DAT231/DIT284

REQUIREMENTS SPECIFICATION
RELEASE 2

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

Change log

Release 1.0 2020-09-21

- Chapter A
  - First draft of sections A1, A2, A3..
  - First draft of stakeholder map, stakeholder analysis, and context diagram.
- Chapter B
  - First draft of sections B1.

Release 2.0 2020-10-05

- Chapter A
  - Second draft of sections A1, A2, and A3.
  - First draft of sections A4 and A5.
  - Second draft of stakeholder map, stakeholder analysis, and context diagram.
  - First draft of use-case diagram and quality grid.
- Chapter B
  - Second draft of section B1.
  - First draft of sections B2, B3, B4, and B5.
  - First draft of user story table, ER-diagram, data tables, functional requirements table, and proposed prioritization tables.
- Chapter C
  - First draft of sections C1, C2, and C3.
  - First draft of UI prototype.

Contents

List of Figures

List of Tables

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.
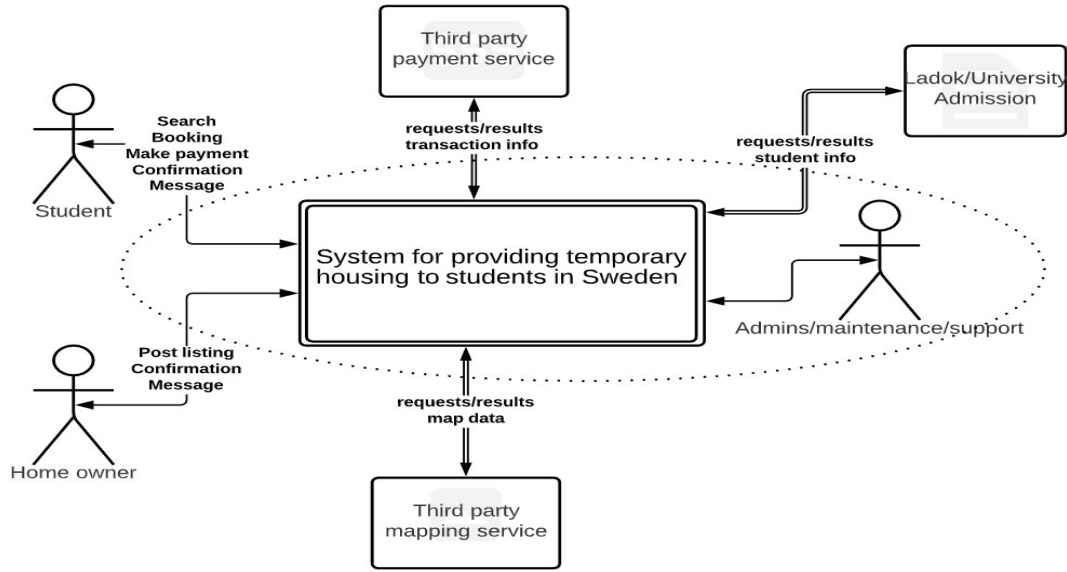
Figure A.1: Context diagram of suggested system

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.
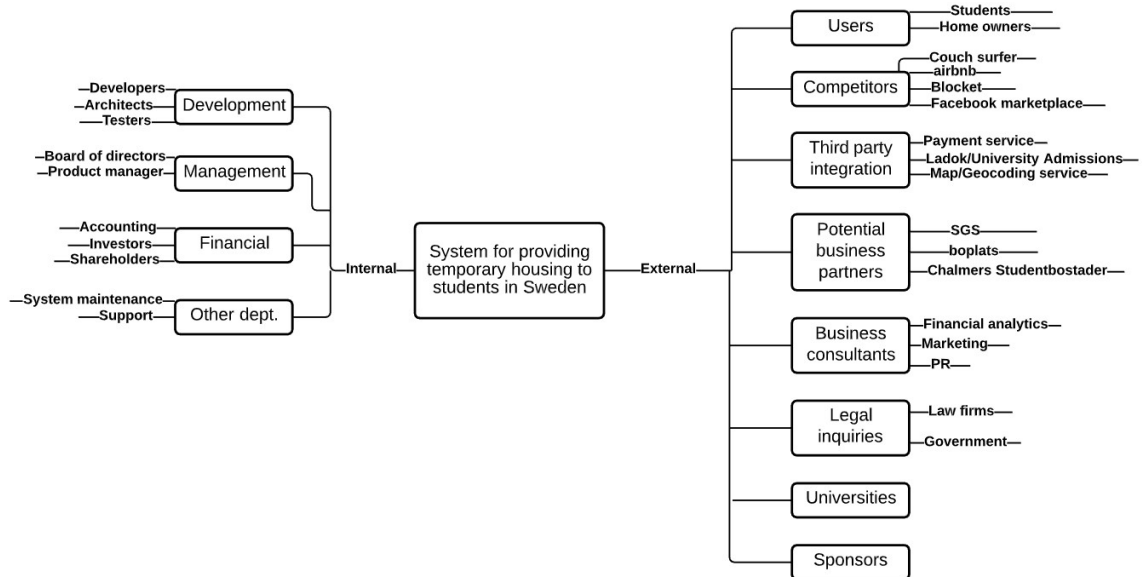
Figure A.2: Stakeholder map

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.
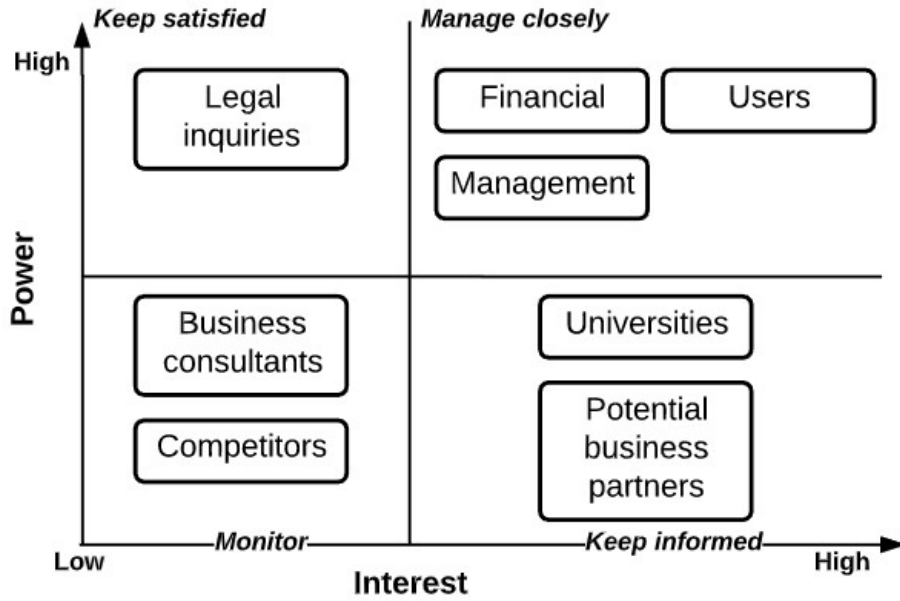
Figure A.3: Stakeholder power and sentiment chart

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 the Table A.1. A priority score out of 100 was also assigned to represent the importance of each business goals to the primary stakeholders.

Goals for the system	Solution vision	Related requirements	Priority
(BG1) People who are less tech-oriented, such as older home owners, or who lacks the language skill can use our system easily.	Fewer total screens, reduced navigation around the system. The lesser the clicks, the better. Enable the option of changing between popular languages.	F.09	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.	FR.01, FR.03, FR.04, FR.05, FR.06	45/100
(BG3) Reduce risk of fraud 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 between home owners and students to avoid fraud. Safe payment methods through trusted third party services are offered.	FR.02, FR.07, FR.08	35/100

Table A.1: Business Goals

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.

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 and receive confirmation of room booking and payment	W3.2, U.06

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
5. Receive payment and booking confirmation and option of signing long term agreement with student.	U.06, U.07, U.05

Table A.3: Logical Flow for Home Owners

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.
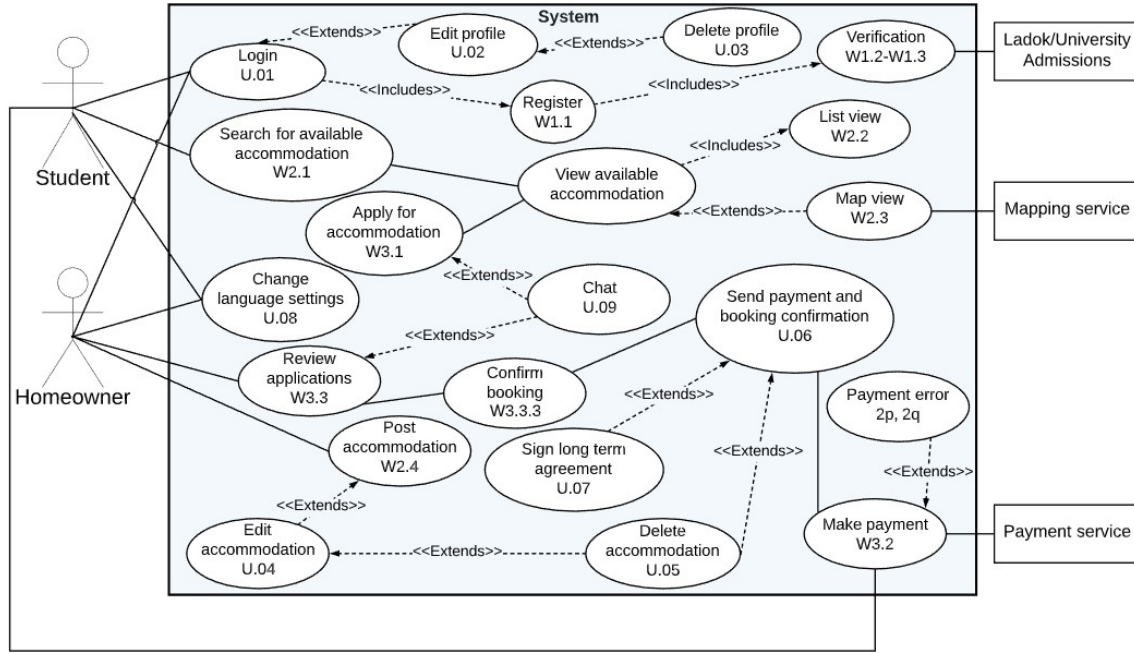
Figure A.4: Use-case diagram

The use case diagram, Figure A.4, shows several UML use cases. The box represents the proposed system. The student as an actor can carry out use cases such as login, search for available accommodations, chat, make a payment, and so on. On the other hand, The homeowner can carry out use cases such as login, post accommodation, review application, and chat with students. This use case has to be supported by the system.

In product terms, each use case might involve several product functions for example register, edit profile, delete the profile, verification, and so on. The homeowner also might have other extended functions such as edit accommodation, delete accommodation, confirm bookings, edit profile, and delete profile.

Ladok/university system is also another actor. It is responsible for managing the student's account and confirmations. Another actor map service provides the map for each accommodation. The payment system is also another actor. It is responsible for handling transactions, payment errors, and sending payment confirmation. The corresponding tasks and user stories are represented in the use case bubbles.

A5. Critical attributes

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.4 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		5			
Usability	2				
Efficiency			X		
Revision	-	-	-	-	-
Maintainability			X		
Test-ability			X		
Flex-ability			X		
Transition	-	-	-	-	-
Portability			X		
Interoperability	3			4	
Re-usability			X		
Install-ability				X	

Table A.4: 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. Also part of one of the business goals.

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. Also one of the business goals.

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.

4. Interoperability: Integration with spreadsheet, etc. unimportant. Build-in information form.

5. Reliability/availability: We aim to help students find affordable temporary and acute housing. Hard to meet user needs if the system were to fail.

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 system in terms of vulnerabilities etc which could give an unauthorized entity access to personal data.
SCALE: Security rating as assessed by an independent third party's review of the system.
METER: Security issues found during the system review.
MUST: Not have any security issues.

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.
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: System Dependability
GIST: The fault tolerance of the system
SCALE: If the system loses the connection to the third party, the user should be informed.
METER: Measurement obtained from 24 hours of usage during testing.
MUST: 100% of the time

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.

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 needs 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.

Work area 1: Account management

This work area compromises 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.

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		B3.2.1.
2. Record personal information		
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	
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	
3. Receive confirmation of verification	System update the account's status to verified and set study location	

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.

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	
2. Receive confirmation of verification	System updates the home owner's status to verified	

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		
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 based on Price, City		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		
1b.	**Problem:** Third party service not available	Usually updating the page or entering search credentials again solves it	
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 after owner has posted. 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		
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	
2p. **Problem:** System filters through the 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

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	

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	
2. Enter payment details		
2p. **Problem:** Third party service not available	Prompt user to try again in a while	
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.	B3.2.5.

W3.3   Select student for communication

This task is for selecting the students interested in the offered accommodation.
There can be multiple students interested in the same accommodation.

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	
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	

B2.   Supplementary user stories

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 tasks, at least from a user perspective, as user stories.

ID	User story
U.01	As a user I want to login to my personal account to give me access to my own profile and user activities in the system.
U.02	As a user I want to be able to edit the details of my profile to keep my information updated.
U.03	As a user I want to be able to delete my account if I no longer have any need for it.
U.04	As a home owner I want to be able to edit accommodation details to keep its information updated.
U.05	As a home owner I want to able to delete my accommodation if I no longer desire to have it posted.
U.06	As a user I want to receive a payment and booking confirmation after finalizing a booking as a convenient reminder where all booking details are specified.
U.07	As a user I want to have the option of signing a long term agreement with an underlying contract as to avoid future legal issues and for protection against frauds.
U.08	As a user I want to able to change language to other popular options if my understanding of the currently used language is very limited.
U.09	As a user I want to able to communicate with other users through a chat system.

Table B.1: User stories

B3.   Data requirements

This section describes the data recorded by the system. The following section shows an Entity-Relationship diagram with a Data Dictionaries for each entity described after.

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 accommodation 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.
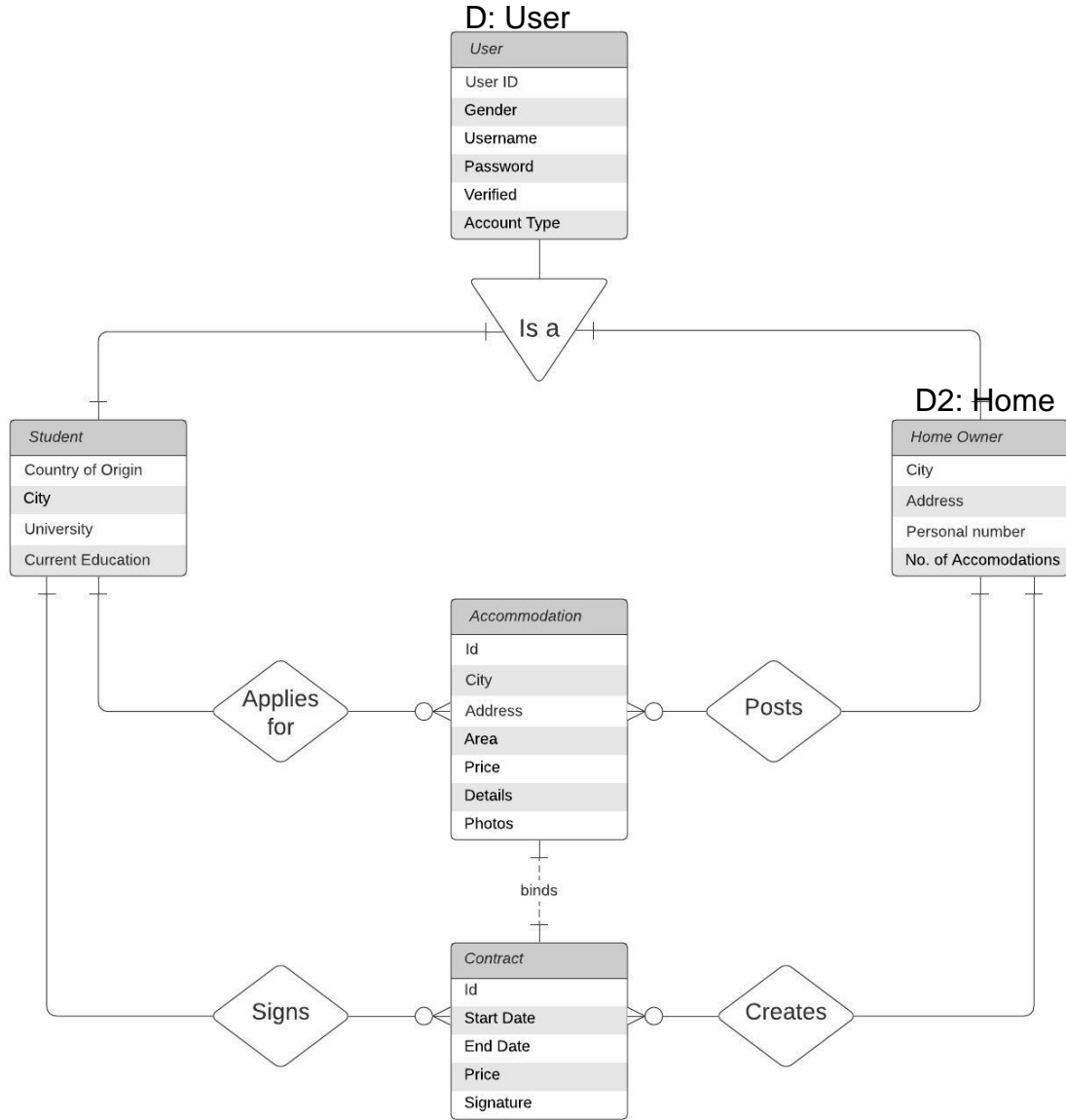
Figure B.1: Entity-Relationship Diagram

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. The Age is automatically

calculated by the system after the Date of Birth has been entered.

Attribute Name	Datatype	Description	Example
ID	Char8	Unique Identifier for each user	STU00012
Gender	String	The User's gender, choose between Male, Female and other	Male
Age	Integer	The User's Age in numbers	22
Username	String	The User's name. This is also the display name	Enrico Maxwell
Password	String	Password for the user, stored as a 256-bit hash value	qiyh4XPJGs...
Verified	Boolean	States if the user is verified by the system or not	1
Account Type	String	Either Student or Home Owner	Student
Date of Birth	Date	The User's date of birth, stored as DDMMYYYY	19041992
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
Country of Origin	String	Student's Birth Country	Israel
City	String	Student's Birth City	Mumbai
University	String	The University where the Student is studying at	University of Gothenburg
Current Education	String	Current Education Level of the Student	Masters of Science
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. 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
City	String	The Home Owner's current City of Residence	Uppsala
Address	String	The Home Owner's current Address	Fridhelmsgatan 46, Gothenburg
Personal Number	String	Swedish Personal Identity Number	950621-921x
Accommodations	String	List of Accommodation Id's posted by the user	[ACC0012, ACC013]
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
Id	Char7	Unique Identifier for the Accommodation	ACC0012
Type	String	Shared or Single housing	Shared
City	String	Location of the Accommodation	Uppsala
Address	String	Complete Address including pin-code	Fridhemsgatan 46, 451 43
Area	Integer	Total Carpet Area occupied in Square meters.	25
Price	Float	Proposed price in Swedish Kronas	5420
Details	Longtext	General description of the Accommodation along any facilities offered	
Furnished	Boolean	Is the Accommodation Furnished or not	0
Deposit	Float	Deposit amount, if any, in Swedish Kronas	10000
Available	Boolean	Check if the Accommodation is still available	1
Available From	Date	Date the Accommodation is available from	23102020
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.

Attribute Name	Datatype	Description	Example
Id	Char6	Unique Identifier for the Contract	CON123
Start Date	Date	Starting Date of the Contract	19102020
End Date	Date	Expiry Date of the Contract	20102025
Price	Float	Finalized Accommodation price in Swedish Kronas	6420
Payment Reference	media	Attachment to the payment made	payment.pdf
Signature	Blob	A digital Signature of the Student	

B4.   Functional requirements

This section provides an overview of the functional requirements of the system. In the table below, Table B.2, the ID of the functional requirement 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 use-cases that must be supported to verify the functional requirement in question.

ID	Category	Description	Related task or user story
FR.01	Account management	The system must allow for users to sign up, edit, and delete accounts.	W1.1, U.01-U.03
FR.02	User verification	The system must be able to verify identities, student status, and accommodations as it is also one of the business goals.	W1.2
FR.03	Accommodation management	The system must users to post, search with filters, browse, view, and delete accommodation listings. Also part of one of the business goals.	W2.1, W2.2, W2.3, W2.4, U.04-U.05
FR.04	Location browsing	The system must allow users to browse accommodation listings by location on a map through a third party service.	W2.5-W2.6
FR.05	Communication	The system must allow for home owners and students to contact each other through a provided chat system. Also part of one of the business goals.	W3.1, U.09
FR.06	Confirmation	The system must send out booking and payment confirmations.	W3.2
FR.07	Contract	The system must allow for home owners and students to reach a long term agreement with an underlying contract.	W3.3
FR.08	Payment	The system must allow users to pay and receive payment using a secure and trusted third party service provider. Also part of one of the business goals.	W3.2
FR.09	Language	The system must allow for users to change language to other popular choices. Also part of one of the business goals.	U.08

Table B.2: Functional Requirements

B5.   Proposed prioritization

Software development project has multiple requirements, budgetary constraints, and tight deadlines. Therefore, there is a need to do software requirements prioritization as it is just impossible to do everything at once. In this subsection, we use two different techniques to make decisions on which set of requirements need to be implemented first and which ones can be delayed until a later release.

B5.1. $100-test

This method consists of three steps. First, we aim at three main stakeholders, which is developers, students and homeowners. Then a fictional $100 is given to each stakeholders. 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.3 and the table which is been ranked can be seen in table B.4

Functional requirement	Developers	Students	Home owners
FR.1 Account management	10	5	5
FR.2 User verification	10	**20**	**15**
FR.3 Accommodation management	10	10	10
FR.4 Location browsing	10	15	5
FR.5 Communication	**20**	**20**	10
FR.6 Confirmation	10	10	10
FR.7 Contract	10	5	15
FR.8 Payment	15	10	**25**
FR.9 Language	5	5	5

Table B.3: Each stakeholder's distribution in the $100 test

Points	Developers	Students	Home owners
5	FR.9	FR.1,7,9	FR.1,4,9
10	FR.1,2,3,4,6,7	FR.3,6,8	FR.3,5,6
15	FR.8	FR.4	FR.2,7
20	FR.5	FR.2,5	
25			FR.8

Table B.4: Each stakeholder's distribution in the $100 test (ranked)

For developers, the most important factor is whether the system function is fast and stable, especially when it comes to the transmission of information, such as chats between students and home owners (FR.5), they hope that the information can be quickly notified to each other to achieve real-time online Communication, so they

invested $20 for the communication function.

Secondly, they also care about whether students and homeowners can safely complete payment transactions (FR.8), because as long as an unsafe transaction occurs, users cannot fully trust our system, and distrusting the system will lead to serious consequences.

For students, we can see from the results that the three most interesting functions are verification (FR.2), location browsing (FR.4), and communication (FR.5).

The reasoning behind the verification becomes the main concern is that students, especially female students, might want to make sure whether the home owner is certified by an authority and verified which would in turn help build trust and make them feel more safe when renting.

The function of location browsing is related to whether students can actually apply for an accommodation in the system. The feature allows students to apply for accommodations that have good locations in terms of being close to the city center or their school in relation to their price. A list with street names does very little for students who has previously not been in the city.

Another point that made them pay attention to is whether they can communicate with the landlord in real-time. One of the students had the experience of using Airbnb, but that experience was not very good, because the user sent multiple landlords a message to rent an accommodation. However, no reply was received in the end. Because of that, they were very concerned about the communication function of the system.

For the homeowners, the most important thing is whether to receive the money within the stipulated time when the transaction occurs(FR.8), and if there are some legal issues involved, what measures the homeowners has to protect their rights(FR.7),so one of our core function is to sign an agreement which including the common legal problems in Sweden and the solutions.

Another point they pay attention to is users verification (FR.2). Because they want to make sure whether the person applying for renting is a trustworthy person or whether they can live in the place stable for a longer time. So verifying the identity of students is also very important for them in this case.

B5.2. Ranking (sorting)

Ranking is a easy prioritization technique which means the most important requirement is ranked 1 and the least important is ranked n (for n requirements).

Ranking seems to be more suitable for a single stakeholder because it might be difficult to align several different stakeholders' views. So we divided our main stakeholders into two groups, one is from students point of view, another is from home owners perspective.

For students, the list of ranking requirements are:
1. FR.4 Location browsing
2. FR.5 Communication
3. FR.2 User verification
4. FR.8 Payment
5. FR.3 Accommodation management
6. FR.6 Confirmation
7. FR.7 Contract
8. FR.1 Account management
9. FR.9 Language

For home owners, the list of ranking requirements are:
1. FR.8 Payment
2. FR.2 User verification
3. FR.7 Contract
4. FR.5 Communication
5. FR.6 Confirmation
6. FR.3 Accommodation management
7. FR.4 Location browsing
8. FR.1 Account management
9. FR.9 Language

Based on these rankings, we can see that different stakeholder prioritize different requirements. However, some requirements such as user verification, payment are important to all stakeholders.

B5.3. Next release

In the next release for this project, more focus will be placed into covering quality requirements, performance requirements, data requirements, and functional requirements.

B5.4. Second release

In the second release for this project, more focus will be place into covering system requirements and refining the existing ones.

B5.5. Future releases

Describe which requirements not to cover in the first two releases.

C

System requirements

This section discusses the system-level requirements that describe the features and behavior of the system. The system should fulfill requirements to satisfy stakeholder needs and requirements. It includes the UI prototypes, detailed data requirements, and acceptance tests.

C1.   UI prototype

A UI prototype was devised in order to outline the core functionality of the system. Figure C.1 shows the home and student verification (W1.2) page of the system, as per the requirements in FR.2. In figure C.2 it is illustrated how the UI would look when trying to find accommodation, i.e., searching (W2.1) and initiating a conversation with a home owner (W3.1). These are related to FR.4 and FR.5. As for when student and home owner has signed an agreement and the student is in the process of paying (W3.2), the resulting UI prototype can be seen in figure C.3. This conforms to FR.8 but is also related to FR.6 and FR.7.
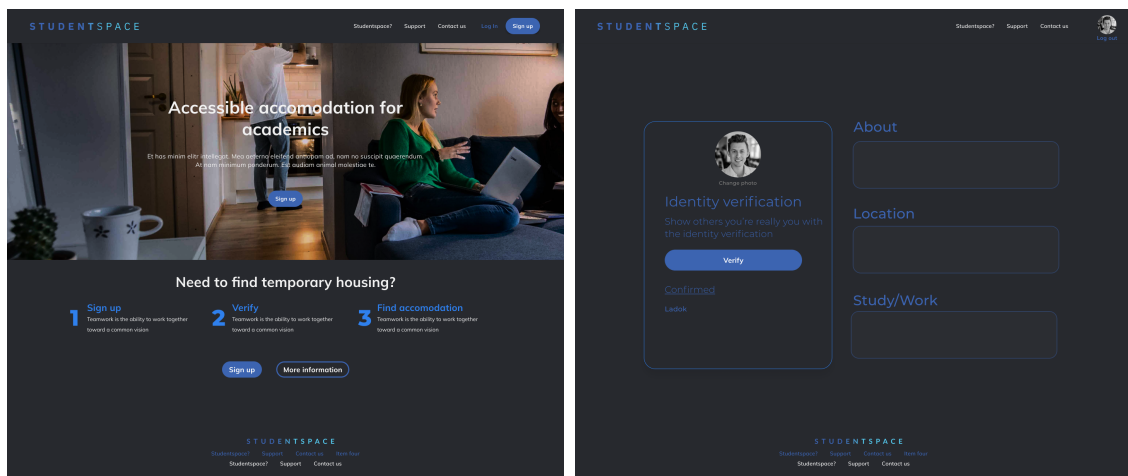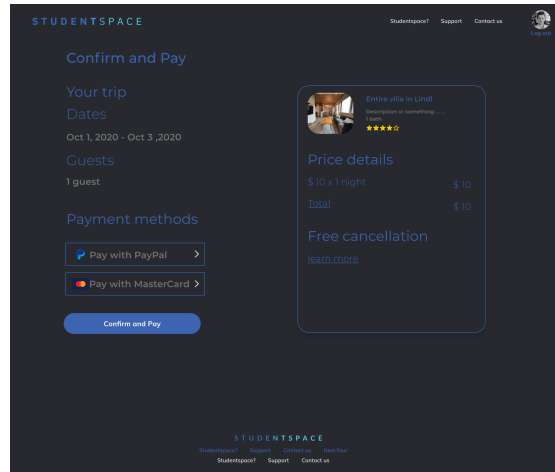

(a) Home page   (b) Verification

Figure C.1: ..
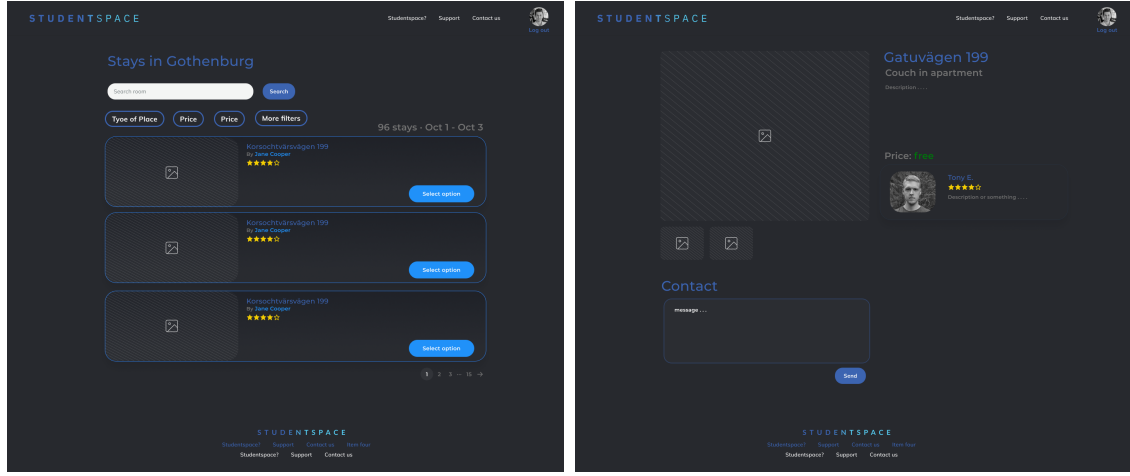
(a) Search   **(b)** Accommodation

Figure C.2: Finding accommodation

Figure C.3: Confirm and pay for accommodation

C2.   Detailed data requirements

The Entity-Relational diagram in Figure B.1 shows that the system has to keep track of data of students, homeowners, accommodation, and contracts. The relationship between the student with the accommodation and the homeowner with accommodation shows that one student can 'applies for' several accommodations and also the homeowner may 'post' several accommodations.

The system must store the following data:
User
The E-R diagram shows that the system must keep users' data. The user can be students or homeowners. The data includes the user's ID, Gender, user Name, password, verified, and account Type.

Student

The system also shows that it must keep track of for each student data; their Country of Origin, city, university, and current Education. Also, for each student, there is a record of the user which is inherited from 'User'.

Homeowner

The system must keep track of each homeowner's data. For each homeowner, there is a record with city, address, personal Number, Accommodations, and user information which is inherited from the 'User'.

Accommodation

The system has to keep track of data about the accommodation. It should include the accommodation's Id, city, address, area, price, details, and photos.

Contract

The system keeps track of the following information for every contract. This includes ID, start Date, end Date, price, and signature.

C3.   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 Sweden, I can been verified through Ladok/Universityadmissions, so I can rent an accommodation in this system.

Acceptance criteria for this user story:

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

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 user, I can be user verified by providing personal identification documents proving my identity, so I can rent and rent out an accommodation in this system with other trustworthy users.

Acceptance criteria for this user story:

A program can compare the personal identification document to the users profile and confirm whether it is the same person.

The user can be verified by the verification system and then use the system service.

The system cannot allowed to rent out a room if an unknown homeowner identity is applied.

User Story 3:

As a Student in the system, I can make payment transactions with homeowners

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.

Homeowner can withdraw their pending credit from their account.

Users can't tap into the payment parties when they have not signed an agreement.

User Story 4:

As a user in the system, I want to receive booking confirmation after I have made a payment.

Acceptance Criteria for this user story:

Students receive confirmation when they have booked rooms.

Student can withdraw their bookings from their accounts.

Students can not sign an agreement with homeowners when they do not confirm the booking.

Homeowners can not accept notification if Students do not confirm their booking.

User Story 5:

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

Acceptance Criteria for this user story:

User can successfully search and view suitable accommodations.

User Story 6:

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

Acceptance Criteria for this user story:

User can successfully post an accommodation listing and accept applications.