DAT231/DIT284

Requirements specification

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*)

Department of Computer Science and Engineering
Chalmers University of Technology
Gothenburg, Sweden 2020

Contents

List of Figures

List of Tables

A
Background and supplier guide

A1. Background and vision

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 lead 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.

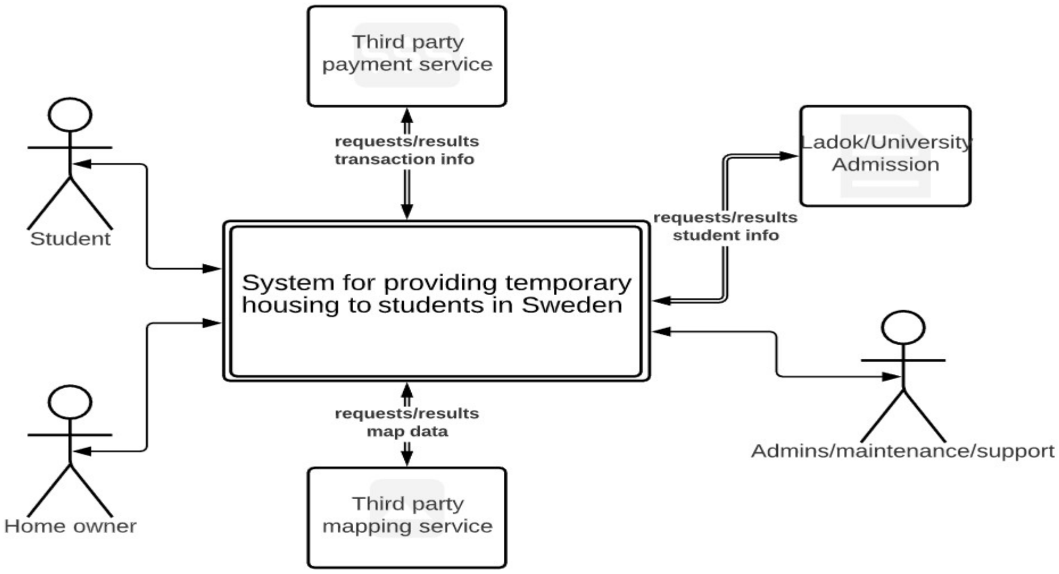
Third party payment service
requests/results transaction info
Student
Ladok/University Admission
requests/results student info
System for providing temporary housing to students in Sweden
requests/results map data
Admins/maintenance/support
Home owner
Third party mapping service

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.

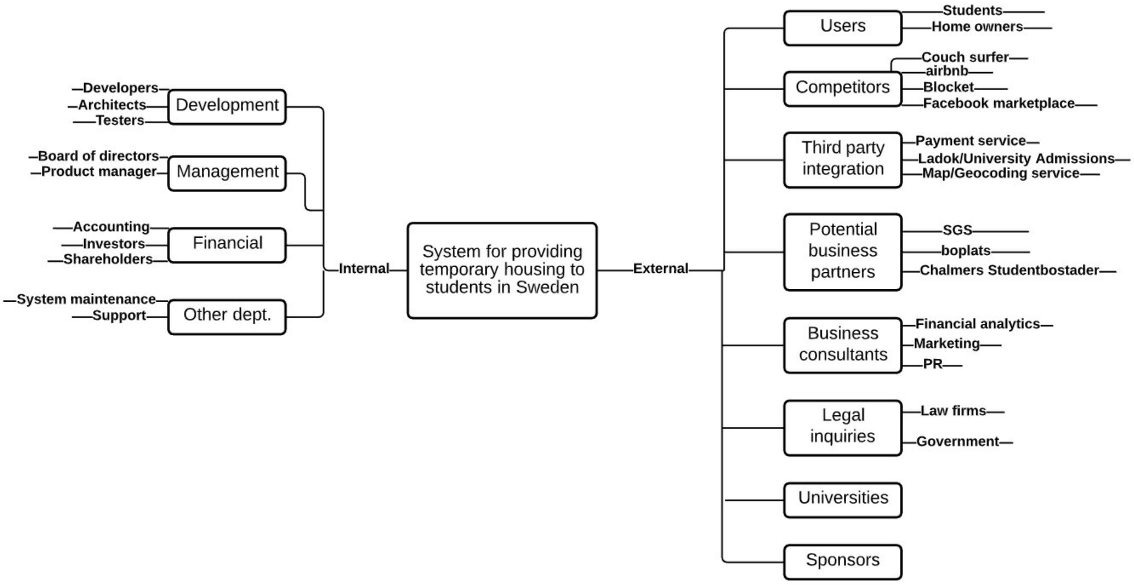

Figure A.2: Stakeholder map

In the stakeholder analysis chart, Figure A.3, an illustration of how major stakeholders interact with the system is displayed.

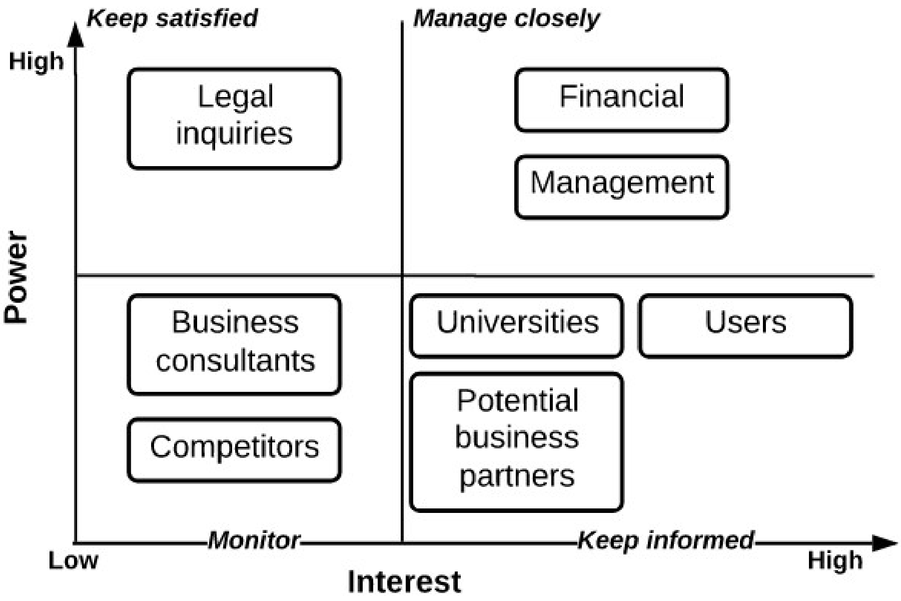

Figure A.3: Stakeholder power and sentiment chart

A3. (missing) Supplier guide

A4. (missing) Overall solution and alternatives

B

High-level demands

This chapter analyzes how the stakeholder's business goals are met through the requirements, how to reduce high-risk requirements, and how to compare proposals.

B1. 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 while others can be repeated several times.

The logical flow is carried out entirely digital. Table B1.1 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 B1.2 details the steps undertaken by the homeowner to display their accommodation in the system. Chapter C shows the details.

Steps in finding accommodation for the student	**Tasks and Subtasks**
1. Create a Student account	C1
2. Fill their personal and study information in detail	C2, C3
3. Student verification	C4
4. Search for available room(s)	C7
5. Browse through the list of room(s)	C8, C9
6. Contact the owner(s) of the room(s) the student is interested in	C12
7. Make payment and receive confirmation of room booking and payment	C13, C14

Table B.1: Logical Flow For Students

Steps in offering accommodation for the home owner	**Tasks and Subtasks**
1. Create an account as an Owner	C5
2. Fill their personal information in detail	C2
3. Post a room with detail information about the room condition	C10
4. Home owner verification	C6
5. Select a student to offer the room to based on home owner preference	C15
6. Sign an agreement with the student, receive payment and booking confirmation, remove listing	C16, C10, C11

Table B.2: Logical Flow for Home Owners

B2. Business goals

The customer's reason to acquire the system is to reach some business goals. Both the customer and supplier expect that the system contributes to the goals as stated below. The supplier can rarely reach the goals alone without the online platform. This means that the goals are requirements for the customer but not for the supplier. They are shown in a table only to provide an overview.

Goals for the system	**Solution vision**	**Related requirements**
1. Reliability in achieving the user tasks	Fewer total screens, reduced navigation around the system. The lesser the clicks, the better. Be able to support all user tasks and perform them correctly under a specific time duration.	(TBA)
2. Connect Homeowners and Students in a common platform	Provide communication channels with an easy point of access. Present static information about the room before communication as to meet expectations. Allow for easy modification of user details.	(TBA)
3. Safe and secure exchanges for all parties	Home owners and students have their identities and status verified. Safe payment methods through trusted third party services are offered. A two-way review system is provided.	(TBA)

Table B.3: Business Goals

B3. (missing) Early proof of concept

B4. (missing) Minimum requirements

B5. (missing) Selection criteria: Highest net benefit

B6. (missing) Selection criteria: Most score points per dollar

C

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

C1. Create a student account

This task shows the process of creating a student account in our system.

Users: Student.
Start: Get access to the website.
End: Registration confirmation .
Frequency: Medium, once per student.
Difficult: Never.

	Subtasks and variants:	Example solutions:	Code:
1.	Open the system and click the sign up as student button		
2.	Fill out basic personal information like username and password		
3.	Register to the system		
3p.	**Problem**: If someone has the same username	Try with different usernames	
5.	Send confirmation message to the user by email or phone		
6.	User opens the link and confirms registration		
7.	Registration completed		

C2. Fill out personal information

This task provides detailed personal information about the user. This will include the user's full name, current addresses, phone number, personal email, social security number, and date of birth.

Users: Student, Owner
Start: Login to the system.
End: Save the information.
Frequency: Medium, at least once per account.
Difficult: Never.

	Subtasks and variants:	Example solutions:	Code:
1.	Go to personal information page		
2.	Fill out personal information in detail		
3.	Save the information		

C3. Fill out study information

This task provides detailed study information about the student. This will include the city, campus, program, program starting and ending dates, study duration, and proof of all the provided documents.

Users: Student.
Start: Access the study information page.
End: Save the information about the study.
Frequency: Medium, at least once per student account.
Difficult: Never.

	Subtasks and variants:	Example solutions:	Code:
1.	Go to study Information page		
2.	Fill out study information in detail		
3.	Save the information		

C4. Student verification

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

Users: Student.
Start: A registered student accesses the verification page.
End: Student 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:	Code:
1.	Click on *Sign in*	Open up a new page where student can sign in through Ladok/Universityadmissions.se	
2.	Sign in with student account acquired from university		
3.	System verify student details		
3p.	**Problem:** Could not verify details	Show an error message and where to go for further help	
4.	Set account status to verified and set study location		

C5. Create an owner account

This task creates an account for homeowner. The steps of this task should be easy for the owner to operate. It's important to minimize the steps used in this task and offer tips to homeowners to make system login easier.

Users: Home Owner.
Start: Get access to the website.
End: Registration confirmation.
Frequency: Medium, at least once per home owner.
Difficult: Never.

	Subtasks and variants:	Example solutions:	Code:
1.	Open the system and click the sign up as home owner button		
1p.	**Problem**:The owner is already in our system		
1q.	**Problem**: If someone has the same username	Try with different usernames	
2.	Send confirmation message to the user by email or phone		
3.	User opens the link and make a confirmation		
4.	Registration completed		
5.	The owner can choose to upload their credentials for verification		

C6. Home owner verification

This task verifies if 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: After a Home owner creates an account and uploads his credentials.
End: After the System have verified home owner documents and put a verified status icon in his account.
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:	Code:
1.	Check uploaded documents	System checks the total amount of documents uploaded	
1a.	**Problem:** No documents available	No further tasks performed	
2.	Verify documents	System scans documents and cross-checks id for verification	
2p.	**Problem:** Documents unable to be verified	Send a message to the user with the error message	
2q.	**Problem:** Verification may not immediately answered by system maintainer, it may take lots of time	Send message to support to take a closer look	
3.	Set account status as verified	Show a Verified icon besides username whenever displayed	

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.

C7. Search for available room(s)

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

Users: Mainly students.
Start: User has accessed page where available accommodation is listed.
End: List of available accommodation is updated.
Frequency: High.
Difficult: Never.

	Subtasks and variants:	Example solutions:	Code:
1.	Go to *find accommodation* page		
2.	Enter search criteria		
3.	Perform search	System search database based verified study location and specified search criteria	
4.	Update list of rooms		
4p.	**Problem:** No search criteria was entered	Show all available rooms based only on study location	
4q.	**Problem:** No results	Display a message stating that no results were found	

C8. Browse list of room(s)

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:	Code:
1.	Filter or sort the list based on price, location etc		
2.	Scroll through the list of accommodation		
2a.	Show next page		
3.	Click on listing to show page with additional information on specific accommodation		

C9. Browse map of rooms

Similar to task C6, 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:	Code:
1.	Click on "Show 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		

C10. 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 click “Post housing” in the menu button.
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:	Code:
1.	Home owner post detail information of accommodation		
1a.	There is an option of payment		
2.	Home owner successfully post listings of accommodation		
2a.	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**	Prompt home owner to fix	
3.	Allow home owner to edit listings after posting them		
3a.	Home owner can post another accommodation		

C11. Remove listing

This task is for removing the booked listing from the database.

Users: (No)
Start: After accommodation booking has been confirmed.
End: Data has been updated.
Frequency: Low.
Difficult: Never.

	Subtasks and variants:	Example solutions:	Code:
1.	Get information from payment page		
2.	Update the accommodation data in the system		
3.	Remove the room from list of rooms		
4.	Trigger "Receive accommodation booking and payment confirmation"(C14) task		

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

C12. 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 and room for 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:**	**Code:**
1.	Enter dates/times		
1p.	**Problem:** Date already taken	Try different dates	
2.	Confirm booking		
3.	A message is sent to the home owner with the dates and times requested, opening a conversation dialog		

C13. 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 has been processed.
Frequency:	Low, at least once per listing.
Difficult:	Never.

	Subtasks and variants:	**Example solutions:**	**Code:**
1.	Click on make a payment after receiving confirmation		
2.	User will be taken to another window where payment info will be entered		
2p.	**Problem:** Third party service not available	Wait a little and try again, refresh the page	
3.	Click on make payment		
3p.	**Problem:** Payment declined or could not be processed	Contact bank	
4.	User will be taken back to system main page		

C14. Receive accommodation booking and payment confirmation

This task creates a page for accommodation booking and payment confirmation to ensure users confirm their booking and payment.

Users: Students and homeowners.
Start: After students pays for the accommodation.
End: When users back to front page.
Frequency: Low, at least once per listing.
Difficult: Never.

	Subtasks and variants:	Example solutions:	Code:
1.	Get information from booking page and payment page		
2.	Create a page with booking and payment information		
3.	Send an email for both students and homeowners		

C15. 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:	Code:
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 on click	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	

C16. Sign an agreement with the student

This task is for signing a formal agreement with the student regarding the accommodation. This agreement is legally binding and only applies to long-term contracts.

Users:	Home Owner.
Start:	Once the accommodation is confirmed with the student and Home Owner.
End:	After the payment is confirmed and the room is removed from available listings .
Frequency:	Low, at least once per listing.
Difficult:	Never.

	Subtasks and variants:	**Example solutions:**	**Code:**
1.	Send agreement to student and home owner	System sends agreement to both parties to sign	
1a.	**Problem:** No students listed	No further tasks performed	
2.	Payment confirmation	Home owner receives payment confirmation after student initiates it in Task C9	
3.	Remove accommodation from listing	Proceed to C11	

D

(missing) Data to record

E

(missing) Other functional requirements

E1. System generated events

E2. Reports

E3. Business rules and complex calculations

E4. Expansion of the system

F

(missing) Integration with external systems

F1. Common integration requirements

F2. Integration with new external systems

G

(missing) Technical IT architecture

G1. Existing hardware and software

G2. New hardware and software

G3. The supplier operates the system

H

(missing) Security

H1. Login and access rights for users

H2. Security management

H3. Protection against data loss

H4. Protection against unintended user actions

H5. Privacy requirements

H6. Protection against threats

I

(missing) Usability and design

I1. Ease-of-learning and task efficiency

I2. Accessibility and Look-and-Feel

J

(missing) Other requirements and deliverables

J1. Other standards to obey

J2. User training

J3. Documentation

J4. Data conversion

J5. Installation

J6. Testing the system

J7. Phasing out

K

(missing) The customer's deliverables

L
(missing) Operation, support, and maintenance

L1. Response times

L2. Availability

L3. Data storage

L4. Support

L5. Maintenance