owner belongs to them. This is accomplished by the system crosschecking the owner's uploaded documents with government or third-party authentication website. See requirement IO.03 in section C1.3. for more information regarding the external system integration.

Users: Home owner.

Start: User accesses the verification page.

End: The home owner's status is updated to verified.

Frequency: Medium. Verification is done every time a home owner account is created. Cannot exceed maximum active users.

Difficult: Unofficial documentation increases complexity in verification.

Subtasks and variants:	Example solutions:	Data:
1. Provide documents proving ownership		
1p. Problem: Unable to verify provided documents	Show user an error message and where to get further help. See integration requirement IO.05 in section C1.3..	
2. Receive confirmation of verification	System updates the home owner's status to verified	B3.2.3.

Work area 2: Accommodation management

This work area comprises every task involving any form of searching for, creating or editing any accommodation.

User profile: Home owners and students.

Environment: Accommodation.

W2.1 Search for available accommodation

This task search through the available accommodation and present student with a list based on study location and search terms.

Users: Students.

Start: User wants to find accommodation.

End: List of available accommodation is updated.

Frequency: High.

Difficult: Never.

Subtasks and variants:	Example solutions:	Data:
1. Enter search criteria		B3.2.4.
2. Perform search	System search database based verified study location and specified search criteria and updates the list of rooms	
2p. Problem: No search criteria was entered	Show all available rooms based only on study location	
2q. Problem: No results	Display a message stating that no results were found	

W2.2 Browse list of accommodations

This task describes how the user can browse the list of rooms to find suitable accommodation.

Users: Student.

Start: After search credentials have been entered and searched for in the database.

End: When a suitable accommodation has been found.

Frequency: High.
Difficult: Never.

Subtasks and variants:	Example solutions:	Data:
1. Filter or sort the list		B3.2.4.
2. Scroll through the list of accommodation		
2a. Show next page		
3. Interact with listing for additional information on specific accommodation		

W2.3 Browse map of accommodation

Similar to task W2.2, however this task involves browsing the available accommodations by location on the map rather than a list.

Users: Students.
Start: After search credentials have been entered and searched for in the database.
End: When a suitable accommodation has been found.
Frequency: High.
Difficult: Never.

Subtasks and variants:	Example solutions:	Data:
1. User changes to map view	See integration requirement IO.02 in section C1.3..	
1b. Problem: Third party service not available	Usually updating the page or entering search credentials again solves it. See integration requirement IO.05 in section C1.3..	
2. Scroll around map as city of interest is shown with available options		

W2.4 Post accommodation(s) with accompanying details

This task creates an association between homeowner and details of their offered accommodation(s). Accommodation with its details can be edited or deleted after owner has posted, for more information, see feature requirements F.01 and F.02 in section B2.. Due to Swedish law, it's important that owners are informed of what they can post on ads. Payment amount can be chosen at this step.

Users: Home Owner.
Start: User accesses page for posting accommodation
End: When the details of accommodation(s) are successfully posted on our system.

Frequency: Medium, at least once per home owner.
Difficult: Never.

Subtasks and variants:	Example solutions:	Data:
1. Enter accommodation information		B3.2.4.
1a. There is an option of payment		
2. Post listing of accommodation	Accommodation with its details is connected with home owner account in our system	
Problem: System filters through the 2p. listing, finds phrases which are not allowed or confusing	Prompt home owner to fix	
2a. Home owner can post another accommodation		

Work area 3: Communication

This work area comprises every task involving any form of communication between parties.

User profile: Both students and home owners. Could assume some experience with the system at this point.

Environment: Communication

Prerequisite: SR.01, SR.04

W3.1 Contact home owner

This task creates an automated message to the home owner from the student account stating the times and dates the student would like to request a booking while leaving an open conversation log, a chat window, and room for possible changes.

Users: Students.

Start: After a suitable accommodation has been found.

End: When a message has been sent to the home owner leaving an open conversation log.

Frequency: High.

Difficult: Never.

Subtasks and variants:	Example solutions:	Data:
1. Enter dates/times		
1p. Problem: Date already taken	Try different dates	
2. Send booking request	A message is sent to the home owner with the dates and times requested, opening a conversation dialog. See feature requirement F.04 in section B2..	B3.2.5.

W3.2 Make payment

After confirming a booking with a home owner, the student will be prompted to pay the agreed amount with the help of a third party payment service provider.

Users: Students.

Start: After receiving a booking confirmation from the home owner.

End: When payment ha been confirmed-

Frequency: Low, at least once per listing.

Difficult: Never.

Subtasks and variants:	Example solutions:	Data:
1. Open payment page	User will be taken to another window where payment info will be entered. See integration requirement IO.01 in section C1.3..	
2. Enter payment details		
2p. Problem: Third party service not available	Prompt user to try again in a while. See integration requirement IO.05 in section C1.3..	
2q. Problem: Payment declined or could not be processed	Contact bank	
3. Receive payment confirmation	System show user confirmation of payment and send a receipt via email. See feature requirement F.03 in section B2..	B3.2.5.

W3.3 Select student for communication

This task is for selecting the students interested in the offered accommodation and potentially offer them a contract. There can be multiple students interested in the same accommodation. For supplemental requirements regarding the contract, see feature requirements F.05 and F.06 in section B2..

Users: Home Owner

Start: Once the accommodation is available and students have applied for it.

- End:** After the home owner ends the communication or after an agreement is signed.
- Frequency:** Low, at least once per listing created up to same amount of existing listings per home owner.
- Difficult:** Never.

Subtasks and variants:	Example solutions:	Data:
1. List all students	System lists all students interested for the selected accommodation	
1a. Problem: No students listed	No further tasks performed	
2. Open chat window	System opens a chat window on clicking the student name. See feature requirement F.04 in section B2.	
2a. Problem: Student account no longer exists	Redirect to error page with custom error code for student not found	
3. Proceed to signing agreement	Home owner can proceed with the selected student to sign an agreement	B3.2.5.

B2. Feature requirements

In order to supplement the previous section on tasks to support, a separate table, Table B.1, in this section describes the smaller or more trivial product functions, at least from a user perspective, as feature requirements. This was done in order to give the system a better sense of completeness by specifying its functionality. In the table, each requirement is given along with a column to relate which task it completes and data entity involved, if any.

ID	Feature requirement	Related main task	Data
F.01	The system shall allow users to edit accommodation details after it has been posted.	W2.4	B3.2.4.
F.02	The system shall allow users to delete a accommodation after it has been posted.	W2.4	B3.2.4.
F.03	The system shall send payment and booking confirmation to affected users after a successful booking.	W3.2	B3.2.5.
F.04	The system shall allow users to communicate with each other through a chat system.	W3.1	B3.2.1.
F.05	The system shall allow users to enter a long term agreement with an underlying contract where the contract details can be edited.	W3.3	B3.2.5.
F.06	The system shall allow users to cancel a signed agreement, in accordance to the contract details, if the pre-defined terms and conditions were not fulfilled.	W3.3	B3.2.5.

Table B.1: Feature requirements

B3. Data requirements

This section describes the data recorded by the system. The following section shows an Entity-Relationship diagram followed by data dictionaries for each entity.

B3.1. Entity Relationship Diagram

The Entity Relationship (ER) diagram is shown in Figure B.1. The principal entities, displayed in boxes, are classes which can have several objects. The system must support multiple objects of each entity, for example multiple students and multiple home owners. The user entity contains common fields from both student and home owner, and merges after the user has selected the account type. Relationships between the entities are denoted using Crow's Foot notation. Students can apply for multiple accommodations and Home owners can post multiple accommodations. Each contract can bind to only one accommodation and likewise only one accom-

modation can be bounded by a contract.

Only domain entities are displayed in the ER diagram. Technical entities such as Administrator and Support are omitted for better clarity. Each entity has several attributes, although only a subset of the attributes are present in the diagram. A comprehensive list of the attributes are present in the data dictionary section B2.2.

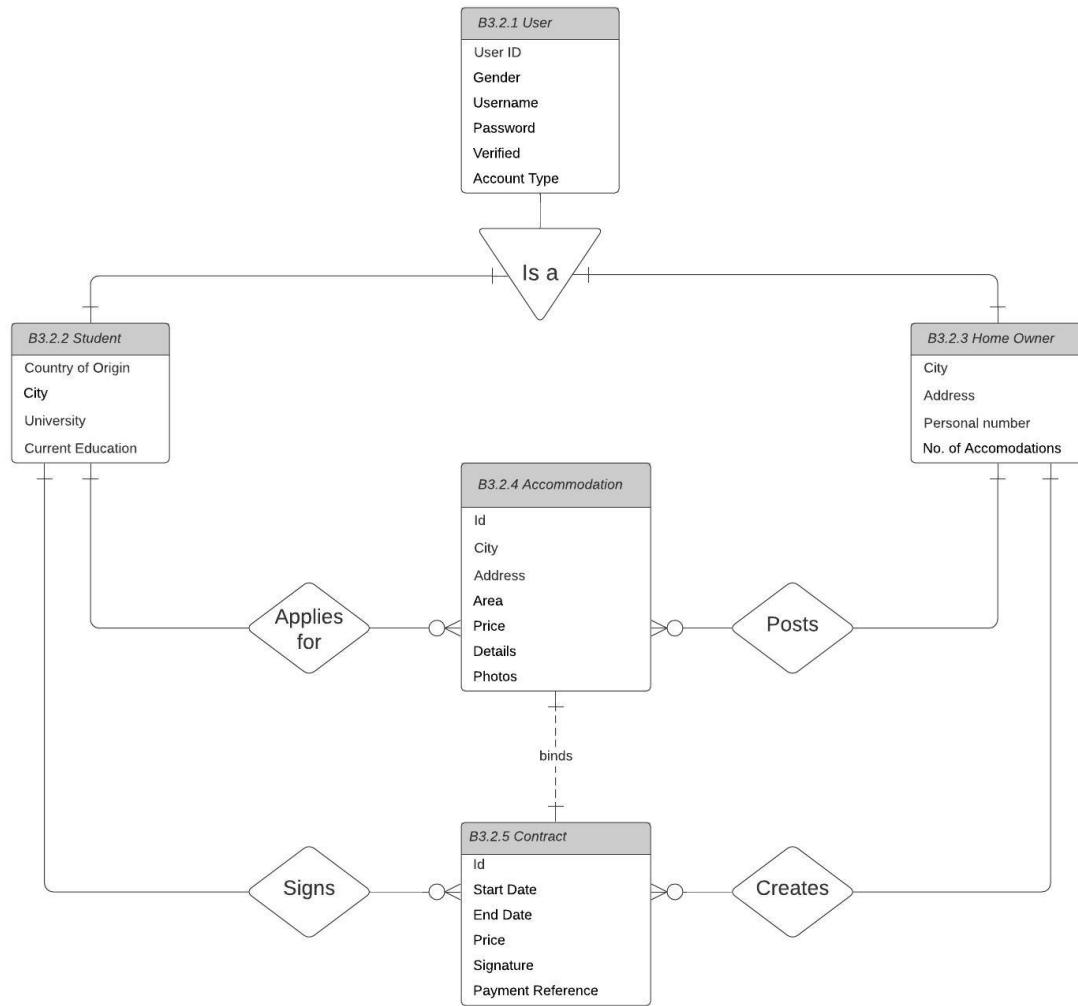


Figure B.1: Entity-Relationship Diagram

B3.2. Data Dictionary

This section displays the full list of entities present in the ER diagram. The tables below list the attribute according to name, datatype, description, and an example.

B3.2.1. User

The user attributes are common attributes shared between the Student and Home Owner. The system keeps track of all modifications done to the attributes and the time when it was modified. Besides ID, verified and age, all other attributes are user specified. The System generates a unique Identifier for each object generated. The password is hashed using the SHA-256 algorithm for increased security; the original password of the user is never stored in the system (see security requirement SR.02 in section C1.1.). The Age is automatically calculated by the system after the Date of Birth has been entered.

Attribute Name	Datatype	Description	Example
1. ID	String	Unique Identifier for each user	STU00012
2. Gender	String	The User's gender, choose between Male, Female and other	Male
3. Age	Integer	The User's Age in numbers	22
4. Username	String	The User's name. This is also the display name	Enrico Maxwell
5. Password	String	Password for the user, stored as a 256-bit hash value	qiyh4XPJGs...
6. Verified	Boolean	States if the user is verified by the system or not	1
7. Account Type	String	Either Student or Home Owner	Student
8. Date of Birth	Date	The User's date of birth, stored as DDMMYYYY	19041992
9. Payment Reference	media	List of Payments made corresponding to contracts	

B3.2.2. Student

Student inherits attributes from the User. Student object is created only if the Account Type attribute has the value 'Student'. All attributes are user specified. Universities are displayed using a drop-down list and are fetched by a third-party integration.

Attribute Name	Datatype	Description	Example
1. Country of Origin	String	Student's Birth Country	Israel
2. City	String	Student's Birth City	Mumbai
3. University	String	The University where the Student is studying at	University of Gothenburg
4. Current Education	String	Current Education Level of the Student	Masters of Science
5. Course	String	Student's current Stream of Education	Game Design and Technology

B3.2.3. Home Owner

Home Owner inherits attributes from the User. Home Owner object is created only if the Account Type attribute has the value "Home Owner". The No of Accommodations increases dynamically per accommodation posted by the Home Owner. The Swedish Personal Identity Number of the Home Owner is a unique identifier of the Home Owner and serves as a means of verification via external integration as shown in C4.2. Accommodations contains a List of Accommodation Ids corresponding to the ones posted by the Home Owner. It is dynamically generated and serves as Foreign Keys that links to the Accommodation List.

Attribute Name	Datatype	Description	Example
1. City	String	The Home Owner's current City of Residence	Uppsala
2. Address	String	The Home Owner's current Address	Fridhemsgatan 46, Gothenburg
3. Personal Number	String	Swedish Personal Identity Number	950621-921x
4. Accommodations	String	List of Accommodation Id's posted by the user	[ACC0012, ACC013]
5. No of Accommodations	Integer	Total Number of Accommodations posted by the Owner	2

B3.2.4. Accommodation

An Accommodation is a place offered by the Home Owner for Students to live in Sweden. The Available attribute is False only if it has been binded to a contract. The Photos attribute accepts .jpg, .png, .jpeg, .gif and .bmp media formats.

Attribute Name	Datatype	Description	Example
1. Id	String	Unique Identifier for the Accommodation	ACC0012
2. Type	String	Shared or Single housing	Shared
3. City	String	Location of the Accommodation	Uppsala
4. Address	String	Complete Address including pin-code	Fridhemsgatan 46, 451 43
5. Area	Integer	Total Carpet Area occupied in Square meters.	25
6. Price	Float	Proposed price in Swedish Kronas	5420
7. Details	Longtext	General description of the Accommodation along any facilities offered	
8. Furnished	Boolean	Is the Accommodation Furnished or not	0
9. Deposit	Float	Deposit amount, if any, in Swedish Kronas	10000
10. Available	Boolean	Check if the Accommodation is still available	1
11. Available From	Date	Date the Accommodation is available from	23102020
12. Photos	media	Images of the Accommodation.	

B3.2.5. Contract

A contract is a legally binding agreement stating the terms and conditions for the chosen Accommodation. It is created by the Home Owner and signed by the Student once the Accommodation is finalized.

The agreement ends when the agreement period expires or cancelled by either party if the pre-defined terms and conditions are not met. The agreement can be cancelled by the student if the other rental arrangement and the state of the accommodation did not match with the information posted on the proposed system. On the other hand, the homeowner can cancel the contract if the student did not follow the guidelines that they agreed on in the chat and/or the information posted about the accommodation.

Attribute Name	Datatype	Description	Example
1. Id	String	Unique Identifier for the Contract	CON123
2. Start Date	Date	Starting Date of the Contract	19102020
3. End Date	Date	Expiry Date of the Contract	20102025
4. Price	Float	Finalized Accommodation price in Swedish Kronas	6420
5. Payment Reference	media	Attachment to the payment made	payment.pdf
6. Signature	Blob	A digital Signature of the Student	

B4. Proposed prioritization

Software development projects often have many requirements, coupled with tight development time. Therefore, it is necessary to prioritize software requirements because it is not possible to accomplish these requirements at the same time. In this section, we use two different techniques to determine which requirements need to be implemented first and which can be deferred to a later release.

B4.1. \$100-test

This method consists of three steps. First, we aim at two main stakeholders; students and homeowners. Then a fictional \$100 is given to each stakeholder. Third, we then let them distribute (or "spend") the money over the requirements to reflect the assessment of a criteria from that stakeholder's point of view. The results of this can be seen in table B.2 and the table which is been ranked can be seen in table B.3

Requirement	Students	Home owners	Average
CR.01 Account management	5	5	5
CR.02 User verification	20	15	17,5
CR.03 Accommodation management	15	20	17,5
CR.04 Searching	15	10	12,5
CR.05 Location browsing	5	5	5
CR.06 Communication	15	10	12,5
CR.07 Confirmation	5	10	7,5
CR.08 Contract	10	15	12,5
CR.09 Payment	10	10	10

Table B.2: Each stakeholder's distribution in the \$100 test

Score	Students	Home owners
5	CR.01, CR.05, CR.07	CR.01, CR.05
10	CR.08, CR.09	CR.04, CR.06, CR.07, CR.09
15	CR.03, CR.04, CR.06	CR.02, CR.08
20	CR.02	CR.03

Table B.3: Each stakeholder's distribution in the \$100 test (ranked)

B4.2. Ranking (sorting)

Ranking is a relatively simple prioritization technique. When using this technique, we rank the most important requirements at the first place and the least important requirements at the end.

The ranking technique is mainly based on the answers to the following three questions:

1. How important is this requirements for stakeholders?
2. How important is this requirements compared to other requirements?
3. Is there any other requirements related to this requirements?

By answering above three questions, we can get the ranking list: The list of ranking requirements are:

1. CR.02 User Verification
2. CR.03 Accommodation management
3. CR.08 Contract
4. CR.01 Account management
5. CR.04 Searching
6. CR.06 Communication
7. CR.07 Confirmation
8. CR.09 Payment
9. CR.05 Location browsing

B4.3. Final prioritization

The final model is merged according to the following rules:

1. As for 100 dollar test, the average score of each requirements is used as the final score of the requirement
2. In the ranking requirement, scores ranging from 15-7 points are given to the requirements of ranking first to ninth.
3. Add the scores of the above two items together, the requirement which have the most scores will be ranked first, and the least will be ranked last

So now we have the final prioritization:

Requirement	\$100 test (avg.)	Ranking	Total	Final rank
CR.02 User Verification	17,5	15	32,5	1
CR.03 Accommodation management	17,5	14	31,5	2
CR.08 Contract	12,5	13	25,5	3
CR.04 Searching	12,5	11	23,5	4
CR.06 Communication	12,5	10	22,5	5
CR.09 Payment	10	8	18	6
CR.01 Account management	5	12	17	7
CR.07 Confirmation	7,5	9	16,5	8
CR.05 Location browsing	5	7	12	9

Table B.4: Final prioritization based on the combined score of the \$100 test and ranking

The purpose of prioritization is to realize the high-ranking requirements to solve the user's problem. The prioritization criteria vary depending on the overall workflow of the system. We use prioritization depending on the phase of the development of the system, which means, for example, the student's main target is to a room, and the homeowner's main target is to rent out the room. Accommodation management could be the highest priority for both users, but there are some other constraints or

functions that need to be fulfilled to satisfy the user's needs. First, users need to register to the system, but the system requires an authorized document from users to create an account and access it, so user verification gets a higher prioritization rank than accommodation management.

User verification is one of the most important functional requirements based on the two prioritization methods that we have discussed above. It is because both the student and the homeowner should have proper documents to create an account and access the systems services. The system allows only verified users so that both users trust each other when they start communicating. Accommodation management comes next to provide some main functions of the system, which are discussed in Work area 2. Then, the contract functional requirements get the next ranking to allow users to sign an agreement after handling the accommodation management requirements.

B4.4. Next release

Based on the prioritization discussed above, we divided the core functionality of this project into three releases. Refer to the traceability matrix, C.2, in section C2. for dependencies of the core requirements.

- CR.02 User Verification
- CR.03 Accommodation management
- CR.08 Contract
- CR.04 Searching
- B3.2.1. User
- B3.2.2. Student
- B3.2.3. Homeowner
- B3.2.4. Accommodation
- B3.2.5. Contract

B4.5. Second release

The second release extends based on the first and adds more core functionality and other requirements of the proposed system. This release would finalize a minimal viable product that would be ready to be sent out to the customer.

- CR.06 Communication
- CR.09 Payment
- CR.01 Account management
- CR.07 Confirmation

B4.6. Future releases

- CR.05 Location browsing

C

System Requirements

This section discusses the system-level requirements that describe the qualities and behavior of the system. The system should fulfill requirements to satisfy stakeholder needs and requirements. It also includes a traceability matrix, the UI prototypes, and acceptance tests.

C1. Other system requirements

This section details other requirements not discussed in any of the previous sections. This includes user documentation, system integration, security requirements and user installation.

C1.1. Integrity/Security

Security is a critical quality factor as reasoned in section A5.1. The system thus shall contain security features providing for authentication, authorization, privacy, availability and data confidentiality. The security requirements are detailed in table C.1.

ID	Security requirement	Example Solution
SR.01	The system shall require user authentication using credentials registered.	Only account holders can use the system to store data.
SR.02	The system shall encrypt and store user passwords.	Refer to section B3.2.1.
SR.03	The system will issue a time-out and log out the user after seven minutes of user inactivity.	
SR.04	The system shall allow only verified users to sign a contract.	This includes both verified Home Owner and verified Student.
SR.05	The system shall allow personal communication details such as email-id and phone number to be exposed to other users based on user settings.	The personal details will be hidden by default.
SR.06	The system shall include a database management system (DBMS) to handle concurrency issues such as read/write errors.	DBMS also prevents data loss in case of access failure (due to no internet or power cut) during a read/write cycle.
SR.07	The system shall comply with the GDPR laws.	Options for editing profile, deleting profile, and requesting information regarding what/how personal data is being stored/used should be added.

Table C.1: Security requirements

C1.2. Usability

Usability requirements include user accessibility and adaptability while using the system's user interface. The system's user interface (as displayed in figure C.3 and C.4) should cater to user specific needs which are detailed in table C.2. The main reasoning used behind most usability requirements is to expand the user market by using a easy to interact with and adaptable user interface that can be tailored to user need.

ID	Usability requirement	Example solution
UR.01	The user interface should be available in languages English, Swedish, French, and Chinese, with the users being able to easily switch between the options.	
UR.02	The user interface should include help documentation to guide user navigation.	The user interface includes an Frequently Asked Questions page (FAQ) for user assistance.
UR.03	The user interface should include provisions for changing basic design elements such as background color and font size.	Such attributes should be defined in the code at a global scope.
UR.04	The user interface shall be adaptive and responsive to the user device.	The code should detect the type of user device and show the UI accordingly.

Table C.2: Usability requirements

C1.3. System integration

The system supports integration with external systems such as third-party payment and user verification systems. The system should also include provisions for integration with other external systems according to user needs. Figure C.1 illustrates the interaction between actors and third party providers while Table C.3 defines the resulting requirements.

ID	Interoperability requirement	Example solution
IO.01	The system must support secure transaction between the users through multiple payment options.	Integrate popular third-party payment service options that are encrypted using the current encryption algorithm for insurance, such as paypal, Klarna, and Swish.
IO.02	The system must support a visual representation of the location of several accommodations.	Integrate third party mapping service such as Google maps.
IO.03	The system must support user verification.	Integrate third party student verification such as Ladok, studentadmission, or university specific login and user verification such as SPAR.
IO.04	The system will support further integration of more system as needed.	Will be based on high demand from users.
IO.05	The system must handle errors that may occur during data transactions with external systems.	Refer to integration guidelines of each third party service integrated.

Table C.3: Interoperability requirements

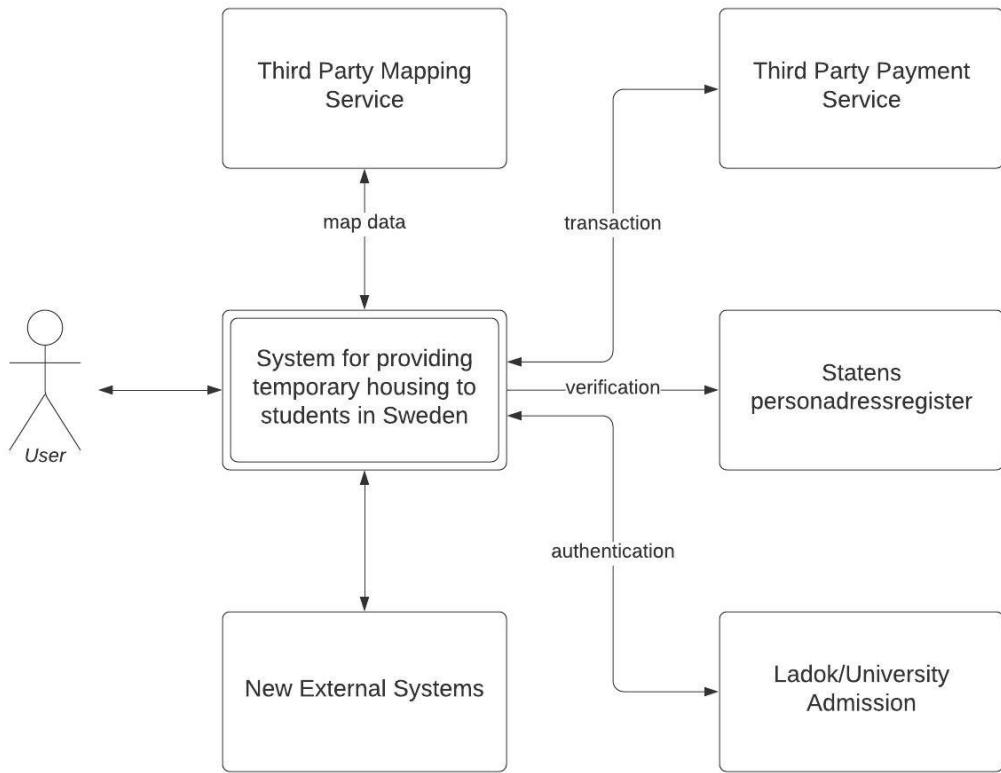


Figure C.1: External System Context Diagram

The third party mapping service is used to fetch a topological map to display geographical location of the accommodations listed. The map data isn't stored by the system and is requested only when accessing the accommodations listing page.

The third-party payment service provides payment gateway used for secure digital payment as described in task W3.2. The payment reference is stored as PDF file for the corresponding contract available for only the respective signatories. User verification can be done via integration with either Ladok or via University Admissions. Ladok is a student administration system used by all the universities in Sweden and has a register of every student studying in Sweden. An alternative to Ladok is University Admissions which also serves as a student database. Statens personadressregister (SPAR) or State Personal Address Register is a public register which contains information of all people residing in Sweden, local or foreign. The system utilizes two-way authentication with SPAR for the Home-owner using their personal number as seen in B3.2.3. Further access points shall be available for integration with new external systems as per customer request.

C2. Traceability matrix

The traceability matrix, Figure C.2, provides a final overview over the system's many goals and requirements, and gives a visual representation of their dependencies, i.e. what is required for each of them to be fulfilled. Best way to read the matrix is finding the requirement or goal of interest on the x-axis and the cells marked with an 'x' on the y-axis are the requirements needed to fulfill it.

The matrix, Figure C.2, proved to be too large to fit on to one page, therefore, a link to the matrix is included in the caption if a more detailed view was desired. Color coding in the matrix was used to improve the matrix's readability, the additional black squares have therefore no implied meaning other than for simple aesthetics. Additionally, to provide more context to the traceability matrix and avoid a constant back-and-forth between requirements, complementary tables C.4 and C.5 offer the ID of the requirements along with a quick description.

ID	Description
BG1	Business goal to expand user market.
BG2	Business goal to create an accommodation platform.
BG3	Business goal of reducing risk of fraud and protecting user privacy.
CR.01	Core requirement for account management.
CR.02	Core requirement for user verification.
CR.03	Core requirement for accommodation management.
CR.04	Core requirement for searching for accommodation.
CR.05	Core requirement for location browsing.
CR.06	Core requirement for user communication.
CR.07	Core requirement for booking and payment confirmation.
CR.08	Core requirement for contract.
CR.09	Core requirement for secure payment.
W1.1	Task for creating account.
W1.2	Task for verifying student.
W1.3	Task for verifying home owner.
W2.1	Task for searching for accommodation.
W2.2	Task for browsing list of accommodation.
W2.3	Task for browsing map of accommodation.
W2.4	Task for posting accommodation.
W3.1	Task for contacting home owner.
W3.2	Task for making a payment.
W3.3	Task for selecting a student.

Table C.4: Traceability ID and goal and requirement descriptions (1)

F.01	Feature requirement for editing an accommodation.
F.02	Feature requirement for deleting an accommodation.
F.03	Feature requirement for sending out payment and booking confirmation.
F.04	Feature requirement for the chat function.
F.05	Feature requirement for long term contracts.
F.06	Feature requirement for cancelling long term contracts.
B3.2.1.	Data requirement for the user.
B3.2.2.	Data requirement for the student.
B3.2.3.	Data requirement for the home owner.
B3.2.4.	Data requirement for the accommodation.
B3.2.5.	Data requirement for the contract.
SR.01	Security requirement for login.
SR.02	Security requirement for encrypting passwords.
SR.03	Security requirement for inactivity time outs.
SR.04	Security requirement for verified users.
SR.05	Security requirement for hiding personal details.
SR.06	Security requirement for handling concurrency issues.
SR.07	Security requirement for following GDPR regulations.
UR.01	Usability requirement for language availability.
UR.02	Usability requirement for offering a FAQ.
UR.03	Usability requirement for an interface with changeable design elements.
UR.04	Usability requirement for user device adaptive user interface.
IO.01	Interoperability requirement for integrating payment options.
IO.02	Interoperability requirement for integrating mapping service.
IO.03	Interoperability requirement for integrating user verification.
IO.04	Interoperability requirement for supporting further integration.
IO.05	Interoperability requirement for handling all external system integration errors.

Table C.5: Traceability ID and goal and requirement descriptions (2)

C2.. TRACEABILITY MATRIX

C. System Requirements

Figure C.2: Traceability matrix for the proposed system (link)

C3. UI prototype

A UI prototype was devised in order to outline the core functionality of the system. Diagram C.3 shows the flow of when user want to be verified in our system, i.e., get in touch with our home page, sign up/sign in and verification. CR.01 and CR.02 are the relevant requirements. In diagram C.4 it is illustrated how the UI would look when trying to find accommodation, i.e., searching, initiating a conversation and request stay with homeowners. These are related to CR.03, CR.04, and CR.05.

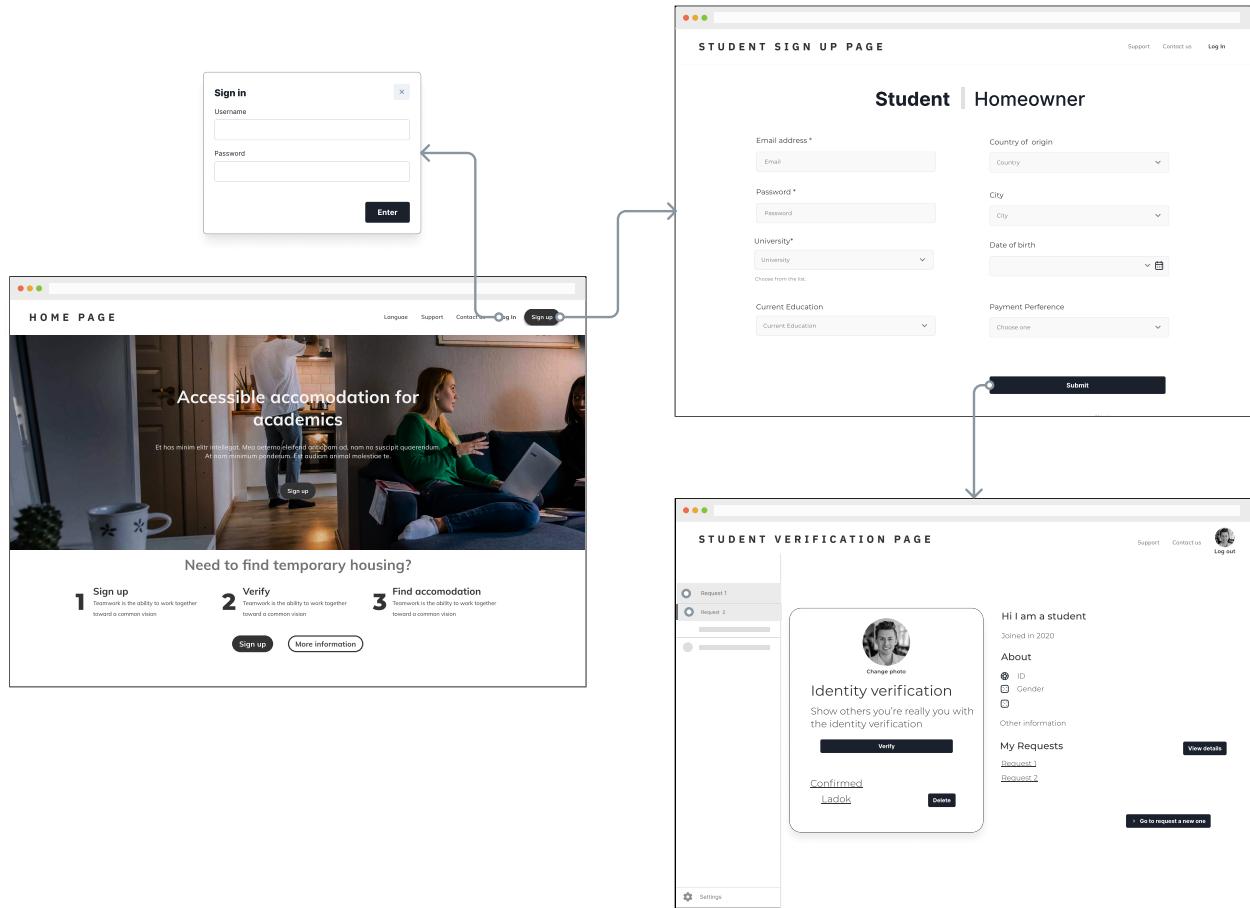


Figure C.3: User flow - The steps that need to be taken for a student to sign up and verify, with the home page as the entry point

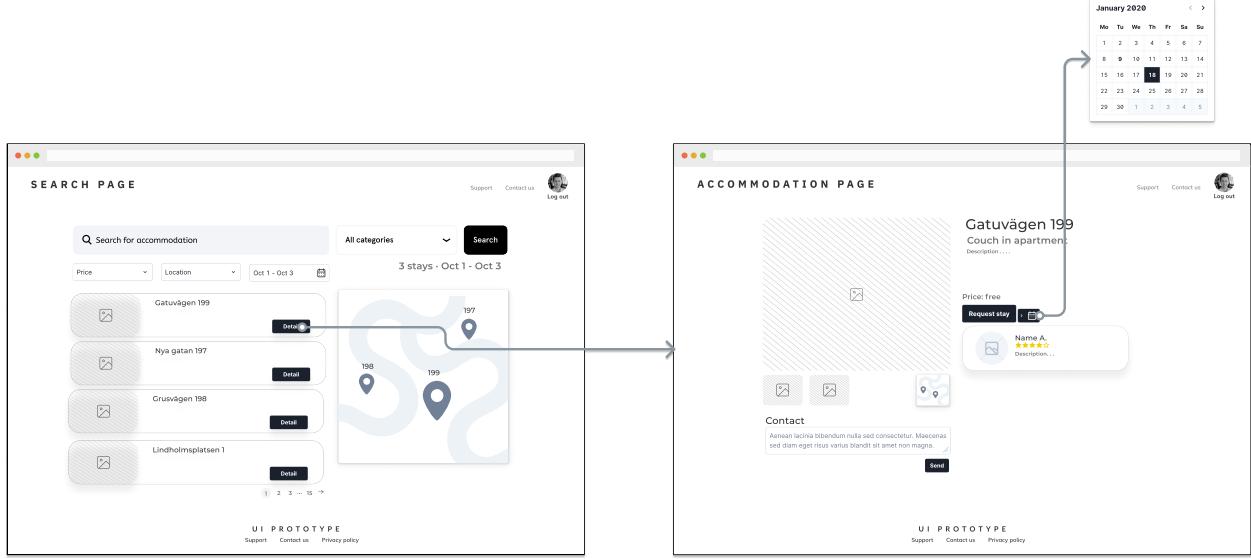


Figure C.4: User flow - Starting at the search page (left), a user can either click on a listing or on the map to be taken to a page showing more information about the selected accommodation (right)

C4. Acceptance tests

Acceptance Tests, which is the phase where the customer decides GO/No-GO for the product. In other words, acceptance tests are our way of being able to confirm that a user story was implemented correctly. Efforts of the development will be awarded by the customer by either accepting or rejecting the product developed.

The reason why we use acceptance tests is to ensure that the product is working in the way and whether the product is acceptable with critical functionality such as verification, payment integration, and booking confirmation in our system.

The below shows some user stories which related to core functionality and the criteria of the user story and whether the product is acceptable depends on whether the criteria is been fulfilled in the user story.

User Story 1:

As a student studying in Gothenburg, Sweden, I can be verified through Ladok/Universityadmissions, so I can rent an accommodation in this system.

Related requirements: W1.2, B3.2.2., IO.03, IO.05.

Acceptance criteria for this user story:

- *The third party verification system has been correctly integrated with the system and displayed.

- *System can handle any error thrown by the external system.

- *The students can be verified by the verification system and then use the system service.

- *The system cannot allowed to rent a room if an unknown student identity is applied.

User Story 2:

As a Student in the system, I can make a successful booking after reaching an agreement.

Related requirements: W3.2, F.03, B3.2.5., IO.01, IO.05.

Acceptance Criteria for This user story:

- *The payment party is integrated with the system and displayed.
- *Student can make transactions safely when they have signed an agreement with homeowners.
- *Any error thrown will be handled by the system and the student will be prompted on how to fix it.
- *Student will receive a payment and booking confirmation.

User Story 3:

As a student in the system, I want find a suitable accommodation through my phone.

Related requirements: W2.1, W2.2, W2.3, UR.04, B3.2.4., IO.02, IO.05.

Acceptance Criteria for this user story:

- *Student can successfully search and view suitable accommodations.
- *Can use the map if desired, where any error caused by the external system is handled.

User Story 4:

As a home owner in the system, I want to be able to receive my first booking application.

Related requirements: W2.4, W3.1, B3.2.4., F.04, B3.2.5., SR.04.

Acceptance Criteria for this user story:

- *Home owner can successfully post an accommodation listing and receive applications through the chat function.