# Group 21- Experience Report

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

***Abstract*—In today's fast-paced world, individuals, particularly students, often face challenges in finding affordable and flexible storage solutions for their short-term or temporary needs. Traditional storage facilities tend to be costly and rigid, making them unsuitable for those with limited budgets. This project introduces a peer-to-peer storage platform aimed at addressing these issues by connecting individuals who need storage with those who have unused space.**

**The platform's primary goal is to provide a cost-efficient and flexible solution by allowing individuals to rent storage space directly from others. It is designed with students as the main target audience, offering them an alternative to expensive third-party storage services. On the other side, individuals with spare storage can list their available space, potentially earning extra income. Key features include secure identity verification, detailed storage space listings (with descriptions, photos, and pricing), and a robust search function to filter available storage options based on size, cost, and rental duration. Users can communicate directly within the platform, manage rental history, and choose from multiple payment methods. The platform is available on web, iOS, and Android, offering a seamless experience across devices. Additionally, a reputation and rating system ensures transparency, trust, and competition within the marketplace.**

**This peer-to-peer storage solution fosters community-driven exchanges, benefiting both renters and storage providers with its user-friendly interface and secure system.**

## I. Introduction

This project report outlines the learning outcomes, challenges faced, and group dynamics experienced during the development of the "Peer-to-Peer Storage Space Sharing Platform for Individuals." The system is designed to help individuals, particularly students, find affordable and temporary storage spaces from others with excess space, as an alternative to more expensive traditional storage solutions. The platform targets cost-efficiency and flexibility, primarily benefiting students with short-term storage needs.

## II. Techniques

### A. Elicitation Techniques

Requirements elicitation incorporates investigating and gathering information about system requirements from users, customers, and other stakeholders. This process must be conducted carefully to clearly understand the demands of all stakeholders. There are various techniques to elicit requirements, in the case of R1, we choose two of those techniques to understand the requirements of the stakeholders which are brainstorming and interviewing.

For R1, we initially interviewed some students as potential stakeholders and plan to extend the interview phase to gather inputs from the other stakeholders on the next version. The creative discussions within our group helped clarify the scope of the project. We learned the importance of exploring different perspectives to shape a well-rounded product. Interviewing a couple of potential customers , who were more familiar with renting storages, helped us to gain new insights and ideas that we wouldn't have considered otherwise. We expect to gather further insights that will enhance the platform's features by interviewing more potential customers.

### B. Creative Elicitation

Using creative techniques like "Assumption Busting" allowed us to explore the concept of platform in new ways. For example, while initially focusing on a mobile-first design, we considered web-based and hybrid app approaches. "Bright Sparks" helped us to analyze the requirements from other perspectives and made us realize that we could introduce a rating and reputation system to boost trust within the user community. The diversity in our group made the creative elicitation more effective, as the different perspectives allowed us to think from multiple angles.

### C. Brainstorming

In the case of elicitation, it is critical to leverage creativity to generate diverse and fresh ideas. We held several brainstorming sessions and applied a couple of creative techniques to develop the peer-to-peer storage platform. In particular, techniques like "Bright Sparks" and "Assumption Busting" paved the way for further adding more functionalities.

- **How Assumption Busting Helped:**
  1) Breaking conventional thinking: In the early stages, we assumed that the application should be mobile-first, considering most of the target audience are students who tend to rely mostly on smartphones to perform daily tasks, However, through assumption busting, we explored other alternatives like web-based or hybrid solutions, considering the possibility that some users might be more comfortable laptop or desktop.
  2) Shifting focus from cost efficiency: Initially, we assumed the platform's appeal was in providing an affordable storage solution. While this is our

primary focus, we later realised that users would prioritise security and convenience. This thought led to integrating complementary features like insurance, identity authentication etc to address the trustworthiness of the product.

- **How Bright Sparks Helped:**
  1) Identifying diverse audiences: During the Bright Sparks session, the discussion led to considering professional entertainers and their storage needs. While students are the primary target audience, the techniques encouraged thinking out of the box which led to recognising other groups, like touring musicians, might benefit from flexible storage solutions to store their instruments.
  2) Generating new ideas: If a high-profile user such as a musician or band wants to store expensive instruments, they would likely need assurances that the storage renter is trustworthy. This idea led to the consideration of a detailed rating system, where users could not only choose a rating score but also put a review to enforce reliability.

### D. Interview

Interviews are highly effective as they enable direct interaction with stakeholders, providing feedback and allowing an in-depth understanding of stakeholder's requirements. Initially, we interviewed a few students as stakeholders and their valuable feedback helped us to identify more requirements and validate existing demands. Each interview was conducted professionally by following the necessary protocols and consent. Several key points were addressed in the discussion:

- Security concerns: Stakeholders raised concerns about security measures, like whether there will be any protocol to ensure product safety, This concerns highlighted the necessity of addressing security measures in our service.
- Insurance coverage: Some stakeholders addressed their concerns about potential harm to their items and asked if insurance would cover such incidents. incidents. Thankfully, we've already planned for this by offering an optional insurance feature that allows customers to get coverage for their items up to a fixed limit.
- Pricing: stakeholders appreciated the low price option for the service, validating our pricing strategy that aligns with customer expectations making it a strong selling point.

## III. Specification and Learning Outcome

Creating use case and context diagrams provided a clearer vision of how our platform will function. These activities helped us apply requirement engineering practices to a real-world scenario.

### A. Context Diagram

Our context diagram has been useful in understanding how different actors, such as storage owners and renters, will interact with the system. This high-level view is helping us plan the core features of the platform. While depicting the context diagram, we began by focusing on the interactions between the system, the storage renter and the customer. We thought of the context in a real-life scenario where a customer rents storage from a storage owner. This helped us to state what information needs to be sent to the system and back to the user. Following the depiction of the renter and customer, we already had thought of other actors/systems that could facilitate the process of renting a storage. We included these actors in our context diagram and their corresponding interactions with the system. The context diagram also allowed us to identify new actors that could be integrated into the system, such as the insurance provider, which could offer coverage for belongings.

### B. Use Case Diagram

The use case diagram allowed us to map out the main functionalities of the system, such as user verification, listing storage spaces, and managing payments. While depicting the use case diagram, we collected the list of already established actors from our context diagram. We thought of features that could be included in our system and associated them with the corresponding actor. Moreover, we tried to depict how different use cases are related to each other using the `<<extends>>` and `<<includes>>` notation. The use case diagram helped us to identify potential edge cases like disputes over rental terms or payment issues.

### C. Quality grid

When starting the quality grid, we first needed to determine which quality factors were most relevant to our storage rental platform. Given the nature of our project, the team selected five key factors that we considered critical, important, or typical for a high-performing system: security, usability, performance, reliability, maintainability, and user interface simplicity. These factors were chosen based on their direct impact on the user experience and the overall success of the platform.

While filling out the grid, we consulted the performance requirements and specific quality requirements sections from our requirements document. This helped us determine which factors were already emphasized in our design, such as real-time search functionality (performance) and data protection (security). Reviewing the document allowed us to align our quality grid with the existing system requirements and focus on aspects that needed further refinement.

Finally, we ranked security as critical because of its importance in complying with GDPR and protecting sensitive data, especially payment information. Usability was marked as critical since the system caters to a wide audience, including students who may have varying levels of technical proficiency. Performance and reliability were ranked as Important but not critical, given the system's need for high concurrency during peak times, but with flexibility in optimizations as the user base grows. Lastly, user interface simplicity was deemed necessary but less important compared to the core functionalities of security and performance.

### D. Task Descriptions

During the process of R2, we chose Task Descriptions as a way to standardize and organize functional requirements. It is an intuitive and easy-to-understand way to define and demonstrate the functional requirements of our system quickly. This approach makes it easier to sort out the operational steps of our system and ensures that we can better plan our functional priorities. In addition, in this way, we are also better able to communicate with relevant stakeholders and ensure that functional requirements are clearly visible.
To improve efficiency, we divide tasks into functional areas, such as user login, storage search, payment, etc. This helps us to locate and handle specific tasks more easily during the design and development process. However, one of the biggest challenges in the task description process is to ensure that each task is detailed without being overly lengthy, especially when considering different variants of the task.
Through the writing of Task Descriptions, we have come to realize the importance of standardizing functional requirements for the success of a project. It will enable us to be clearer and more confident when communicating with stakeholders. It would have been nice to have the opportunity to validate the Task Descriptions with feedback from actual users to ensure that the Task Descriptions were more closely aligned with the actual requirements.

### E. Performance & Quality Requirements

We specified performance criteria, such as a fast response time for search queries and real-time communication between users. Non-functional aspects, like scalability for an increasing user base, will need further validation with actual usage data.

### F. Data Requirements

The data requirements section has been divided into three main categories that we believe are the most important to the system: user data, storage data, and payment data based on the use cases and core functionalities. In addition to this, we have envisaged specific usage scenarios to refine the data requirements. These data requirements were defined internally but will need to be adjusted as we gather more insights from real users.

### G. Data Dictionary

We chose to write the Data Dictionary because it is a crucial part of the overall documentation. It provides a standardized way of defining and describing every data element involved in the system.
In the process of writing it, I encountered several challenges, among which sometimes the boundaries may not be clear enough when defining certain classes. For example, certain data fields may belong to more than one class, and then we need to carefully consider where the data belongs and what it is used for to ensure that there are no conflicts. By writing the Data Dictionary, we gained a deeper understanding of the data flow of the whole system.

### H. Prioritization

Once all the requirements were defined, the next step involved prioritizing them. The goal of prioritization is to logically organize the development process, ensuring that the most essential functional requirements are built first to address the users' key needs. Since the importance of each functional requirement differs depending on factors such as user experience, development cost, and business objectives, we carefully considered our approach to prioritization.
In our project, we used the 100 Dollar Test and AHP to prioritize the functional requirements. The combination of these two methods reflects the weighting of the requirements of different stakeholders while making a reasonable judgment on the relative importance of each feature.

### I. The 100 Dollar test

The 100 Dollar Test approach is simple and intuitive and allows stakeholders to easily participate in the prioritization discussion. 100 Dollar Test This method is simple and intuitive, allowing stakeholders to easily participate in prioritization discussions, and it expresses the relative importance of each functional requirement in a quantitative manner, making it easier to synthesize stakeholder views.
In this test, each stakeholder was given a budget of 100 points to allocate to different functional requirements. They allocated these budgets to the features they considered most important based on their individual needs and expectations. This process helped us to better understand the prioritization of the needs of different user groups.

### J. Analytical Hierarchy Process

AHP allows us to meticulously compare functional requirements across multiple dimensions, especially to find a balance between user experience and development costs. This approach helps us to combine multiple decision criteria to produce a more comprehensive prioritization result.
In the AHP ranking, we chose user experience and development cost as the main dimensions of comparison. When evaluating user experience, we focused on how each feature would enhance convenience and improve user satisfaction with the platform. On the other hand, for development cost, we assessed the complexity of the technology, the time required, and the resources needed to implement each feature. This approach allowed us to prioritize the features that are most important to our users while also considering the project's feasibility. By doing this, we could ensure that the development process would proceed efficiently within the constraints of our available time and resources.

### K. Trade-offs between $100 test and AHP

As mentioned previously, the $100 test is a simpler and more intuitive method to prioritize requirements and takes less time. However, compared to AHP, $100 test introduces personal biases since it relies on the personal perspectives of each stakeholder when allocating $100 dollar to the set of requirements. Additionally, this method becomes more

complicated to allocate $100 dollars as the number of requirements and interdependent requirements grow.

On the other hand, AHP provides a systematic approach to prioritize requirements which is more useful in our case since our requirement specification was more complex and involve multiple stakeholders. However, as a drawback, AHP was more time consuming and mathematically intensive. Additionally, we needed to be more trained in the domain to make more accurate assumptions to obtain more precise results.

In conclusion, the $100 test would be a better technique to use, especially when the scope of the project is small or in the early stages of the project where requirements need to be prioritized faster. AHP is more suited to complex projects with interdependent requirements providing a more structured and accurate set of prioritized requirements that justifies the additional time investment. Moreover, combining both $100 test and AHP would also be effective when prioritizing requirements.

## IV. System Requirements

The process of gathering system requirements involved talking to key people to find out what the system needs to do. We held meetings, asked questions, and looked at the current systems to understand what users wanted. We learned the system needs to be fast, able to grow, and secure. However, it was hard to decide which features were the most important because different people had different needs. This showed us how important it is to communicate clearly and make changes as we go to meet everyone's needs.

### *A. UI Prototyping*

The screen designs were based on the functional requirements outlined in the previous sections. To better understand the user's flow through the app, we utilized an Activity Diagram, which helped us identify the necessary screens for design. Leveraging our prior knowledge of UX design principles and mobile app usage experience, we crafted an intuitive and user-friendly interface that is easy to navigate.

## V. Quality Assurance of Requirements

Throughout the duration of the project, we ensured the quality of our requirements according to certain criteria and depending on the type of requirement. Additionally, we also planned and documented the quality assurance process for both our requirements and those of our peer group, with details provided in our review report. In both the cases of our report and peer report, we followed the set criteria for ensuring quality assurance of the requirements specification.

Firstly, we made sure that all requirements had a unique identifier and can be traceable across different types of requirement and within the traceability matrix. While doing so, we double checked our traceability matrix to identify any missing requirements that could meet our business goals or recognise any unnecessary requirements for our project. We also took into consideration the peer review of our report to improve our requirements spec and fill any gaps in consistency and traceability. To this extent, the **Consistency Check** and **Content Check** techniques came in handy to maintain the quality assurance of our requirements specification. Similarly, we used the same V&V techniques and **Structural Check** to review our assigned peer-report for inconsistencies and missing content in their specification.

Moreover, we validated our requirements through UI prototypes and cross-checked it with possible stakeholders to make sure the requirements meet user needs. Our group also conducted acceptance testing, where several test cases were created and the process helped us identify gaps and make sure most of our requirements met quality standards.

Finally, we followed certain criteria to ensure the quality of our own and our peer's requirement specifications. These criteria included correctness, completeness, clarity, consistency, modifiability, verifiability, traceability, and prioritizability. However, an exception was made for Use Cases, which were structured as high-level User Stories designed to be flexible.

## VI. Group Dynamics

Working as a group, we learned to manage diverse schedules and utilize individual strengths effectively. Having team members from different backgrounds and fields helped broaden the discussion and generate creative ideas during the elicitation phase. For example, we can understand the needs of different markets and consumers from multiple perspectives, which helps us to make our products more responsive to the needs of the global marketplace.
We also implemented a team agreement early on to help manage communication and task distribution. The following demonstrates our way of working as a group:

- Teamwork
    1) Our team had people from different backgrounds, which gave us many new ideas.
    2) We used our different experiences to understand what different users need.
    3) We made a team agreement early to help with communication, sharing tasks, and solving problems.
- Organizing Tasks
    1) We gave tasks to each person based on their strengths, which helped keep us organized.
    2) Brainstorming sessions helped us come up with ideas and decide on the most important features.
    3) Clear roles made sure we worked efficiently and avoided repeating tasks.
- Communication
    1) We used Slack and Trello to stay in touch and share updates.

2) Weekly meetings helped us talk about progress and fix any issues.

## VII. Conclusion

For the final release, we have based our system requirements on group discussions and shared knowledge from lecture slides and our course book. In future work, we can validate these assumptions and our system with more user feedback to refine the platform and its functionality.