Requirements Document

Version: R3
Latest Update: 28-10-2024

Group 21

Authors:

- Shradha Shinde (shradha@student.chalmers.se)
- Oluwatosin Omotoyinbo (twisstosin@gmail.com)
- Ruixuan Li (ruixuanl@student.chalmers.se)
- Saif Sayed (gussayedfa@student.gu.se)
- Suvrangshu Barua (suvrangshu.turno@gmail.com)
- Zihan Kuang (zihan_kuang@outlook.com)

Content

Version History

Version	Date	People	Work
R3	2024-10-28	Zihan,Ruixuan, Shradha	Updated functional requirements using Planguage (including experience)
		Saif	Add Traceability Matrix
		Saif	Add Trade-offs and Quality Assurance in Experience report
		Suvrangshu	Acceptance Tests
		Zihan	Add Quality Grid (including experience)
		Ruixuan	Add Data Dictionary (including experience)
		Zihan,Ruixuan	Complete Prioritization (including experience)
		Zihan, Suvrangshu	Complete next release
		Saif	Design UI Prototypes (including experience)
		Oluwatosin, Suvrangshu	R1 Revision Updates
R2	2024-10-06	Zihan	Stakeholder Map (including experience)
		Saif, Suvrangshu, Zihan	Updated Context Diagram (including experience)
		Shradha, Ruixuan	Functional & Performance Requirements (including experience)
		Ruixuan	Data Requirements (including experience)

Note: This document's grammar and formatting were checked using ChatGPT in some of the sections.

		Oluwatosin, Suvrangshu	Format Documentation, High-level Business Case, Core Functionality, User Requirements Specifications, High-level Performance Requirements (including experience)
		Shradha, Saif, Suvrangshu	Interview Planning (including experience)
		Saif	Updated Use Case Diagram and Descriptions, Format Documentation (including experience)
R1	2024-09-22	Suvrangshu, Saif, Zihan	Context Diagram (including experience)
		Suvrangshu,Zihan, Shradha	Business Goals (including experience)
		Shradha,Zihan	Stakeholders with diagram (including experience)
		Saif	Use Case Diagram (including experience)
		Suvrangshu, Ruixuan	Use Case Descriptions (including experience)
		Shradha,Zihan	Goal Domain Tracing Table (including experience)

To navigate through this document easily, use the table of contents to jump directly to specific sections of this document you are interested in. There are four sections in total: **1.** High-Level Description, **2.** Core Functionality, **3.** User Requirements Specification and **4.** System Requirements. Each high-level section contains several sub-sections relevant to that particular section.

High-level Description

This part of the requirements document scopes the project and provides the key ideas.

Background

Most storage facilities today offer fixed rental durations at high prices, which often do not meet the needs of individuals—especially students and young professionals—who require affordable, flexible storage solutions. Many students and young workers struggle to find convenient and budget-friendly storage options to temporarily store their belongings while securing permanent housing. This challenge is even greater for international students and workers who have relocated from abroad.

Note: This document's grammar and formatting were checked using ChatGPT in some of the sections.

Our project aims to develop a system that offers an alternative to traditional storage facilities, enabling individuals to rent nearby storage spaces at affordable rates and for flexible time periods.

Goal and Scope

The goal of the system is to enable individuals, especially students, to rent temporary storage spaces from other people at a lower cost than traditional/third-party providers. This system will also allow those with spare storage space to offer it and earn extra income.

The scope of the project is to create a seamless system for peer-to-peer storage space sharing, and the service will be available to users with web-enabled devices. This project focuses primarily on students but is applicable to anyone with storage needs or space to rent.

Primary Target Audience:

- **Students:** The platform primarily caters to students, who often need temporary and affordable storage solutions due to dorm relocations, travel, or short-term housing arrangements.

Secondary Target Audience:

- **General Renters:** While the platform focuses on students, it is also designed for anyone with temporary storage needs, such as people in-between moves, travelers, or those with limited space at home.
- **Storage Providers:** Individuals with extra storage space, regardless of their demographic, who want to monetize their unused areas and generate additional income.

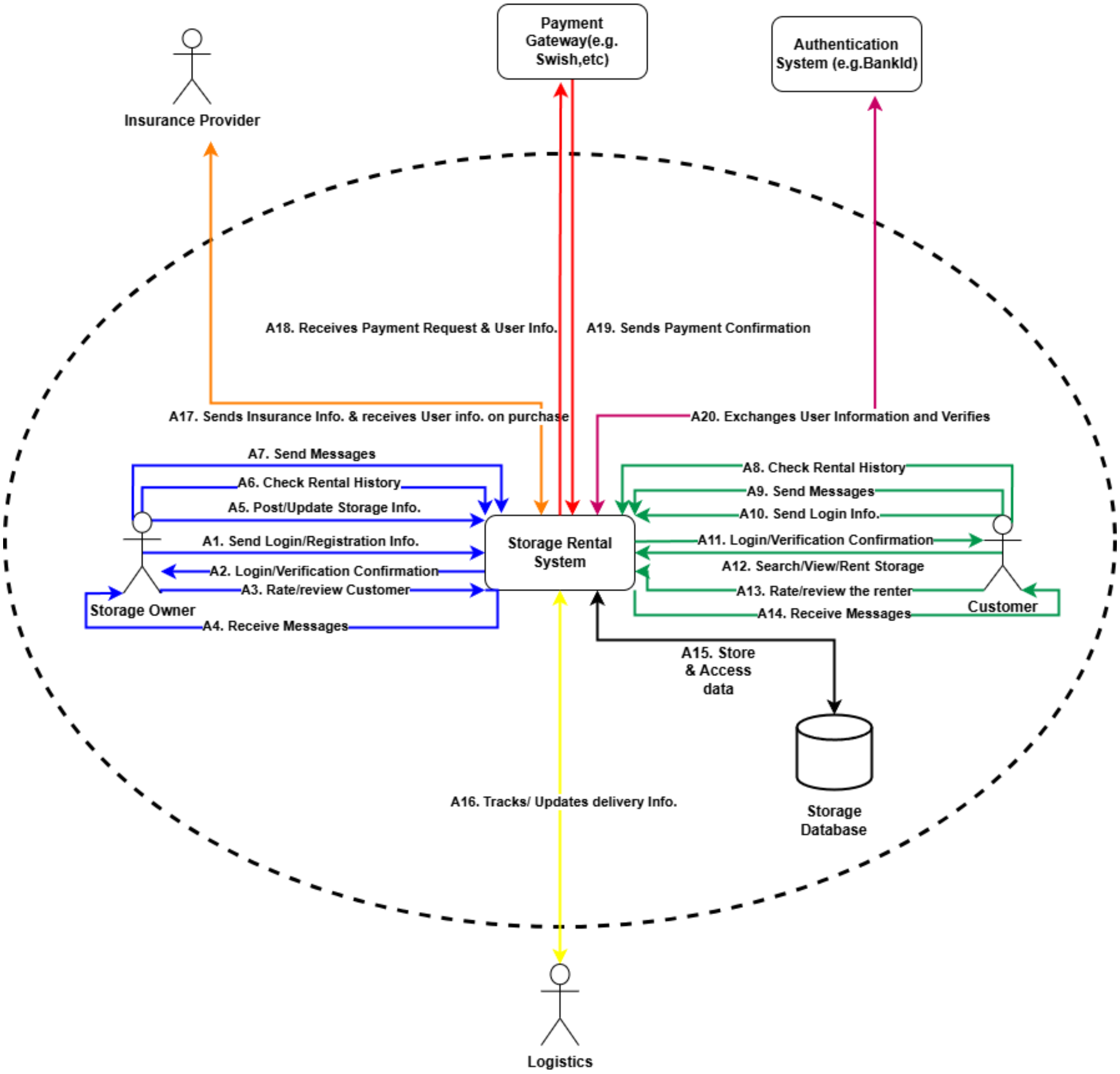

Payment Gateway(e.g. Swish,etc)
Authentication System (e.g.BankId)
Insurance Provider
A18. Receives Payment Request & User Info.
A19. Sends Payment Confirmation
A17. Sends Insurance Info. & receives User info. on purchase
A20. Exchanges User Information and Verifies
A7. Send Messages
A6. Check Rental History
A5. Post/Update Storage Info.
A1. Send Login/Registration Info.
A2. Login/Verification Confirmation
A3. Rate/review Customer
Storage Owner
A4. Receive Messages
Storage Rental System
A8. Check Rental History
A9. Send Messages
A10. Send Login Info.
A11. Login/Verification Confirmation
A12. Search/View/Rent Storage
A13. Rate/review the renter
A14. Receive Messages
Customer
A15. Store & Access data
Storage Database
A16. Tracks/ Updates delivery Info.
Logistics

Fig 1. Context Diagram of the Storage Rental System

Legend	**Key**
Action	A

Description

A storage rental system that links customers and renters while handling payments, authentication, insurance, and logistics is depicted in the context diagram **(see Fig 1)**. It also shows how a database is used to safely store system data.

Transactions for storage rents and the associated item delivery are managed by the payment gateway. The

Note: This document's grammar and formatting were checked using ChatGPT in some of the sections.

logistics team updates the system with real-time delivery information, which is assumed to be collected from users during the registration process.The system also gives users the opportunity to pay extra for an insurance package that they want to buy from a provider. Users—both customers and renters—can evaluate and review one another based on their experiences in order to promote trust. Lastly, individuals may access and review their rental history whenever they want thanks to the system.

Business Case and Stakeholder Map

This section will include the stakeholders for the platform we are creating, but also the goal domain tracing (**see Table 1**) and the business goals .

Business Goals

1. Platform Goal:

The main goal is to create a secure and efficient online platform where people can easily rent and offer storage spaces. The platform should handle user information safely, work smoothly even with many users, and be able to grow as demand increases. We want the system to be reliable and accessible from any device, ensuring a smooth experience for all users.

2. Empowering Renters:

Renters should be able to easily list their available storage spaces, manage their listings, and make changes whenever needed. This includes adding photos, descriptions, and adjusting pricing or availability. The goal is to make sure the process is simple and quick, so renters feel confident using the platform to earn extra income from their unused space.

3. Customer Experience:

Customers (people who need storage) should have a hassle-free experience when searching for storage spaces that fit their needs. This includes finding storage based on size, location, and price, with the ability to easily book and pay for the space. Our goal is to ensure that customers find the platform easy to use and that they can trust the system to handle their personal information securely.

4. Partnership with Service Providers:

We aim to build strong relationships with third-party companies, such as payment processors, delivery services, and insurance providers. These partnerships will ensure that users can securely pay for their rentals, track deliveries, and add insurance if needed. The platform should maintain high-quality standards by working closely with these providers to deliver a seamless experience.

Goal Domain Tracing

	Automate Booking	Goal/Functionality	Automate Rental Processes	Platform Security	Good Performance	Scalable System	Payment System	Delivery Function
Platform Security(BG1)	●	●		●	●	●	●	
Storage Owner (BG2)	●				●			
Customer Experience (BG3)	●	●		●	●		●	●
Service Providers (BG4)		●		●			●	●

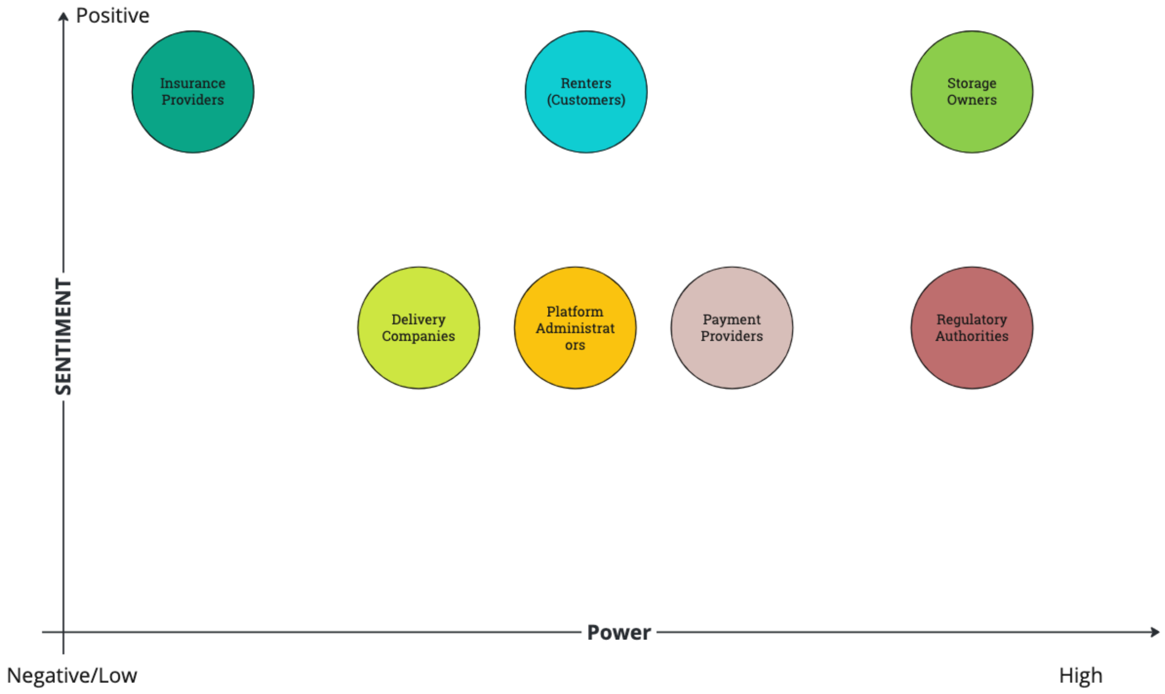
Table 1. Goal Domain Table

Stakeholder Map

Name	Relationship	Representative	Sentiment/Power
Storage Owners	Provider	Property Owners	Positive/High
Renters (Customers)	User	Students, Movers	Positive/Medium
Platform Administrators	Manager/Facilitator	System Admins	Neutral/Medium
Payment Providers	Payment Service Provider	PayPal, Stripe, etc.	Neutral/Medium
Insurance Providers	Risk Manager	Insurance Companies	Positive/Low
Regulatory Authorities	Legal/Regulatory Body	Government Agencies	Neutral/High
Delivery Companies	Delivery Partner	Logistics Firms	Neutral/Medium

Table 2. Stakeholders Table

Note: This document's grammar and formatting were checked using ChatGPT in some of the sections.



Note: This document's grammar and formatting were checked using ChatGPT in some of the sections.

Core Functionality

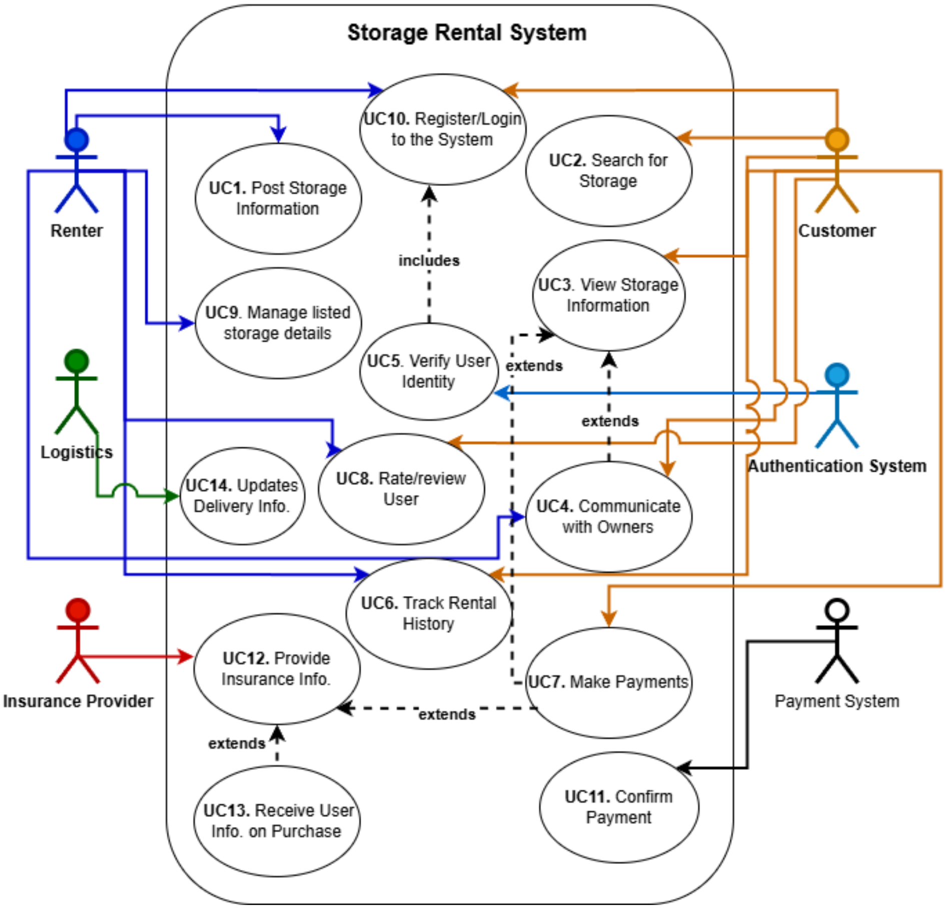

Storage Rental System
Renter
Logistics
Insurance Provider
Customer
Authentication System
Payment System
UC10. Register/Login to the System
UC1. Post Storage Information
UC2. Search for Storage
UC9. Manage listed storage details
includes
UC3. View Storage Information
UC5. Verify User Identity
extends
extends
UC14. Updates Delivery Info.
UC8. Rate/review User
UC4. Communicate with Owners
UC6. Track Rental History
UC12. Provide Insurance Info.
UC7. Make Payments
extends
extends
UC13. Receive User Info. on Purchase
UC11. Confirm Payment

Fig 2. Use Case Diagram

The platform will include the following core functionalities:

UC1 List Storage Space (related to A5):
As a storage owner, I want to list my available storage spaces with descriptions, photos, sizes, costs, and availability so that renters can easily find and book my space.

UC2 Search for Storage(related to A12) :
As a renter, I want to search for storage spaces based on size, cost, type, location, and rental duration so that I can find a storage option that fits my specific needs.

UC3 View Storage Details (related to A13, A15):
As a renter, I want to view detailed information about the storage spaces, including photos and descriptions, so that I can confidently choose the best option.

UC4 Communications (related to A4,A7,A9, A14):
As a renter, I want to communicate directly with storage owners within the platform so that I can clarify details or ask any necessary questions before booking.

UC5 Verify Identity and Listings (related to A20):
As a renter or storage owner, I want the platform to verify user identities and storage space listings so that I feel secure and can trust other users on the platform.

UC6 Track Rental History (related to A6 & A8) :
As a renter or storage owner, I want to see the history of all my rentals, both past and current, so that I can manage my rental activity and have easy access to my previous bookings.

UC7 Make Payments (related to A18):
As a renter, I want to securely pay for storage through multiple payment options so that I can choose a method that's most convenient and secure for me.

UC8 Rate and Review (related to A3):
As a renter, I want to rate storage owners, and as a storage owner, I want to rate renters, so that both parties can build a reliable reputation and promote trust.

UC9 Manage Listed Storage Details (related to A5):
As a storage owner, I want to update information about my posted storage spaces so that I can keep my listings accurate and attract more renters.

UC10 Register and Login (related to A1 ,A2, A10, A11):
As a renter or storage owner, I want to register and log in to the platform so that I can access my account and manage my rentals easily.

UC11 Confirm Payment (related to A19):
As a renter, I want the platform to confirm my payment securely through a payment gateway so that I know my transaction is completed successfully.

UC12 Provide Insurance Information (related to A17):
As a renter, I want to purchase insurance for my belongings stored in the facility so that I feel assured that my items are protected.

UC13 Receive User Information (related to A17):
As a renter, I want the platform to securely send my information to the insurance provider upon purchase so that I can ensure my belongings are covered by the policy.

UC14 Update Delivery Information (related to A16):
As a renter, I want to receive real-time updates about the delivery of my items so that I can track my belongings during transport.

Performance Requirements, Specific Quality Requirements, Constraints

This section provides an overview of the system's non-functional requirements, with more detailed explanations to follow in later sections.

Performance Requirements

PR1. The system must handle real-time searches and updates, and ensure that storage space availability is always current.

PR2. The system should have high concurrency processing capability to ensure stable operation when used by multiple people simultaneously.

Specific Quality Requirements

QR1. Security must be prioritized, particularly in storing sensitive user information like payment data and verifying user identities.

QR2. The user interface should be simple and easy to understand so that the user can learn how to use the system in a short time.

QR3. The system should be compatible and support running on different devices.

Constraints

C1. The system must be affordable to develop and maintain, targeting students and individuals moving to new places as primary users. It must also comply with GDPR due to the collection of personal data.

User Requirements Specification

These current requirements were obtained through various elicitation techniques, including explicit and creative elicitations, introspective group consensus, interviews, brainstorming, and questionnaires. This provides a holistic understanding of user needs and preferences, but further research is needed to validate these requirements against real potential users and stakeholders, such as delivery and insurance providers, for feasibility and relevance to the real world, and to list what will need future improvement.

Functional Requirements

This section explains the main features the platform needs to work properly. It covers things like how users can create accounts, search for storage, manage listings, talk to each other, and make payments. Each feature is connected to how people will use the platform.

User Authentication & Accounts

Task 1 - FR1	**Log in, create accounts, and verify identity - UC10, PR2, QR1, QR2, QR3, C1, C3**
Purpose	To ensure users can safely have access to the platform.
Trigger	The user has selected the login page.
Pre-condition	The user must possess a valid email or third-party authentication method, and provide necessary identity verification information.
Frequency	The user is most likely to create an account only once, and might log in every day or verify their identity during the first-time account setup.
Critical	During peak moving seasons.
Sub-tasks: 1. Select login or create a new account. 2. Provide required information and complete identity verification for new registrations. 3. Enter email and password to log in.	
Variants: 1. The user can reset their password if they forget it. 2. The user can log in using third-party authentication services.	

Storage Search & Listing

Task 2 - FR2	**Storage Search - UC2, UC3, PR1, PR2, QR2, QR3, QR4, C3, C2**
Purpose	Supports the renters to search for and view detailed information about available storage spaces based on size, cost, location, type, and rental duration criteria.
Trigger	The renter has accessed the search page for available storage spaces.
Pre-condition	The renter must be logged into the platform.
Frequency	Renters may search for storage spaces several times a day.
Critical	During peak moving seasons.
Sub-tasks: 1. Enter search criteria. 2. Show a list of available storage spaces that match the criteria. 3. Select a storage space and view detailed information about it.	
Variants:	

1. Renters can save search criteria for future searches. 2. Renters can view recently searched storage spaces for quicker access.

Task 3 - FR3	**Listing - UC1, PR1, PR2, QR2, QR3, C1, QR4, C3, C2**
Purpose	Supports the storage owners to list their available storage spaces, providing details such as size, availability, location, and cost.
Trigger	The storage owner has logged in and has accessed the page to list a new storage space.
Pre-condition	The storage owner must be logged into the platform.
Frequency	The storage owners may list new storage spaces whenever they have available space.
Critical	During university move-in seasons.
Sub-tasks: 1. Enter information on storage space, including size, location, cost, availability, and so on. 2. Upload photos and additional descriptions for the storage space. 3. Submit the listing for review and publishing.	
Variants: 1. The storage owner can save the edits as a draft and continue editing later.	

Communication & Interaction

Task 4 - FR4	**Chatting system - UC4, PR1, PR2, QR1, QR2, QR3, C3**
Purpose	Supports renters and storage owners to communicate directly through a messaging feature within the platform to discuss rental details.
Trigger	A renter has selected a storage space and initiated contact with the owner.
Pre-condition	Both the renter and the storage owner must be logged into the platform. The renter has viewed a specific storage space and wishes to inquire about further information.
Frequency	Renters and storage owners can communicate multiple times a day.
Critical	During university move-in seasons.
Sub-tasks: 1. The renter sends a message to inquire about the selected storage space. 2. The storage owner receives the message and responds to the inquiry with the necessary details. 3. Both parties can exchange messages until rental details are confirmed.	

Note: This document's grammar and formatting were checked using ChatGPT in some of the sections.

Variants: 1. Users can set notification preferences for new messages. 2. The chatting system can support attachments such as pictures.

Payment & Financial Transactions

Task 5 - FR5	**Payment & Financial Transactions - UC7, UC11, PR1, PR2, QR1, QR2, QR3, PR3, C2, C3**
Purpose	Supports renters to make secure payments through multiple integrated payment systems, and to view the transaction history.
Trigger	The renter has selected a storage space and proceeds to the payment to complete the rental transaction or the renter accesses the view transaction history page.
Pre-condition	The renter must be logged into the platform.
Frequency	Renters can make payments or view the transaction history several times a day.
Critical	During university move-in seasons.
Sub-tasks: 1. Select the preferred payment method. 2. Enter detailed information and confirm payment. 3. Receive a confirmation message of successful payment. 4. View past transaction history.	
Variants: 1. Renters can receive notifications for successful payments or failed transactions.	

Rental History & Management

Task 6 - FR6	**Rental History - UC6, PR2, QR2, QR3, QR4, C3**
Purpose	Help users (both renters and owners) view and track their rental history, including ongoing and past transactions.
Trigger	When someone who bought before clicked the history button
Pre-condition	The user has at least one completed or ongoing rental transaction in their account.
Frequency	Estimated 1-2 views per active user per month.
Critical	When facing dispute resolution, refund requests, or issues related to rental agreements (e.g. missed payments, incorrect transaction records).

Note: This document's grammar and formatting were checked using ChatGPT in some of the sections.

Sub-tasks: 1. Retrieves user’s rental history from our database. 2. present and visualize the data 3. Allow users to filter the transactions by date, storage location, or transaction status
Variants: 1. If a user clicks the "History" button but has no prior transactions, display a message indicating they have no history yet. 2. iIf users have a large number of transactions, then the system should implement pagination or a load-more option to manage performance and display.

Task 7 - FR7	**Management Of Storage Spaces- UC9, PR1, PR2, QR2, QR3, QR4, C2, C3**
Purpose	Storage owners can edit, update, and delete information about their posted storage spaces
Trigger	when a storage owner selects one of their posted storage spaces from the management dashboard and chooses to either edit, update, or delete the listing.
Pre-condition	The owner has one or more storage spaces listed on the platform.
Frequency	Estimated that this action could occur several times per month per active storage owner.
Critical	When updating or removing storage listings, especially if a space is no longer available or changes in pricing affect current renters.
Sub-tasks: 1. Get storage listing data from our database. 2. The owner updates the information as needed 3. Once the updates are made, the system must validate the changes and save them to the database.	
Variants: 1. Deleted listings are archived for a set period in case the owner wants to restore or review old listings.	

Task 8 - FR8	**Administrators Management - UC5, PR2, QR2, QR3, C1, QR4, C3**
Purpose	Administrators can review and moderate information from storage owners and renters.
Trigger	When an administrator receives a report of inappropriate content or behavior, or during routine reviews of storage listings and user interactions on the platform
Pre-condition	There is either flagged content (by users or automated systems) or content that requires periodic review.

Note: This document's grammar and formatting were checked using ChatGPT in some of the sections.

Frequency	Estimated that administrators review content daily, especially as the user base grows.
Critical	Maintaining the platform's integrity, ensuring that users feel safe and that content complies with legal and platform policies.
Sub-tasks: 1. The administrator accesses flagged or reported content. 2. Depending on the findings, the administrator can: ● Edit the content (e.g., remove inappropriate words or phrases). ● Flag content for further review. ● Temporarily hide or remove the listing or message. ● Ban or suspend the user if they are a repeat offender. 3.After taking action, the administrator must notify the affected users	
Variants: 1. Users who repeatedly violate policies are flagged, and stricter measures (e.g., permanent ban) may be implemented. 2. In cases where reports are found to be false or malicious, the administrator may need to warn or penalize users who abuse the reporting system.	

Rating & Review System

Task 9 - FR9	**Rating System - UC8, PR2, QR2, QR3, C3**
Purpose	Enable both renters and storage owners to rate and review each other after a rental transaction. This functionality aims to build trust and transparency on the platform
Trigger	Once a rental transaction is completed
Pre-condition	The platform sends an automated prompt or notification to remind users to leave a review.
Frequency	After each completed rental transaction.
Critical	Critical for resolving disputes, as reviews can help identify problematic behavior or recurring issues with specific users.
Sub-tasks: 1. After a rental is complete, the system sends a prompt to both the renter and the storage owner. 2. Users can rate the other party on a scale. 3. Once both the rating and review are submitted, the system validates the input and records it in the platform's database. 4.The rating and review are publicly displayed on the user's profile.	
Variants:	

1. If a user does not submit a review within a set period (e.g., 7 days), the system may send a reminder or mark the transaction as “unrated.” 2. The platform may allow users to leave reviews anonymously to ensure honest feedback without fear of retaliation.

Security

Task 10 - FR10	**Security - UC12,UC13, PR2, QR1, QR2, QR3, C1, C2, C3**
Purpose	The system should offer customers the option to purchase insurance for their belongings stored in the facility**.**
Trigger	When a renter books a storage space
Pre-condition	The renter is in the process of booking or finalizing their storage space.
Frequency	Every time a customer completes a storage booking
Critical	When users storing valuable or sensitive items
Sub-tasks: 1. During the booking process, the system presents an option to add insurance coverage for the stored items. 2. The system fetches real-time insurance quotes from partnered insurance providers. 3. The total cost of the rental is recalculated, including the insurance fee, and the renter is shown the updated price. 4. Once the transaction is completed, the system generates a digital insurance policy for the renter, outlining the coverage terms.	
Variants: 1. Renters may be allowed to purchase insurance at a later time, before the rental period begins, through their account management page. 2. For high-value items, the system could automatically recommend insurance	

Delivery

Task 11 - FR11	**Delivery - UC14, PR2, QR2, QR3, QR4, C3**
Purpose	The system should be able to track and provide real-time delivery updates to the user.
Trigger	When a delivery is initiated
Pre-condition	The user has scheduled a delivery through the platform.
Frequency	Every time a delivery is in progress.

Critical	Critical for user experience and trust, especially when valuable or sensitive items are being delivered.
Sub-tasks: 1. Once the delivery is scheduled, the system connects with the logistics provider's tracking system and starts receiving real-time data 2. The system displays real-time updates to the user. 3. The system sends push notifications, SMS, or email alerts to the user at key points. 4. Once the delivery is completed, the system marks the delivery as "Delivered" and prompts the user to confirm they've received their items.	
Variants: 1. If the delivery fails (e.g., the user isn't available to receive the items), the system reschedules the delivery and notifies the user of the new delivery date and time.	

Proposed prioritization

100$ Test

In this section, the functional requirements are prioritized by performing the 100$ test method (**see Table 3**). The final prioritization was derived by distributing 100 points across all requirements according to their importance. The stakeholders were asked to allocate their points based on what they found most important. Below is the final prioritization based on the combined results from all stakeholders.

Stakeholder	FR1	FR2	FR3	FR4	FR5	FR6	FR7	FR9	FR10	FR11
Storage Owners	25	10	20	5	15	5	10	5	3	2
Renters	20	15	10	8	10	7	5	5	10	10
Platform Administrators	30	8	15	5	10	5	7	5	10	5
Total Amount	**75**	**33**	**45**	**18**	**35**	**17**	**22**	**15**	**23**	**17**

Table 3. Prioritization Table using 100$ Test

Note: This document's grammar and formatting were checked using ChatGPT in some of the sections.

Final Prioritization (Based on Total Points):

1. User Authentication & Accounts (75 points)
2. Listing (45 points)
3. Payment & Financial Transactions (35 points)
4. Storage Search (33 points)
5. Management of Storage Spaces (22 points)
6. Security (Insurance Options) (23 points)
7. Rental History (17 points)
8. Delivery Tracking (17 points)
9. Chatting System (18 points)
10. Rating & Review System (15 points)

Analytical Hierarchy Process (AHP)

Here we choose to use the Analytical Hierarchy Process to prioritize functional requirements (**see Table 4**). The criteria selected for comparing each functional requirement are user experience and development costs. In this process, we assume both criteria are equally important.

User Experience-Based Comparison

The first comparison is based on user experience, and the following is the corresponding priority matrix:

	FR1	FR2	FR3	FR4	FR5	FR6	FR7	FR8	FR9	FR10	FR11
FR1	1	1/7	1/7	1/5	1	1	1/7	1/3	1/3	1	1
FR2	7	1	1	3	5	5	1	3	3	5	5
FR3	7	1	1	3	5	5	1	3	3	5	5
FR4	5	1/3	1/3	1	3	3	1/3	1	1	1	1
FR5	1	1/3	1/5	1/5	1	3	1/5	1/3	1/3	1	1
FR6	1	1/5	1/5	1/3	1/3	1	1/5	1/3	1/3	1/3	1/3
FR7	7	1	1	3	5	5	1	1	1	5	5
FR8	3	1/3	1/3	1	3	3	1	1	1	5	5
FR9	3	1/3	1/3	1	3	3	1	1	1	3	3
FR10	1	1/5	1/5	1	1	3	1/5	1/5	1/3	1	1
FR11	1	1/5	1/5	1	1	3	1/5	1/5	1/3	1	1

Table 4. Prioritization Table using AHP

Note: This document's grammar and formatting were checked using ChatGPT in some of the sections.

The resulting priority vector based on user experience is as follows:
FR1: 0.030
FR2: 0.203
FR3: 0.203
FR4: 0.077
FR5: 0.041
FR6: 0.027
FR7: 0.171
FR8: 0.082
FR9: 0.097
FR10: 0.035
FR11: 0.035
The Consistency Ratio for this comparison is 0.004, which is less than 0.1, indicating that the comparisons are consistent.

Development Costs-Based Comparison
Next, the comparison is made based on development costs, and the following is the corresponding priority matrix:

	FR1	FR2	FR3	FR4	FR5	FR6	FR7	FR8	FR9	FR10	FR11
FR1	1	1/3	1	3	5	3	5	7	6	5	5
FR2	3	1	2	4	3	3	5	6	5	4	5
FR3	1	1/2	1	3	2	2	4	5	4	4	3
FR4	1/3	1/4	1/3	1	3	2	3	4	3	3	2
FR5	1/5	1/3	1/2	1/3	1	3	2	4	2	3	2
FR6	1/3	1/3	1/2	1/2	1/3	1	2	3	2	2	2
FR7	1/5	1/5	1/4	1/3	1/2	1/2	1	2	2	2	2
FR8	1/7	1/6	1/5	1/4	1/4	1/3	1/2	1	3	3	2
FR9	1/6	1/5	1/4	1/3	1/2	1/2	1/2	1/3	1	2	2
FR10	1/5	1/4	1/4	1/3	1/3	1/2	1/2	1/3	1/2	1	2
FR11	1/5	1/5	1/3	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1

Table 5. Development Costs-Based Comparison

The resulting priority vector based on development costs is as follows:

FR1: 0.184

FR2: 0.236

FR3: 0.154

FR4: 0.098

FR5: 0.079

FR6: 0.063

FR7: 0.046

FR8: 0.043

FR9: 0.034

FR10: 0.032

FR11: 0.030

The Consistency Ratio for this comparison is 0.077, which is less than 0.1, indicating that the comparisons are consistent.

Since we assume that user experience and development costs are equally important, the priority weights for each functional requirement are aggregated by averaging the two priority vectors.

Taking into account the correlation between functions,

The final prioritization is as follows:

1. Log in, create accounts, and verify identity
2. Storage Search
3. Listing
4. Management of Storage Spaces
5. Payment & Financial Transactions
6. Rating & Review System
7. Administrators Management
8. Chatting system
9. Rental History
10. Security
11. Delivery

Prioritized Non-functional Requirements

1. **Security** must be prioritized, particularly in storing sensitive user information like payment data and verifying user identities.
2. The system must be **affordable** to develop and maintain, targeting students and individuals moving to new places as primary users. It must also comply with GDPR due to the collection of personal data.
3. The system must handle **real-time** searches and **updates**, and ensure that storage space availability is always current.
4. The system should have **high concurrency processing** capability to ensure stable operation when used by multiple people simultaneously.
5. The system should be compatible and support running on **different devices**.

Note: This document's grammar and formatting were checked using ChatGPT in some of the sections.

6. The user interface should be **simple** and **easy** to understand so that the user can learn how to use the system in a short time.

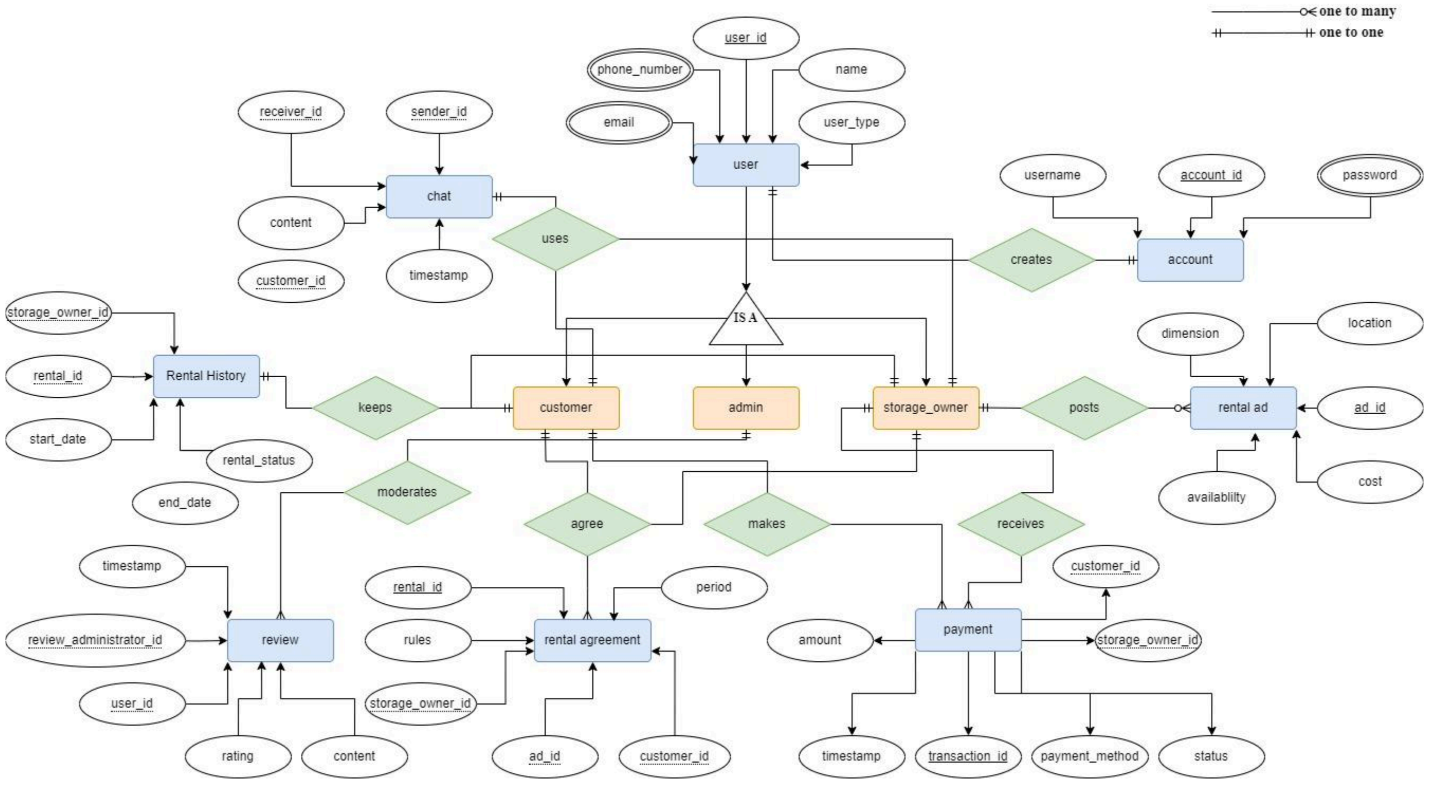
The **top priority** is security, as it is critical for protecting sensitive user data, including payment information and identity verification. Any breach in security would directly impact user trust and could lead to legal repercussions, which will lead to serious problems. **Affordability and GDPR compliance** are ranked second because they ensure that the system can be sustained financially and legally.

Real-time searches and updates are the third most important, as they form the core functionality of the platform—keeping storage availability current. Following this, **high concurrency processing** ensures that the system can handle multiple users at the same time without performance degradation, which is important but can be optimized based on actual usage levels.

Device compatibility comes next, as it improves the user experience by allowing the system to run on multiple devices. However, it does not directly affect the system's core operations as much as the other requirements. Lastly, **user interface simplicity** is important for ease of use but is less critical than the underlying performance and security functionalities. A less intuitive interface can still function if the core system is secure and performs well, but the reverse would not be true.

Data Requirements

ER diagram



Note: This document's grammar and formatting were checked using ChatGPT in some of the sections.

Fig 3. ER-model depicting how the entities (blue rectangles), with their attributes, can be modeled through relationships (shown with green diamond shapes).

User Data:

- **DR1:** Information on users such as names, contact details like phone number and email, and user type (customer, admin or renter).
- **DR2:** User accounts including username and password for logging in.
- **DR3**: Chat information including message content, sender ID, receiver ID and send timestamp.
- **DR4:** Rental history includes rental ID, start and end dates, renter and customer ID, and rental status.

Rental Storage Data:

- **DR5:** Rental Ads can be posted by renters, including location, dimension, cost and availability.
- **DR6:** Rental agreement details, including renter ID, customer ID, Ad ID, rules, rent period and other attributes.
- **DR7:** Administrator review information including review timestamp, rating results, user ID and review administrator ID.

Payment Data:

- **DR8:** Payment transaction information includes transaction ID, transaction amount, transaction timestamp, renter and customer IDs, payment method, and payment status.

Data directory

DR1
Class: User A user can be a renter, storage owner, or administrator.
Methods: **CreateAccount():** Creates a new account for the user. **UpdateProfile(name, email, phone_number):** Allows users to update their personal information.
Attributes: **User ID:** Unique identifier for the user. **Name:** Full name of the user. **Email:** Email address of the user. **Phone Number:** Contact phone number of the user. **User Type:** Specifies whether the user is a customer, renter, or administrator.

Table 6. Data dictionary for DR1

DR2

Note: This document's grammar and formatting were checked using ChatGPT in some of the sections.

Class: Account User account information, including credentials for login.
Methods: **ChangePassword(oldPassword, newPassword)**: Allows users to change their account password. Login(email, password): Users can login with their email and password. UpdateProfile(name, email, phone_number): Allows users to update their personal information.
Attributes: **Account ID**: Unique identifier for the account. **Username**: The username used by the user to log in. **Password**: Password for account access.

Table 7: Data dictionary for DR2

DR3
Class: Rental Ad Details of storage spaces posted by storage owners for rental.
Methods: **PostAd(location, dimension, cost)**: Allows storage owners to post a new rental ad. **UpdateAd(adID, newInformation)**: Updates the information of an existing ad. **DeleteAd(adID)**: Deletes a rental ad from the platform.
Attributes: **Ad ID**: Unique identifier for the rental ad. **Dimension**: Size of the storage space. **Location**: Geographic location of the storage space. **Cost**: The rental price for the storage space. **Availability**: Indicates whether the space is available or rented out.

Table 8: Data dictionary for DR3

DR4
Class: Payment Records payments made for rental agreements, capturing information about transactions between renters and storage owners.
Attributes: **Transaction ID**: Unique identifier for the payment transaction. **Renter ID**: The unique ID of the renter who made the payment. **Customer ID**: The unique ID of the storage owner receiving the payment. **Amount**: The total amount of the payment made. **Payment Method**: The method used for payment. **Status**: The current status of the payment.

Note: This document's grammar and formatting were checked using ChatGPT in some of the sections.

Timestamp: Date and time when the payment was processed.

Table 9: Data dictionary for DR4

DR5
Class: Rental Agreement The legal agreement between a renter and a storage owner regarding the use of storage space.
Methods: **CreateAgreement(renterID, customerID, adID, period, rules):** Creates a new rental agreement between a renter and a storage owner based on the provided storage listing. **UpdateAgreement(agreementID, newPeriod, newRules)**: Updates the rental agreement details.
Attributes: **Rental ID**: Unique identifier for the rental agreement. **Renter ID**: Unique identifier for the renter in the agreement. **Customer ID**: Unique identifier for the storage owner in the agreement. **Ad ID**: Reference to the associated rental ad. **Period**: The period of the rental agreement. **Rules:** The terms and conditions agreed upon by both parties for the rental period.

Table 10: Data dictionary for DR5

DR6
Class: Review Feedback left by a user after a rental transaction is completed.
Attributes: **User ID**: Unique identifier of the user who left the review. **Review Administrator ID**: Unique identifier for the administrator moderating the review. **Rating**: A numerical rating provided by the user. **Content**: Textual content or feedback provided by the user. **Timestamp**: The date and time when the review was submitted or moderated.

Table 11: Data dictionary for DR6

DR7
Class: Rental History The rental history records details of past and current rentals, including essential information about each rental transaction.
Attributes: **Rental ID**: Unique identifier for the rental transaction. **Renter ID**: Unique identifier for the renter associated with the rental. **Start Date**: The start date of the rental period.

Note: This document's grammar and formatting were checked using ChatGPT in some of the sections.

End Date: The end date of the rental period. **Rental Status**: Indicates the current status of the rental.

Table 12: Data dictionary for DR7

DR8
Class: Chat The chat feature allows communication between renters and storage owners within the platform, enabling them to exchange messages regarding rental details.
Attributes: **Sender ID**: The unique ID of the user sending the message. **Receiver ID**: The unique ID of the user receiving the message. **Content**: The content being sent. **Timestamp**: The date and time when the message was sent.

Table 13: Data dictionary for DR8

Detailed Performance Requirements, Specific Quality Requirements, Constraints [W.I.P]

This section explains the non-functional requirements for the peer-to-peer storage space-sharing platform. These requirements will help the platform work well, support business goals, and make sure everyone involved is happy with it.

Performance Requirements

The platform should operate seamlessly across Web, iOS, and Android with minimal downtime or latency. The platform should never feel like it's running slow hence both static and dynamic pages of the website should load quickly. Here we can find more detailed requirements

PR1. Search and Listing Speed: The system must handle real-time searches and updates, and ensure that storage space availability is always current. Results should show up in less than 2 seconds, even when many people use the platform. **(mentioned in previous sections but detailed here)**

PR3. Payment Processing: Payments should finish in minimum time assured by the provider.

PR4. Data Updates: Any changes users make to their listings (like posting, editing, or deleting) should happen immediately.

Specific Quality Requirements

This section covers the key quality aspects the platform must meet to ensure it works properly, is easy to use, and keeps user data secure.

QR4. Correctness and Reliability: The platform must work correctly in all its features. For example, when a user rents a storage space, the system must register the correct details for the renter, the storage space, and the rental period.

QR2. Usability: The platform should be very easy to use. Users should be able to find and list storage spaces without getting confused. **(mentioned in previous sections but detailed here)**

QR1. Security: Security should be the top priority, especially when handling sensitive user information like payment details and confirming user identities. **(mentioned in previous sections but detailed here)**

QR3. Cross-Platform Compatibility: The platform must work well on web browsers, iOS, and Android, so users can have a smooth experience no matter what device they use, with basic requirements. **(mentioned in previous sections but detailed here)**

- **Smartphones and Tablets**:
    - **OS**: Android 9.0 or iOS 12+
    - **RAM**: 2 GB
    - **Storage**: 500 MB available
    - **Browser**: Chrome, Safari, or Firefox
    - **Network**: 4G or Wi-Fi
- **Laptops and Desktops**:
    - **OS**: Windows 10, macOS 10.13+, Linux (Ubuntu 18.04+)
    - **RAM**: 4 GB
    - **Storage**: 1 GB available
    - **Browser**: Chrome, Firefox, Edge, or Safari
    - **Network**: Broadband (5 Mbps+)

Quality grid

Quality Factors	Critical	Important	As Usual	Unimportant	Ignore
Security	X				
Usability	X				
Performance		X			
Reliability		X			
Maintainability		X			

Note: This document's grammar and formatting were checked using ChatGPT in some of the sections.

Quality Factors	Critical	Important	As Usual	Unimportant	Ignore
User interface simplicity			X		

Table 14. Quality Grid

Concerns:

1. Security: GDPR compliance must be ensured for user data protection, especially payments.
2. Usability: The system should be intuitive, as it caters to a wide audience (students, general renters).
3. Performance: Real-time search and high concurrency support are crucial during peak usage times.
4. Reliability: Consistent uptime is necessary during high-demand periods.
5. user interface simplicity: important for ease of use but is less critical than the underlying performance and security functionalities

Constraint

This section explains the limits the platform needs to follow.

C2. Budget Constraints: Revenue will primarily come from users in form of percentage charges, helping to ensure the platform remains accessible.

C1. Storage Regulations: The platform must follow local laws about renting storage spaces. **(mentioned in previous sections but detailed here)**

C3. Technical Limitations: The platform should work well on older devices or slower internet connections without needing big upgrades.

3. System Requirements

This section goes into more detail about how the system will work and what it needs to do. We will explain the key features and technical requirements that are important for the platform. These are the specific things the system must handle to work well and meet the needs of users, like students who want to rent storage or offer their own space.

3.1 Detailed System Requirements

Here are the specific features the system needs to have:

- **SR1: User Accounts and Login** (Related to **FR1**)
  - The system shall allow users to sign up, log in, and verify their identity using an email or third-party authentication services like Google (FR1).

Note: This document's grammar and formatting were checked using ChatGPT in some of the sections.

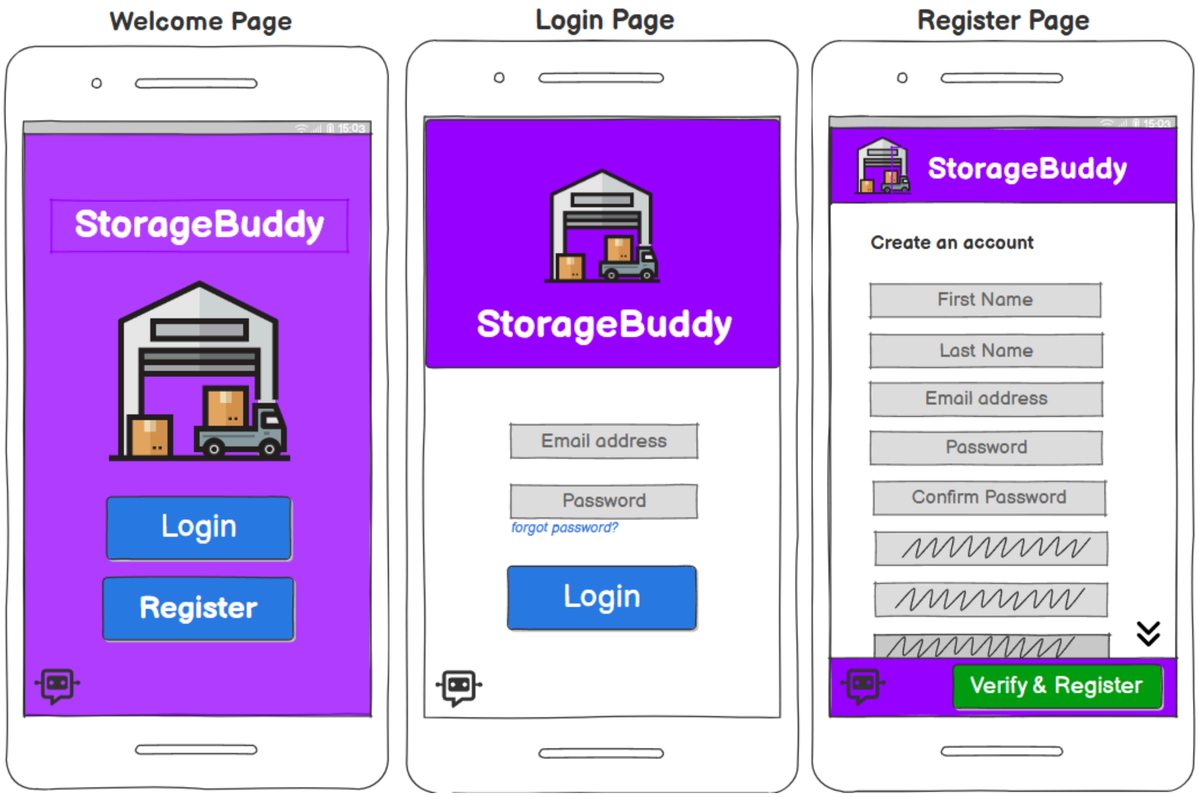
    - The system shall provide password recovery options using email or security questions (FR1).
    - The system shall verify user identity through email confirmation or a third-party service (FR1).
- **SR2: Listing and Searching for Storage** (Related to **FR2**, **FR3**)
    - Storage owners shall be able to list available storage spaces, providing details such as size, location, cost, and photos (FR3).
    - Renters shall be able to search for storage spaces using filters like size, location, cost, and availability, with search results displayed in under 2 seconds (FR2).
    - The system must ensure real-time updates of storage availability (FR2).
- **SR3: Payments** (Related to **FR5**)
    - The system shall support secure payments via credit cards, PayPal, and other methods using HTTPS encryption (FR5).
    - Payments shall be processed within 5 seconds, with confirmation provided to both the renter and the owner (FR5).
    - Users shall be able to view their transaction history (FR5).
- **SR4: Messaging Between Users** (Related to **FR4**)
    - The system shall provide a secure messaging feature allowing renters and owners to communicate directly (FR4).
    - Users shall be able to send attachments (e.g., images, documents) via the messaging platform (FR4).
- **SR5: Ratings and Reviews** (Related to **FR9**)
    - After a transaction, renters and owners shall be able to rate and review each other (FR9).
    - Reviews shall be allowed anonymously to promote honest feedback (FR9).
    - Ratings and reviews shall be displayed on user profiles (FR9).
- **SR6: Delivery Tracking** (Related to **FR11**)
    - The system shall integrate with third-party delivery services to provide real-time tracking information to users (FR11).
    - Users shall receive notifications (via email, SMS, or the app) regarding their delivery status (FR11).
- **SR7: Security** (Related to **FR10**)
    - All data transfers between clients and servers shall be encrypted using HTTPS (FR10).
    - Passwords shall be stored securely using hashing and salting methods (FR10).
    - The system shall comply with GDPR, allowing users to request deletion or anonymization of their personal data (FR10).
- **SR8: Performance and Scalability** (Related to **FR1**, **FR2**, **FR3**, **FR5**)
    - The system shall handle up to 10,000 concurrent users without performance degradation (FR1, FR2, FR3, FR5).
    - It shall be scalable to meet peak demand during high-traffic periods like university move-ins (FR1, FR5).
    - The system must recover quickly from server failures to avoid data loss.
- **SR9: Platform Compatibility** (Related to **FR1**, **FR2**, **FR3**, **FR5**)
    - The system shall work as a mobile-friendly web application optimized for Chrome, Firefox, Safari, and Edge (FR1, FR2, FR3, FR5).

Note: This document's grammar and formatting were checked using ChatGPT in some of the sections.

    - It shall also be available as an app for both Android and iOS, with full functionality, including searching, listing, payments, and messaging (FR1, FR2, FR3, FR5).
    - The platform shall be responsive to various screen sizes to ensure ease of use on both mobile and desktop devices.
- **SR10: Data Privacy and GDPR Compliance** (Related to **FR10**)
    - The system must comply with GDPR regulations, allowing users to request the deletion or anonymization of their personal data.
    - Any third-party services integrated into the platform must also follow GDPR rules to ensure full data protection.
- **SR11: Accessibility** (Related to **FR1**, **FR2**)
    - The system must be designed to be accessible for all users, including those with visual impairments or color blindness.
    - The platform shall use responsive design to ensure that it works well on both desktop and mobile devices, with features such as alt text for images and readable fonts.

- **SR12: System Architecture** (Related to **FR1**, **FR2**, **FR5**)
    - The system must use a microservice-based architecture, ensuring that backend services are modular, independently deployable, and scalable.
    - The platform shall have a RESTful API that handles requests from both web and mobile clients, enabling smooth interaction between frontend and backend components.
    - The platform must support horizontal scaling to handle increased traffic during peak times, such as university move-in periods.

3.2 UI Prototype



Note: This document's grammar and formatting were checked using ChatGPT in some of the sections.

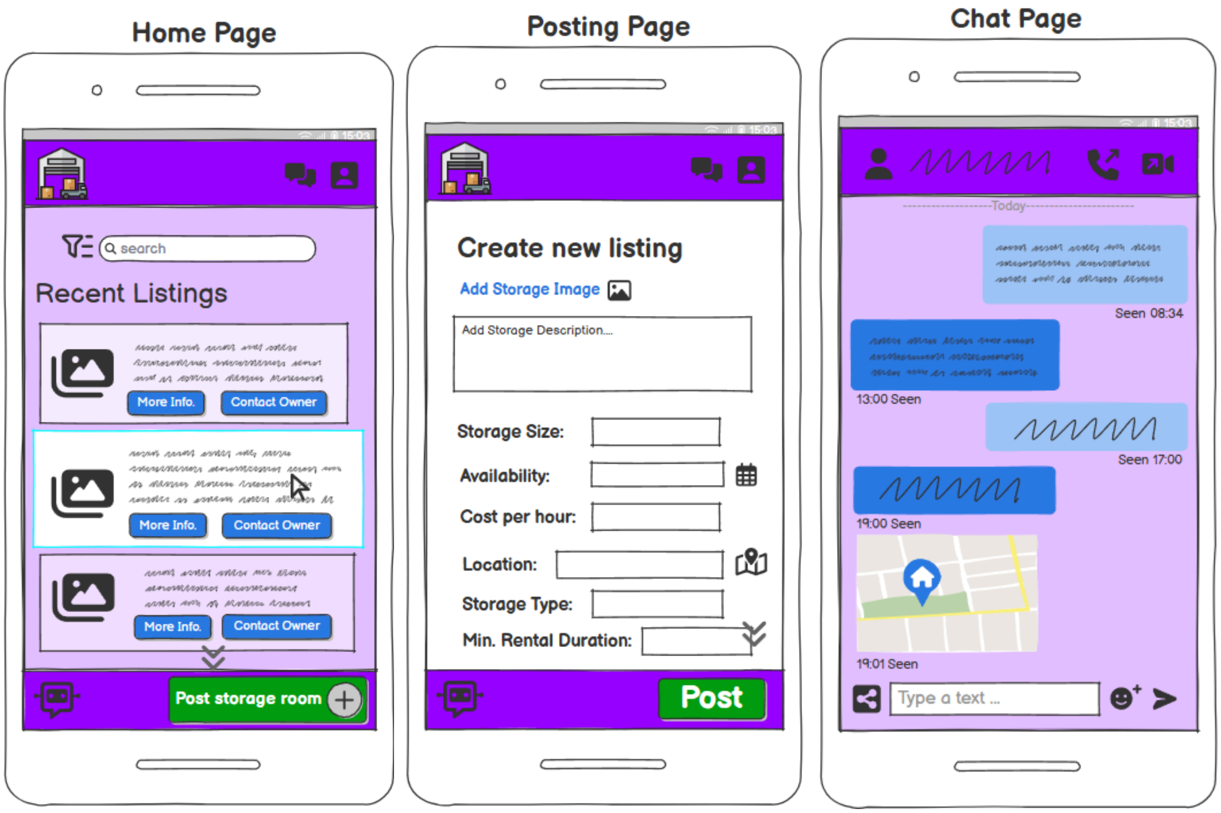


Fig 4. UI Prototypes

Subset of functionalities covered by the prototypes: Login, Registration, View/Post/Rent/Search Storage Listing, Chat Functionality.

3.2 Acceptance Test

The main purpose of the acceptance test for our project is to validate that the product corresponds to the needs of users and stakeholders and is ready to launch in the market. To stand out in the market, verifying that the product operates in real-life circumstances and offers end-users solutions based on their needs is important. Without a User Acceptance Test (UAT) a company might miss out on some important flaws or malfunctions that can cause dissatisfaction for the users. The diagram below shows the key stages of the User Acceptance Test

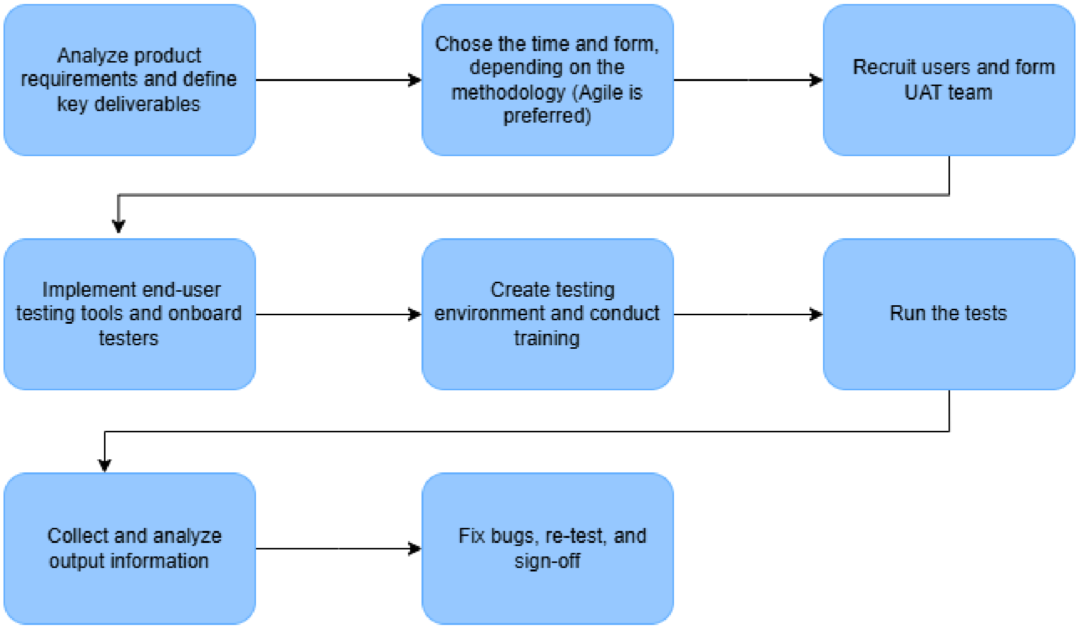

Analyze product requirements and define key deliverables
Chose the time and form, depending on the methodology (Agile is preferred)
Recruit users and form UAT team
Implement end-user testing tools and onboard testers
Create testing environment and conduct training
Run the tests
Collect and analyze output information
Fix bugs, re-test, and sign-off

Fig 5. The key stages of the User Acceptance Test

For our product, we formulated a UAT test plan to detect any flaws. Some of the essential test cases are mentioned below

1. User Authentication and Accounts

   Test Case 1.1: User Registration or Sign-Up
   - Scenario: A new user attempts to create an account using an email id.
   - Conditions: The individual should be on the StorageBuddy app registration page.
   - Steps:
     - The new user enters a valid email address, a password and other required information such as first name and surname.
     - The new user clicks on the “Verify and Register” button.
     - An email confirmation is sent to the user's email.
     - The new user clicks on the confirmation link in the email,
   - Expected Result: The user successfully creates an account and automatically logs in the account.

   Test Case 1.2: Password Recovery
   - Scenario: A registered user forgets the password and tries to recover it.
   - Condition: The user clicks “forget password?” on the Login Page.
   - Steps:
     - The user enters the registered email.
     - The user clicks on the “Submit” button.
     - The user receives a password reset link in the email.
     - The user clicks the reset link and enters a new password.
   - Expected Result: The password is successfully changed and the user can now log in again with the new password.
2. Storage Listings and Searching

   Test Case 2.1: Create a storage listing

Note: This document's grammar and formatting were checked using ChatGPT in some of the sections.

- Scenario: A storage owner creates a new listing to rent.
- Condition: The storage owner is logged in on the account
- Steps:
  - The storage owner goes to the “List Storage” page.
  - Fill in all required details (size of the storage, location, availability duration, price).
  - The storage owner adds some photos of the storage.
  - Clicks on the “Submit” button
- Expected Result: The storage listing should be created successfully and appear in the search result.

3. Payment Process

   Test Case 3.1: Payment Completion

   - Scenario: A user completes payment for a rental.
   - Condition: The user presses the “Continue and Pay” button to initiate payment.
   - Steps:
     - The user proceeds to the payment page.
     - The user selects a payment method (debit or credit card, Swish, Paypal etc).
     - Enter payment information.
     - Click on the “Confirm Payment” button.
   - Expected Result: Payment should be processed within a few seconds (depending on the internet connection speed) and the user should receive a confirmation by email or SMS.

4. Chatting System

   Test Case 4.1: Send a message

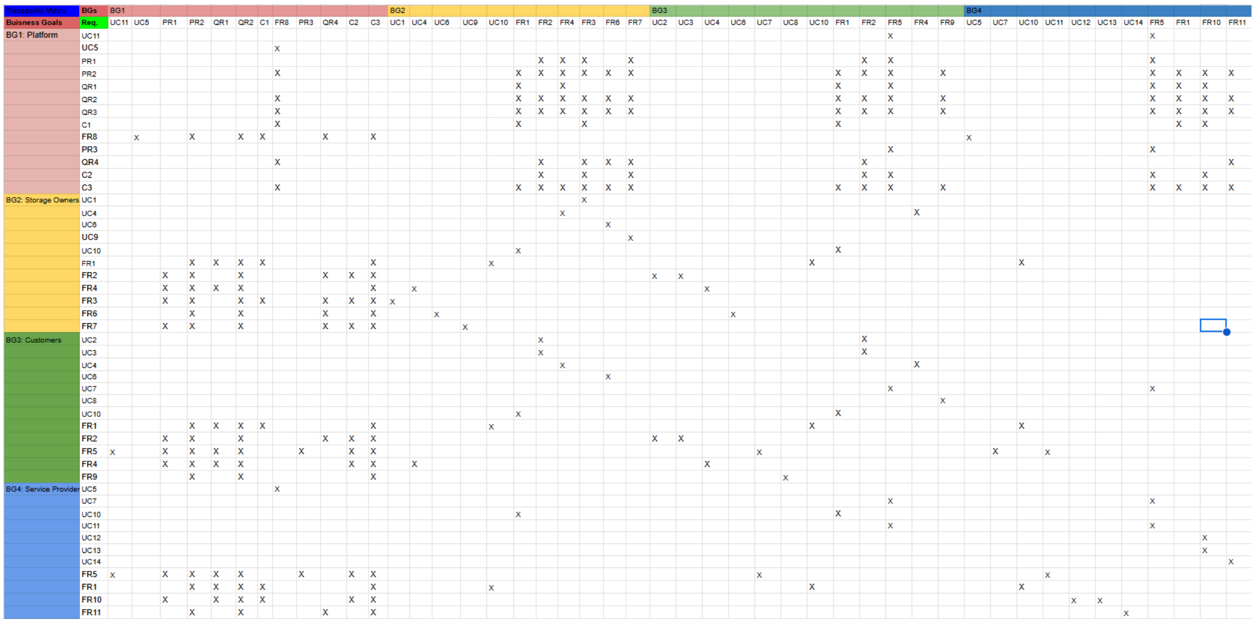
   - Scenario: A user sends a message to a storage owner
   - Condition: The user should be logged in to send and the storage owner should be logged in to see the message.
   - Steps:
     - The user views a storage listing and clicks “Contact Owner”
     - The user types a message and presses the “Send” button.
   - Expected Result: The message is sent and the receiver (storage owner) should receive a notification.

5. Performance and Scalability

   Test Case 5.1: Handle Concurrent Users

   - Scenario: We test with a JMeter to simulate 1000 concurrent users using the platform.
   - Steps:
     - Go through the necessary steps to create the simulation.
     - Generate a test report for future reference .
   - Expected Result: The platform should be able to handle the load without any performance degradation.

5. Traceability Matrix



Traceability Matrix	BGs	BG1												BG2											BG3												BG4										
Buisness Goals	Req.	UC11	UC5	PR1	PR2	QR1	QR2	C1	FR8	PR3	QR4	C2	C3	UC1	UC4	UC6	UC9	UC10	FR1	FR2	FR4	FR3	FR6	FR7	UC2	UC3	UC4	UC6	UC7	UC8	UC10	FR1	FR2	FR5	FR4	FR9	UC5	UC7	UC10	UC11	UC12	UC13	UC14	FR5	FR1	FR10	FR11
BG1: Platform	UC11																																	X										X			
	UC5								X																																						
	PR1																			X	X	X		X									X	X										X			
	PR2								X										X	X	X	X	X	X								X	X	X		X								X	X	X	X
	QR1																		X		X											X		X										X	X	X	
	QR2								X										X	X	X	X	X	X								X	X	X		X								X	X	X	X
	QR3								X										X	X	X	X	X	X								X	X	X		X								X	X	X	X
	C1								X										X			X										X													X	X	
	FR8		X		X		X	X			X		X																								X										
	PR3																																	X										X			
	QR4								X											X		X	X	X									X														X
	C2																			X		X		X									X	X										X		X	
	C3								X										X	X	X	X	X	X								X	X	X		X								X	X	X	X
BG2: Storage Owners	UC1																					X																									
	UC4																				X														X												
	UC6																						X																								
	UC9																							X																							
	UC10																		X													X															
	FR1				X	X	X	X					X					X													X								X								
	FR2			X	X		X				X	X	X												X	X																					
	FR4			X	X	X	X						X		X												X																				
	FR3			X	X		X	X			X	X	X	X																																	
	FR6				X		X				X		X			X												X																			
	FR7			X	X		X				X	X	X				X																														
BG3: Customers	UC2																			X													X														
	UC3																			X													X														
	UC4																				X														X												
	UC6																						X																								
	UC7																																	X										X			
	UC8																																			X											
	UC10																		X													X															
	FR1				X	X	X	X					X					X													X								X								
	FR2			X	X		X				X	X	X												X	X																					
	FR5	X		X	X	X	X			X		X	X																X									X		X							
	FR4			X	X	X	X					X	X		X												X																				
	FR9				X		X						X																	X																	
BG4: Service Provider	UC5								X																																						
	UC7																																	X										X			
	UC10																		X													X															
	UC11																																	X										X			
	UC12																																													X	
	UC13																																													X	
	UC14																																														X
	FR5	X		X	X	X	X			X		X	X																X											X							
	FR1				X	X	X	X					X					X													X								X								
	FR10			X		X	X	X				X	X																												X	X					
	FR11				X		X				X		X																														X				

Fig 6. Traceability Matrix

For better clarity, visit the following link: https://docs.google.com/spreadsheets/d/1DKfulrSWnRrmEAkvEUMF281SnEkWbAePF_4TizDXQAQ/edit?usp=sharing

Understanding:

The cross signs are mainly concentrated around business goal 2 and 3. Moderate concentration with Business goal 4. Little concentration with business goal 1. This could mean we are missing actions that could be mapped to business goal 1. Also, this diagram helped us to recognise some requirements we could’ve added.

Note: This document's grammar and formatting were checked using ChatGPT in some of the sections.