SpatiaList: a multi-user persistent post-it placement application in mixed-reality

Beyzanur Coban
ETHZ
Zurich, Switzerland
bcoban@ethz.ch

Zeno Hamers
ETHZ
Zurich, Switzerland
zhamers@ethz.ch

Deepana Ishtaweera
ETHZ
Zurich, Switzerland
dishtaweera@ethz.ch

Malo Ranzetti
ETHZ
Zurich, Switzerland
mranzetti@ethz.ch

Mihai Dusmanu
Microsoft Research
Zurich, Switzerland
mihai.dusmanu@microsoft.com

Patrick Misteli
Microsoft Research
Zurich, Switzerland
patrick.misteli@microsoft.com

Abstract

SpatiaList is a HoloLens 2 application designed for spatially persistent digital post-its in a Mixed Reality (MR) environment. This project, in collaboration with Microsoft, is intended to showcase the potential of MR technologies to enable multi-user collaboration scenarios. SpatiaList maintains consistency across sessions, space and users of MR objects, which is not implemented out-of-the-box in current MR devices. The system architecture is designed to support persistence of holographic post-its, the core component being Azure Spatial Anchors (ASA). A GUI was designed (for MR devices) and on the Web (for non-MR devices). A user study with 16 participants has been performed to evaluate the app's performance. Spatial and session consistency was observed. In stable internet conditions, multi-user teamwork scenarios could be effectively performed. The user study shows an overall positive sentiment, with key recommendations to provide more visual cues for the post-it creation step and to enable users to work on the same post-it simultaneously.

1. Introduction

MR technology, as per [8], transforms team interactions by offering heightened customizability, richer content, and seamless cross-device sharing. With the SpatiaList application, traditional post-it notes are made smarter. SpatiaList, a HoloLens 2 app, allows users to create and interact with persistent digital post-its in a Mixed Reality (MR) environment. The application, conducted in partnership with Microsoft, is designed to be spatially and session persistent, two properties that are crucial for collaboration in MR environments.

The app offers diverse customisation options: users can adjust post-it colours, text, and scale. Post-its can also be easily repositioned by a grabbing motion. Additionally, a gesture-based web interface was created, facilitating content transfer from conventional devices to the MR landscape. Details about the app's design and the implementation of these features are outlined in the method section of this paper.

To evaluate the user experience of the SpatiaList application, a user study was conducted with 16 participants. A succinct visual demonstration of the SpatiaList functionalities together with key suggestions from the user study can be found in the results section.

2. Background

The project aims to showcase how Mixed Reality (MR) enhances collaboration in multi-user, ideation scenarios. As per [1], HoloLens2 provides significant improvements to living environments and work efficiency. In these teamwork sessions, where digital elements (like holographic post-it notes) need to be shown in the same place for all users, spatial persistence is vital. The Azure Spatial Anchors (ASA) documentation shows the importance of spatially persistent hologram placement in multi-user cases [7].

As of now, maintaining consistency across time, space, users and sessions of MR objects is not implemented out-of-the-box in MR devices. This project builds on top of the ASA framework developed by Microsoft, which provides core mapping and localisation functionalities for MR projects. These functionalities are extended into a modular MR framework that can handle complex multi-user scenarios. By designing a user-friendly GUI both in Unity (for MR devices) and on the Web (for non-MR devices), this project aims to improve user experience.

3. Method

To achieve the spatial and session persistence of the post-its, the system architecture depicted in Figure 1 is implemented. It consists of three major components: Backend, HoloLens application (Unity), and Web Frontend.

The backend comprises distinct sub-components: Map Creation and Localisation (Azure Spatial Anchors), Data Storage (CosmosDB), and Backend API connection (FastAPI). Map Creation and Localisation involve communication with the Azure Spatial Anchors API, managing spatial anchor data in the cloud during mapping and relocalisation (covered in ASA subsection 3.1). Azure Cosmos DB, Microsoft's database solution, stores post-it data (content, anchor assignment, group info, etc.), detailed in subsection 3.2. For standardised client communication, the authors developed a public API using Fast API, enabling CRUD (Create, Read, Update, Delete) operations on the database content (explained in subsection 3.3).

The HoloLens application, developed using Unity [11] and the Microsoft Mixed Reality Toolkit (MRTK) [6] allows the user to place and visualise post-its in the MR space. The application communicates with the ASA backend via Microsoft's .NET Spatial Anchors package for mapping and localisation of spatial anchors. Once an anchor is located and identified by ASA, the SpatiaList API collects further necessary information. Details about the app's functionalities are provided in subsection 4.1.

The frontend responsive web dashboard, built with Tailwind CSS [10], allows users to view MR space post-its through a web browser on their smartphone. Additionally, it features a gesture-based function enabling users to swipe new post-its into MR space, bypassing the need for typing using the MR keyboard. More implementation details can be found in subsection 3.4.

3.1. Azure Spatial Anchors

Azure Spatial Anchors (ASA) provides a spatial persistence framework, with most of the relocation work outsourced to the Microsoft Azure platform. By using the vast array of sensors present on MR devices (in our case HoloLens), ASA constructs a 3-dimensional point cloud of the ambient space as the user moves around and stores it on the Azure cloud. After enough data points have been collected, users can then place Spatial Anchors in this mapped space. It is important to note that ASA uses the underlying OpenXR framework for spatial understanding [2]. In the HoloLens application, Spatial Anchors are represented using a simple 3D model of an anchor, figure 2 depicts a visualisation of a spatial anchor. A Spatial Anchor can best be understood as a local coordinate frame system fixed at a location in space that other objects (e.g. post-its) can be attached to. The design idea behind ASA was that every object (post-it) should have its own Spatial Anchor. Yet, this is a costly assump-

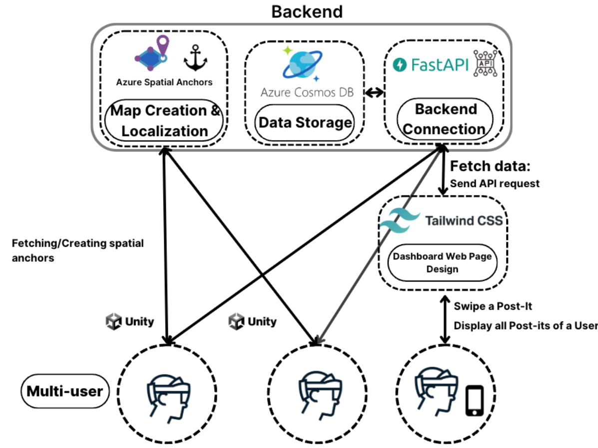

Figure 1. Overall System Architecture: Unity app (bottom left), backend consisting of Azure Spatial Anchors, Azure Cosmos DB and FastAPI (top) and frontend (middle right). Multiple HoloLens devices can query the system simultaneously.

tion as relocating anchors becomes rapidly more resource-intensive as their numbers grow. Based on the assumptions that post-its are usually placed quite close to each other (ex. around a desk or on the same wall), the authors opted to design a system where multiple objects can be attached to the same anchor, provided that the anchor is close enough.

To manage Spatial Anchors, the user first creates a new group/room or enters an existing room as visualised in the 6. Anchors can be placed in the MR space by entering the mapping mode. A short air tap gesture creates an anchor in the position of the hand tap as depicted in 2. To ensure the successful extraction of features around the anchor, the user may move around the created anchor. Upon success, the anchor is saved in the ASA cloud and returns a new unique anchor_id. This anchor_id and the corresponding group name are saved to CosmosDB through the API endpoints. Whenever the user enters a room, the application runs a background task ("Localisation mode") in which all the existing anchor_ids from the entered the group are fetched from the API. These anchor_ids are added to a "Watcher", a method provided by the ASA Unity library that fires an asynchronous callback when an anchor has been located in the local MR space. ASA sets an arbitrary limit of 30 anchors per watcher, which has caused us quite some trouble and further justifies the design choice of attaching multiple objects to the same Spatial Anchor. However, it seems one can create multiple watchers, but documentation regarding how multiple watchers are managed is lacking.

It is left up to the user to place Spatial Anchors efficiently: indeed, the user should place them to maximise coverage of the local space by distributing them uniformly with a spacing of around two meters between anchors. Another smart placement choice is to place a higher distribution of anchors around key points (like a desk) where objects are more likely to be placed. It is important to note that the Spatial Anchor placement step could have been entirely automated with the following simple algorithm: when placing an object, if possible attach to the closest anchor less than three meters away, else create a new anchor at the object position. However, the authors found it academically interesting to explore anchor placement and visualise the persistence of anchors.

Once the placement of anchors is complete, the user can create post-its using the same air tap gesture by entering Create Mode. A new post-it is considered a draft: the app waits until the user is done editing the post-it before the information is sent to the database through the API. To properly persist the object, the spatial transform of the object in terms of position, rotation and scale relative to a local Spatial Anchor has to be found. As multiple anchors may be present in the space and the error of localisation increases with the distance between the anchor and post-it, the following simple decision rule is used:

Let A be the set of all anchors, where each anchor $a_i \in A$ has a position $\vec{p}_i$ in 3D space. Given a position $\vec{v}$ in 3D space, the function `GetClosestAnchor` can be defined as:

$$\texttt{GetClosestAnchor}(\vec{v}) = \begin{cases} (\text{null}, \infty) & \text{if } |A| = 0 \\ \operatorname*{argmin}_{a_i \in A} \|\vec{v} - \vec{p}_i\| & \text{otherwise} \end{cases}$$

Figure 3 shows an example of how post-its are attached to the nearest anchors in the space. Once the post-its are attached to the nearest anchor, the post-it content and reference anchor_id is saved to CosmosDB through the API.

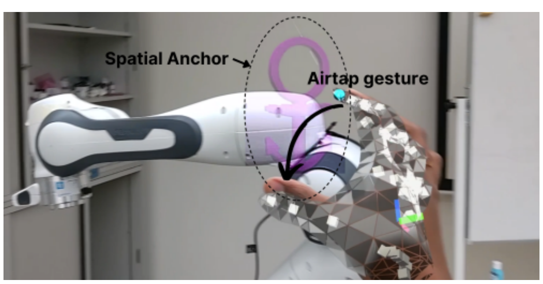

Figure 2. Spatial Anchor visualisation: Representation of the spatial anchor in 3D space as seen by the user after creation (AirTap gesture). Future created post-its are attached to this spatial anchor to ensure spatial consistency.

3.2. Azure Cosmos DB

Azure Cosmos DB [3] is a NoSQL database offered by Microsoft through the Azure platform. Cosmos DB provides low latency R/W operations (the read and write latency is less than 10 milliseconds at the 99th percentile [4]),

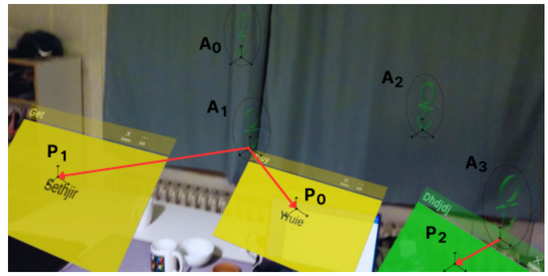

Figure 3. Attaching post-its to anchors: The post-its are attached to the nearest anchor, red arrows denote the positional vector transform between the anchor and persisted object. P_0 and P_1 is attached to A_1, P_2 is attached to A_3

which is critical for multi-user interactive applications. The database consists of separate containers used to store users, groups, anchors and post-its as structured data. The group container is used to manage the relation between anchor_ids in the space and the users who have access to them by joining the group. This allows multiple groups to exist in the same physical space and provides separation of the created post-it notes. The post-its are modelled as a unique postit_id, its attached anchor_id and the group_id. Naturally, the post-it model also contains an encoding of the relative transform from the anchor to its local position, as well as its content data (title, text, colour and content type). When a Spatial Anchor is located and its callback executed, the client will query all post-its attached to the anchor (using its identifier). Therefore, the post-it container is optimised by partitioning data on the anchor_id property, yielding lower loading times. Another optimisation is the implementation of an API endpoint which can return a hash of the state of the post-it container, which changes only when previously fetched data has become stale (ex. another user uploads a post-it). This means that rather than going through all post-its to check for modifications, the client can simply wait until there is data to pull. Further work could include implementing WebSockets-based communication between the components of the system.

Another container is also used as a buffer between the Web interface and the client to save draft post-its generated from the gesture-based interface until they are fetched by the user client.

3.3. Backend Connection: FastAPI

FastAPI is a modern, fast (high-performance) web framework for building APIs with Python 3.7+ based on standard Python type hints, significantly easing the development and maintenance of robust and scalable web applications. The API endpoints only exist to provide essential CRUD (Create, Read, Update, Delete) functionalities for post-its, anchors, users, and groups. Since most of the computing work is done through ASA and CosmosDB, the essential functionalities of FastAPI where more than sufficient to provide a standardised interface. Postman [9] is used to prototype and test the endpoints in a sandbox environment and for documenting the API.

3.4. Mobile-friendly Web-interface

A mobile web interface was developed with Tailwind CSS [10] to enhance user experience and provide responsive content for any device (accessible at SpatiaList). Compared to traditional post-its that only exist in the physical space, the SpatiaList post-its can seamlessly transfer between the MR space and the digital space. Our web interface facilitates this interconnection. Users can access a dashboard where they can use a swiping gesture to create new post-its from their smartphone (see 3.4.1) and view their saved post-its (see 3.4.2). This serves to show the possibility of inter-device compatibility in the MR space.

3.4.1 Swipe: Smartphone to Mixed-Reality

User experience with the Hololens2 identified typing on the holographic keyboard as a key inconvenience (many citing slow speed and typos). The swipe web page (see figure 4) provides the functionality to 'swipe' post-its from a user's smartphone to the mixed reality environment on the Hololens. Users can enter their username, post-it title and text directly on their smartphone, after which they can 'swipe' into MR space by sliding the yellow box from bottom to top. The swiped post-it is rapidly loaded and visualised in front of the user in the MR space. Figure 9 (a) shows a post-it that has many lines of text which was swiped from the smartphone.
Rich content can also be selected and swiped using the web interface. The post-it is then rendered by loading the provided URL to a HTML file or web page by switching the content type to "media". To enable this, the experimental MRTK WebView2 package was used [5] which allows us to render web content directly in MR. Figure 9 (b) shows a use case where a rich content-enabled post-it is used to show a leaderboard.

3.4.2 List: Post-it Filtering and Visualization

The mobile web interface provides the functionality to view the post-its created thus taking them from the MR space to the digital space. This provides the functionality to copy the content of the post-it notes and transfer into messages, email, etc from smartphone. Figure 5 shows the user interface. Users can filter the post-its based on the group or the username. Once the "Show Post-Its" button is pressed, the filtered post-its are displayed in a scrollable section. The respective title, content, and colour are used to visualise the post-its.

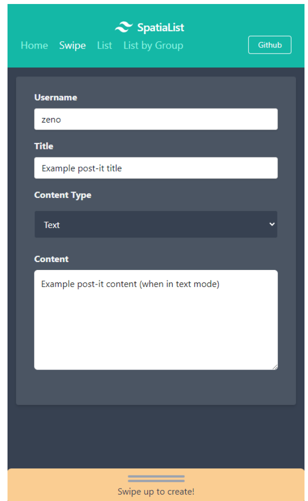

Figure 4. Mobile web interface for the interactive swipe feature: user enters username, title, content type (text/media), content (text/url). User can swipe post-it and visualise it in MR space 1m away oriented towards the user.

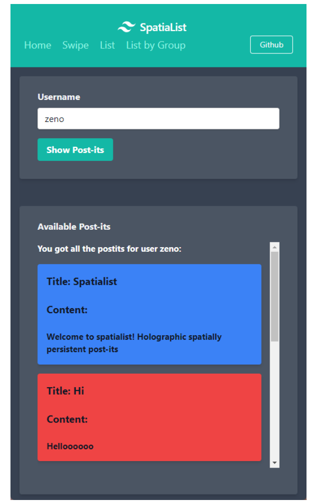

Figure 5. Mobile web interface for the post-it visualisation: Users can filter the post-its based on the group or the username. The respective title, content, and colour are used in the visualisation.

4. Results

The project yielded significant positive results in two primary domains: the application's functionalities and its usability. Firstly, a review highlighting the application's main functionalities and user experience is detailed. Then, results of the usability study performed on 16 sample users is analysed. The following subsections (4.1 and 4.2) delve into the outcomes derived from the implementation of SpatiaList.

4.1. Application functionality

4.1.1 Spatial Persistence and Performance

As part of the usability study (see section 4.2 for the full analysis), 16 participants were questioned and analysed while going through the user experience of the SpatiaList application. Both the authors and the study participants reported spatially persistent post-its, and hence spatial anchors, with little drift in the MR environment. No noticeable additional positioning error or drift was observed from attaching multiple objects to the same anchor.

The multi-user functionalities (CRUD features across all devices) performed well (latency under 5 seconds) in spaces with a stable internet connection. Naturally, an unstable internet connection severely hinders usability of the application through increased communication time between users.

4.1.2 User Experience

Once the user enters the app, the user is prompted to enter an existing group or to create a new group as depicted in Figure 6 (b). The concept of groups allows the users to have viewing control of the post-its placed and provides a means of filtering the post-its, as multiple groups can exist in the same physical space. To create a new post-it, the user needs to raise either the left or right-hand palm facing up, showing the hand menu (see Figure 6). By selecting the Create Post-it button, the user can place a post-it in the MR space by performing an airtap gesture. This will create a post-it in the space at the location of the airtap facing towards the user. The default view of the post-it is shown in Figure 7. Users can edit the text and title by clicking on the edit text/title buttons, change the colour of post-its using the colour buttons on top, and rotate and scale the post-it as shown in Figure 8. In addition, the user can delete, save, and toggle the edit mode of the post-it, putting in a modifiable (draft) state.

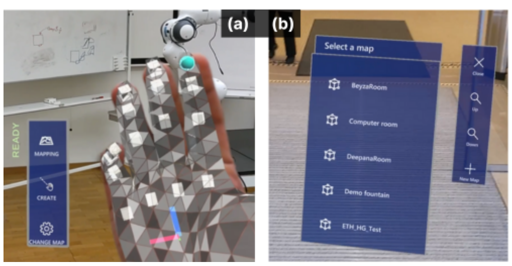

Figure 6. (a) hand menu pops up when the user raises eiher hand, user can either start mapping, placing or change group. Status text indicates the user's current selected mode. (b) The group selection UI enables users to scroll through available groups,or to create a new group.

Figure 7. Post-it User Interface: User can perform the CRUD and customization (color, scale, text) operations.

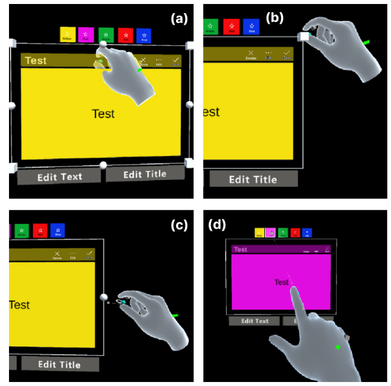

Figure 8. Interacting with Post-it: (a) user can grab the post-it from the title bar and move it in the MR space, (b) user can grab from the corner and scale, and rotate the post-it, (c) user can grab the ball-shaped handle to tilt and pan the post-it, (d) user can change the colour using the pressable colour buttons.

The spatial persistence of post-its provides users with a contextual understanding of information, which is particularly useful in spatially oriented tasks like design, planning, or navigation. Figure 9 shows two example usages of the SpatiaList application.

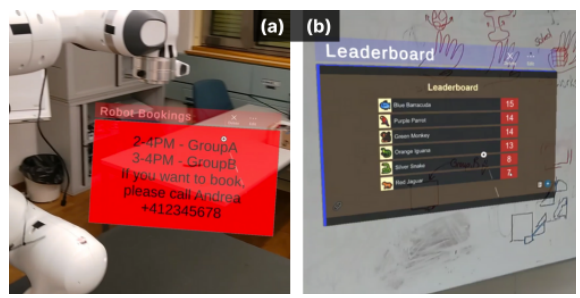

Figure 9. Example application scenarios for SpatiaList: (a) a post-it is used to book a timeslot for the robotic arm at the Soft Robotics Lab ETHZ, (b) a post-it is used to share leaderboard information (rich content) in a multi-user game

4.2. Usability Study Findings

This section presents insights from SpatiaList's usability study conducted across various settings with 16 participants. The study was conducted over multiple days, in varied environments and classes around the ETHZ campus. It aims to gauge user interaction, satisfaction, and overall experience within the mixed-reality environment. Subsequent subsections detail methodologies, observations, and key findings, providing valuable insights into SpatiaList's practical application and user reception.

4.2.1 Research Method

The authors initiated the study by assigning participants with a specific task within SpatiaList, focusing on the fundamental action of creating a post-it note and modifying its colour. The objective was to observe and measure the time participants took to complete this basic task, providing insights into the application's ease of use and navigability. On average, task completion times hovered around 34 seconds, reflecting adequate efficiency and intuitiveness of performing basic functions within the application.

After being given time to explore the application, participants filled out a survey covering demographics and detailed feedback on their SpatiaList experience. They evaluated ease of use, commented on UI design, and shared subjective insights, providing diverse perspectives. This structured approach combined quantitative task completion data with qualitative feedback for a comprehensive usability and user experience evaluation.

4.2.2 Key Findings

Even though approximately 44% of our participants were not very familiar with MR technologies, 75% of participants found creating/editing post-it features relatively easy (see Figure 10). This gives the rough estimate that around half of novice MR users feel confident using the application, which is satisfactory but leaves room for improvement.

Figure 10. Survey results on ease of use: majority (75%) of the users found interacting with post-its easy despite 44% of participants not very familiar with MR technologies

Based on our survey results, a significant majority of participants, 62.6%, found it straightforward to perform specified tasks within the application. Moreover, a substantial 75% of users reported quick adaptation to the app's interface and features, showcasing its ease of use, even for newcomers (see Figure 11). This supports the goal of providing intuitive functionalities and rapid familiarisation for users.

The survey revealed mixed feedback on the multi-user support feature. While 37.5% found it positively impacted collaboration, 62.5% indicated that while the feature did offer some level of collaboration. This suggests that while

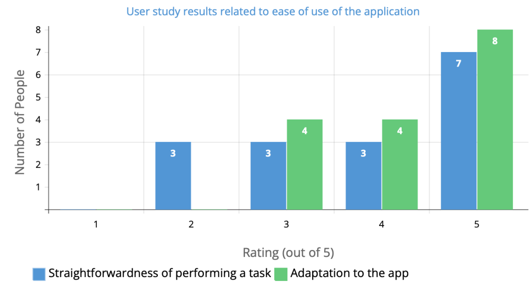

Figure 11. Survey result on adaptation to the app: 62.6% users found it straightforward to perform the specified tasks while 75% of users reported quick adaptation to the app's interface and features.

the multi-user support feature was beneficial to some extent, there are opportunities to enhance its functionality or user experience to better meet the collaborative needs of the users.

The survey results emphasise a strong preference, with 87.5% favouring the use of a web interface on a smartphone due to familiarity and quicker typing. This preference highlights the importance of user-friendly input methods. However, an interesting contrast emerged. 62.5% expressed concerns about potential reduction of immersion compared to direct interaction with the Hololens (see Figures 12 and 12). This dual-interface challenge does impact the goal of providing a seamless experience while interacting with the application.

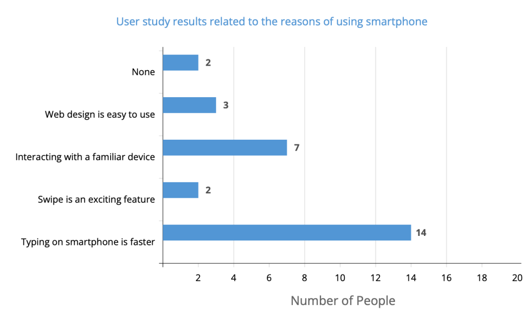

Figure 12. Survey results on web-interface: 62.5% expressed concerns about the potential reduction of immersive experience

Lastly, the authors queried participants regarding their most difficult experiences while using the application. 45% of them gave the same answer: creating the post-it. Even though the post-its are created by short tap movement with the hand, people found that part difficult due to the tap not being recognised by the device correctly.

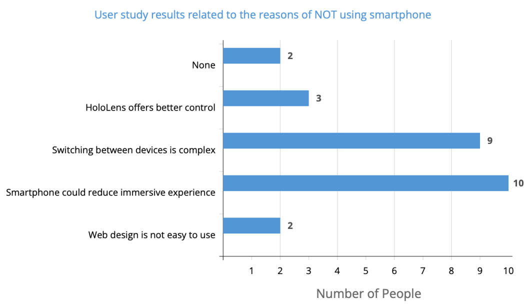

Figure 13. Survey results on web-interface: 87.5% favour the use of a web interface on a smartphone

4.2.3 Suggestions

The following suggestions were mentioned by users during the study:

- Enhance the onboarding process with more accessible tutorials and additional guidance. Given that a significant proportion (44%) of participants weren't very familiar with Mixed Reality (MR) technologies, additional guidance might aid users in acclimating to the MR environment more quickly.
- Provide additional visual cues or a brief interactive tutorial to create post-its. Since a substantial number (45%) found creating post-its (AirTap gesture) challenging, a dedicated tutorial would make this process more clear.
- Enhance the multi-user support of the application. 62.5% of survey participants identified room for improvement in this area. A specific suggestion is to enable multiple users to work on the same post-it simultaneously.

5. Discussion and Future Work

5.1. Discussion

This project has served to show both the capabilities of ASA as well as its current drawbacks.
During development, the following limitations to ASA were witnessed. First, some undocumented interactions between ASA and OpenXR led to unstable operations, causing memory errors when deleting game objects with a Spatial Anchor attached while the session is running. Furthermore, watchers are immutable and have a limit on anchors they can manage. This can lead to complexity and inefficiency in handling fluctuating anchor lists for large-scale applications. Finally, the latency and computation induced by transacting with the ASA API imply that it is mostly unrealistic to create many anchors (ie. one for each object). A more efficient approach would be to adopt a system like this project, registering anchors only when necessary using a conservative decision algorithm.
One final discussion to be had is the efficiency of the system compared to the benefits received by the user. Although this project is a proof of concept, widespread adoption would require a significant benefit gained by the user. Indeed, our usability study highlights that our application needs to provide significantly more benefits than traditional pen and paper post-its to justify both the investment costs for a MR device as well as the added complexity of wearing and interacting with a MR device. Thus, the following further improvements are proposed to maximise the user value proposition.

5.2. Future Work

Based on the feedback from users and experience during development, the following areas of improvement are proposed:

- Enabling exporting of post-its to widely used productivity tools such as Microsoft OneNote or Google Workspace.
- Integrating speech-to-text technology to create, edit, or manipulate post-its.
- Implementing more robust security measures for the system. Strengthening security protocols, implementing encryption standards, and incorporating authentication measures.
- Allow multiple users to see modifications while a user is editing post-its.

6. Conclusion

SpatiaList showcases the potential of MR technology to enable multi-user collaboration and information sharing, in both the digital and physical space. By integrating Azure Spatial Anchors, Cosmos DB and FastAPI in the system architecture, consistency across sessions, space and users has been successfully implemented. A user-friendly GUI was designed (for MR devices) and on the Web (for non-MR devices) to enhance user experience. A user study with 16 participants was performed to evaluate the app's performance. Participants reported accurate spatial and session persistency. In stable internet conditions, multi-user teamwork scenarios were properly supported. The user study shows an overall positive sentiment, with key recommendations to provide more visual cues for the post-it creation step and to enable users to work on the same post-it simultaneously. Generally, more cues and instructions should be given on how to interact in the mixed-reality environment. Based on user feedback, developing post-it export services to productivity tools (e.g. Microsoft OneNote) and integrating speech-to-text technology are identified as areas of future work.

References

[1] Hung-Jui Guo and Balakrishnan Prabhakaran. Hololens 2 technical evaluation as mixed reality guide, 2022. 1

[2] Krhonos Group. Openxr overview. https://www.khronos.org/openxr/. 2

[3] Microsoft. Azure cosmos db documentation. https://learn.microsoft.com/en-us/azure/cosmos-db/. 3

[4] Microsoft. Consistency levels in azure cosmos db. https://learn.microsoft.com/en-us/azure/cosmos-db/consistency-levels. 3

[5] Microsoft. Get started with webview2 in hololens 2 unity apps (preview). https://learn.microsoft.com/en-us/microsoft-edge/webview2/get-started/hololens2. 4

[6] Microsoft. Mixed reality toolkit documentation. https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/?view=mrtkunity-2022-05. 2

[7] Microsoft Azure Spatial Anchors Documentation. Anchor relationships and wayfinding. https://learn.microsoft.com/en-us/azure/spatial-anchors. 1

[8] T. Papadopoulos, K. Evangelidis, T. H. Kaskalis, G. Evangelidis, and S. Sylaiou. Interactions in augmented and mixed reality: An overview. *Applied Sciences*, 11(18):8752, 2021. 1

[9] Postman. Postman. https://www.postman.com/. 4

[10] Tailwind CSS. Tailwind css: A utility-first css framework for rapid ui development. https://tailwindcss.com. 2, 4

[11] Unity. Unity. https://unity.com. 2