Software Engineering Department
ORT Braude College

Capstone Project Phase B

EnlightenedCircls

Or Meira Balmas

Ofir Galai

Supervisor: Dr. Anat Dahan

24-1-D-22

1. General description

1.1 The problem that we want to solve:

Families across the country are managing the demanding task of caring for children who require constant attention and supervision. This involves ensuring a suitable environment for the child, engaging in leisure activities, and maintaining the household's daily routine. Parents face significant challenges in balancing the needs of their child with their own well-being and the needs of other family members. Often, they need support from professionals and others who have experienced similar situations. The demands of caregiving can make it difficult for parents to find time for themselves, highlighting the need for additional support and resources.

1.2 Our system:

Our application serves as a social platform for families with children with special needs. The purpose of the application is to create social communities in order to alleviate the families' loneliness and sense of helplessness and even encourage mutual help**.**

Our system operates in a sequential structure where the user first logs in with his active account, and if none exists, a new one is created.

After logging in, the user reaches the main menu, which displays all the primary activities available in our application. These activities represent the needs we aim to address and support for our users. The system's objectives are visible in the main menu, which includes actions such as:

1. Chatting with an expert for assistance when needed

2. Chatting with other users to exchange tips and advice

3. Searching for social events for families

4. Viewing frequently asked questions and answers

5. Joining a community based on location and similar criteria, enabling users to connect with others on shared topics and experiences

6. Receiving and providing quick assistance based on location

Clicking on any of these activities leads to dedicated screens that provide more information and allow further interaction within the app.

1.3 Our users:

Our target audience for our application consists families with children who have special needs and require assistance in areas where our app provides solutions, such as counseling and support.

This audience includes:

1. Siblings: Brothers or sisters who want to take their sibling with special needs to social events or gatherings with other children who have similar experiences, providing a comfortable environment for interaction.

2. Parents: Parents who are facing challenges and need advice from experts or other family members who have gone through similar experiences and can offer tips and support.

3. Other Family Members: Relatives who are involved in caregiving and seek guidance or support in managing the unique challenges they face.

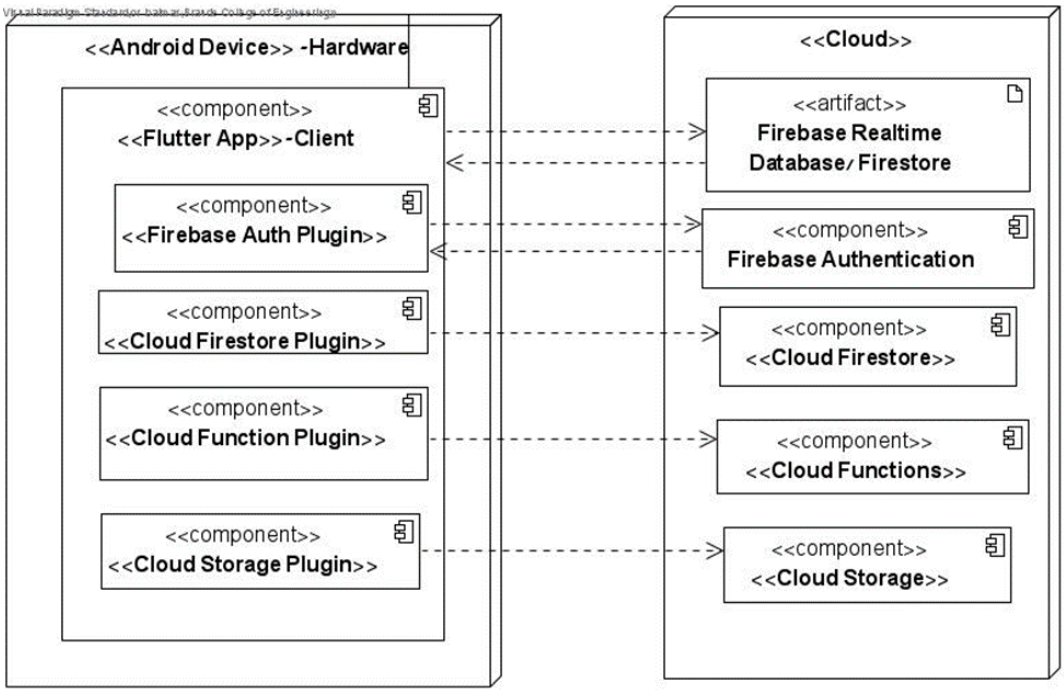
Our application offers various resources and activities designed to support these families, helping them connect with others in similar situations and providing access to expert advice and community support. This approach aims to create a supportive network that enhances the quality of life for both the children with special needs and their families.

In addition to providing a platform for interaction and support, the application also facilitates participation in social events and community activities, which are crucial for the social integration and well-being of children with special needs and their families.

2. Solution description:

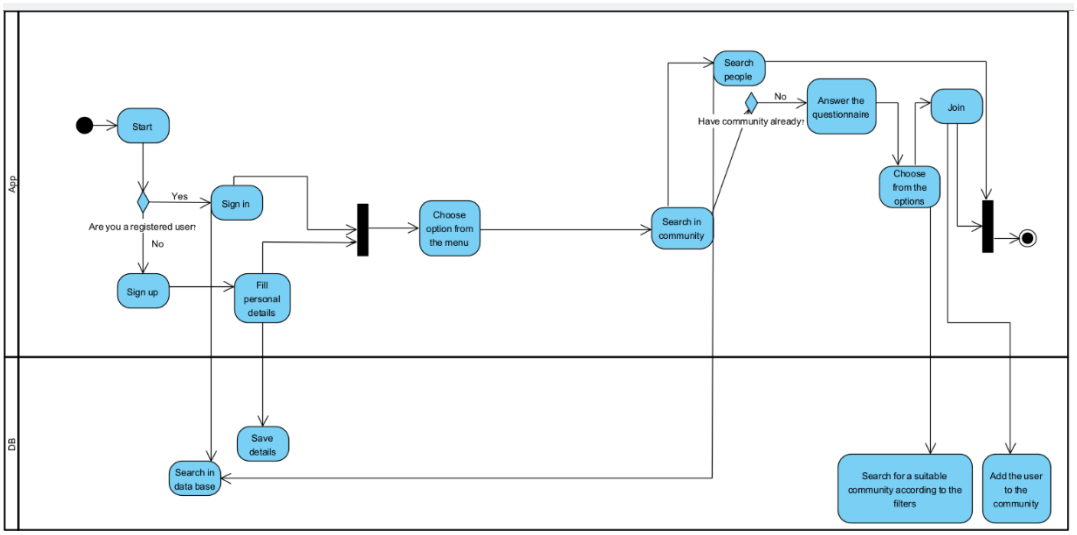
Our solution includes an application that we developed using Android Studio and VS CODE using the DART programming language, as well as the database we chose Fire Base -Fire Store .

Figure 1: System architecture



Activity: This Activity diagram shows the flow of the actions when the user selects the community search option in the menu. The actions in the application and in the data base.

Figure 2: Activity diagram



In the In initial phase of building the system, we conducted online research to understand the existing algorithms. Following this, we began working on the screen sketches we wanted to include in the system. We developed a client-server infrastructure and a basic database. Subsequently, we started developing all the main functions according to screens and the steps within each screen in the main menu. At each stage, we consulted with potential users to gather feedback and make adjustments to improve user experience, such as changing screen colors, font size, and more.

For screen design, we used Figma.

The algorithm we implemented :

In our project, we implemented the **Louvain algorithm** to detect communities based on a user graph, where each connection between users is weighted according to various criteria. We created a graph where each user is represented as a node, and the connections between users are represented by weights calculated based on geographical proximity, shared hobbies, age, pets, and disabilities. Users who share more of these characteristics are given stronger connections in the graph.

By dynamically adjusting the weights of these connections, we were able to account for more than just physical proximity, incorporating personal attributes as well. After building the graph, we ran the Louvain algorithm to efficiently identify communities by grouping users who are closely related according to the specified factors.

We successfully achieved smart and automatic grouping of users into communities, where the algorithm considered a combination of geographical closeness and shared characteristics. This approach allows for a more personalized community formation, enhancing the accuracy of user matching based on shared traits and interests. However, due to time constraints and the complexity of the implementation, we were unable to allow users to join more than one community or to remove form the community they choose.
We were not able to use the dynamic auxiliary operations we talked about in part A due to time constraints and the complexity of the implementation .

3. Our challenges:

During development, we faced several hardware and engineering challenges:

1. **Hardware Issues**: Using Android Studio in combination with VS Code requires a lot of system resources, including RAM and processing, which can cause computers with less powerful hardware to run slowly .Our personal computers were not powerful enough and crashed during executions. To overcome this, we used a computer provided by one of the team members from her workplace, which had the necessary capabilities to handle the workload.

2. **Engineering Challenges**: We encountered difficulties with our algorithm for community development. Implementing it proved challenging, so we decided to simplify the algorithm by reducing its complexity. Initially, we had planned for each user to know the name of the community to which they were assigned, but we decided to limit this feature. Additionally, we restricted users to joining only one community at a time. These changes were made to ensure the system's functionality within the limited time frame we had. While these adjustments were necessary, we plan to improve and optimize the algorithm in the future to enhance its capabilities and user experience.

3. **User Experience:** Designing an intuitive and user-friendly interface requires careful consideration and may involve multiple iterations based on user feedback. This includes adjusting visual elements like colors and font sizes to improve accessibility and usability. We design an

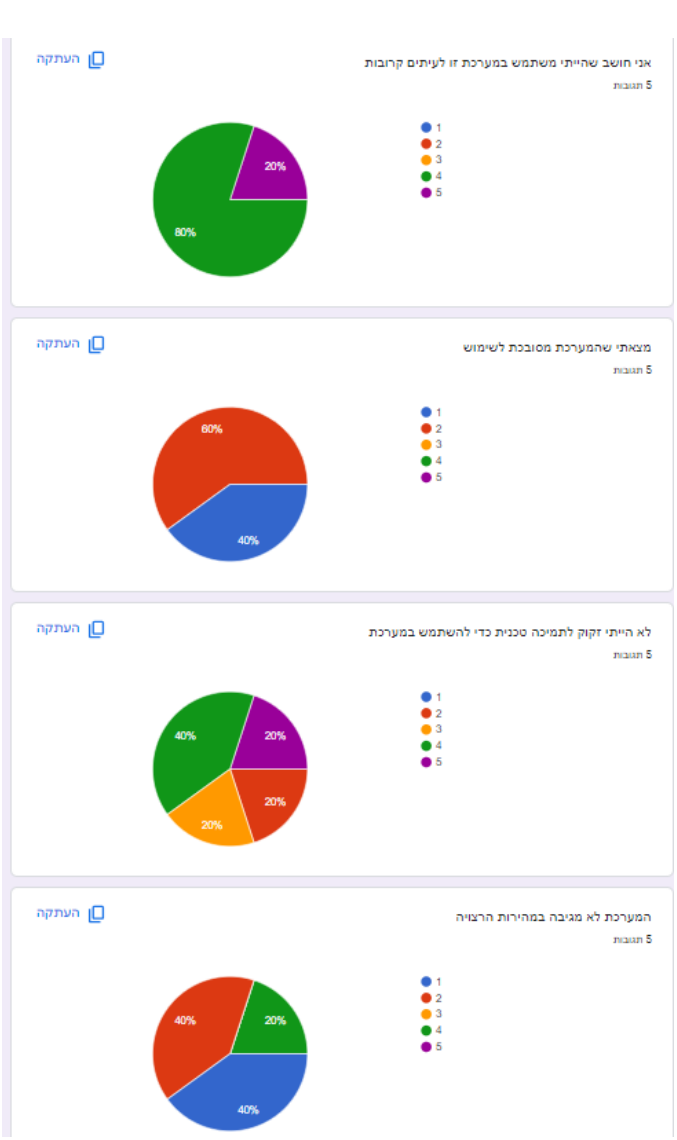
intuitive and user-friendly interface requires careful consideration and may involve multiple iterations based on user feedback. This includes adjusting visual elements like colors and font sizes to improve accessibility and usability. In addition, during the development process, we received a lot of feedback from potential users that was sometimes contradictory, for example sometimes there were users who preferred the font in the application to be a certain color and others preferred a different color. After joint consultation we decided to make decisions according to what most users responded.

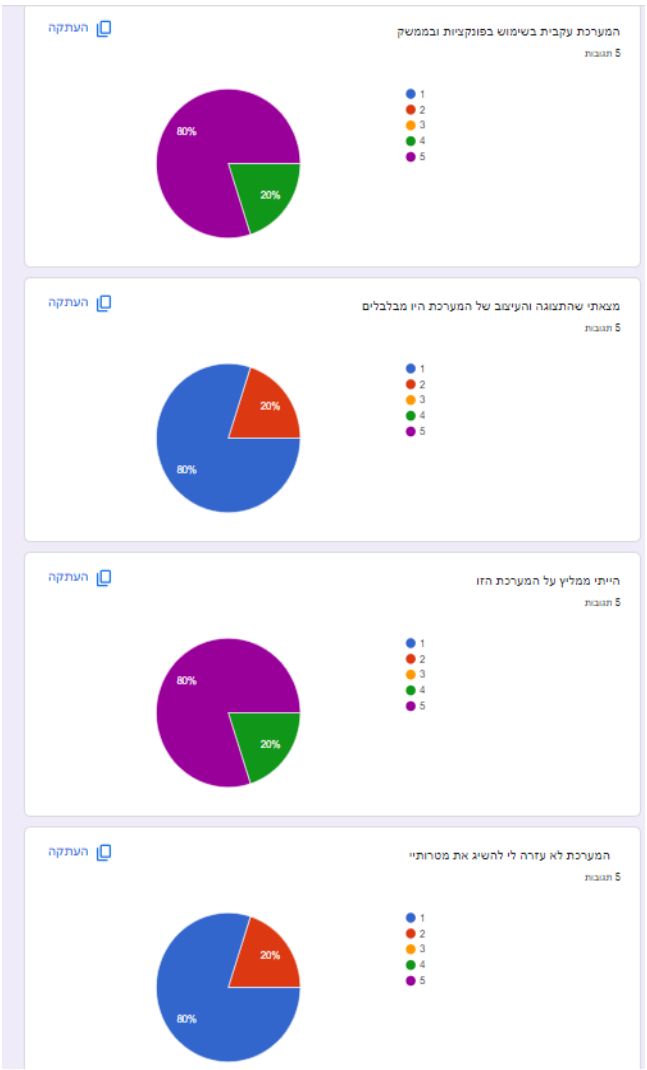
During the development process, we faced various challenges, including hardware issues and algorithmic complexity, but we managed to find creative and practical solutions. We improved the user experience by designing a friendly and accessible interface, relying on feedback from potential users. Additionally, we simplified the algorithms to ensure the system's functionality within the time and resources available to us. All of this was done to provide users with a high-quality and efficient experience that meets their needs in the best possible way.

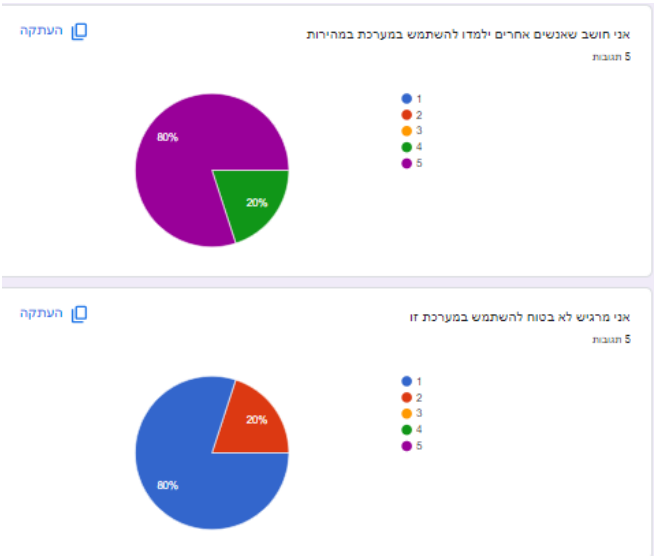
4. Project User evaluation:

In our project, we chose to utilize the System Usability Scale (SUS) questionnaire to gather feedback on our application. The SUS is a widely recognized tool for assessing the usability of a system, application, or product. It provides a quick and reliable method for evaluating the user experience, which is crucial for ensuring that our app meets the needs and expectations of its users.

The SUS feedback gave us valuable insights into how users perceived our application. For instance, if users found certain features confusing or difficult to navigate, this information guided us in making necessary adjustments to improve the user experience. Additionally, the SUS results helped validate our design choices, confirming that most aspects of the app met the desired usability standards.







The average score of the questionnaire is 87.5

5.System testing:

5.1 Unit Testing:

The tests performed on the system to evaluate its functionality. We tested edge cases to ensure that the application functions as intended.

Test Name	Test description	Was a desired result obtained?	Real Result after test
1.Input validation test at registration	1.1wrong email and wrong password 1.2 correct password and username 1.3 correct password wrong username 1.4 correct username wrong password 1.5 password is shorter than 6 characters 1.5 empty	V V V V V V V	1.1 invalid credential 1.2 sign in and show the user screen 1.3 invalid credential 1.4 invalid credential 1.5 Password must be at least 6 characters 1.6 Email and password is empty
2.Database connection test	2.1 save new users to data base- fire store 2.2 adds users to commuties	V V	2.1 saves in fire store all the info on user- saves in the collection and in authentication

			2.2 saves the user extra data in community in fire store – shows a message community joined successfully
3. Successful registration test	3.1 can add new users to the app	V	3.1User saved and showed the opening screen
4.FAQ functionality test	4.1 there are answers to FAQ	V	4.1all FAQ answers are avalible
5.Chat Interface Availability test	5.1 users can send other user messages 5.2 user can send experts messages 5.3 users can receive response from other users	V V V	5.1 user can choose which another user he wants to send a message 5.2 user can choose which expert he wants to send a message 5.3user receive response form chat- a pop up message shows up in phone
6.Questionnaire Access test	6.1 user can answer the questionnaire to get into community	V	6.1 user can answer all questions- after the user answers the most suitable communities are shown
7.Help request	7.1 user can post request for help form other users 7.2 7.2 user can offer help the other users	V V	7.1 user can upload request for help - the request is shown and a message show - successful request 7.2 user can help other - when pressing help button a chat open
8. Event test	8.1user can post a new event	V	user can post event - the event is shown on the event page

5.2 Simulation Results

To test our community-building algorithm, we added several users to the application. By completing the questionnaires, different and diverse communities were suggested to each user according to their preferred match. It can be seen that users with similar profiles were grouped into the same community and that several communities were created based on the characteristics of the users.

These are the profiles of the features of all the users we tested in the simulation- in total we tested 15 users:

Username	**Age of child**	**Geographical location**	**Other features filled in the Tell About Yourself questionnaire**	**A community to which the algorithm belongs**
Ofir galai	3	Akko	Autism Spectrum Disorder, music, cat	Do-Gooders
Or balmas	4	Kiryat Bailik	Autism Spectrum Disorder, music, sports, cat ,dog	Do-Gooders
Yonit Dagan	6	Haifa	Visual Impairment, cooking, learning ,rabbit	Social Battalion
Sapir fadida	4	Kiryat Yam	Visual Impairment, Photography ,Cat, dog	Town Union
Dani Cohen	10	Carmiel	Autism Spectrum Disorder, cooking ,reading ,fish	Positivity Squad
Ron Haim	9	Kiryat Motzkin	Autism Spectrum Disorder, cooking ,fish	Positivity Squad
Agam Levi	12	Haifa	Visual Impairment ,Reading ,dog	Town Union
Gali Avram	12	Akko	Visual Impairment, Reading ,Photography ,dog	Town Union
Gal Friedman	5	Carmiel	Gardening ,Reading, Painting, Cerebral Palsy Bird ,Horse	Social Circle
Dana Levi	11	Nahariya	Gardening ,Painting, Cerebral Palsy Horse	Social Circle

Tomer Hamiliton	7	Carmiel	Learning ,Traveling ,Physical Disability,bird	Friendship Forum
Amber Goth	9	Carmiel	Painting ,Photography ,Cerebral Palsy, dog	Social Circle
Alex Smith	5	Haifa	Gardening, Reading, Panting Cerebral Palsy ,bird	Social Circle
Hanna May	10	Akko	Visual Impairment ,cooking ,rabbit	Social Battalion
Emma Brown	4	Haifa	Autism Spectrum Disorder,sports, music	Do-Gooders

5.2 Analysis of the simulation results

The **Louvain algorithm** successfully divided the 15 users into distinct communities based on multiple criteria, including geographical location, shared hobbies, disabilities, and the age of the user's child.

1. **Community Formation**: The algorithm created several distinct communities such as *Do-Gooders*, *Town Union*, *Social Battalion*, and *Positivity Squad*. These communities are formed by clustering users who share common characteristics.
2. **Shared Features:**
    - **Hobbies and Disabilities**: Users with similar hobbies and disabilities were often placed in the same community. For example, those with visual impairments and interests like cooking or photography were grouped together in communities such as *Town Union* and *Social Battalion*.
    - **Geographical Proximity**: Users from similar locations, like *Akko*, *Carmiel*, and *Haifa*, were sometimes grouped together. However, geographical proximity was not the dominant factor in community formation.
3. **Diversity in Communities**: Some communities, such as *Social Circle*, had a higher concentration of users with specific shared traits like gardening and cerebral palsy, while others like *Do-Gooders* displayed more diversity in user characteristics, such as different disabilities or a wider range of hobbies.
4. **Focus on Commonalities**: The algorithm seemed to prioritize shared hobbies and disabilities over geographic location and child's age when determining community membership. For example, users with similar disabilities like autism spectrum disorder were grouped together even if they were from different locations.

Overall, the algorithm effectively created personalized communities based on shared interests and traits, though certain factors like geography played a lesser role compared to shared hobbies and disabilities. It is important to say that sometimes the algorithm offered users several options for a community that could suit them.

Figure 3: simulation results -Distribution of Communities

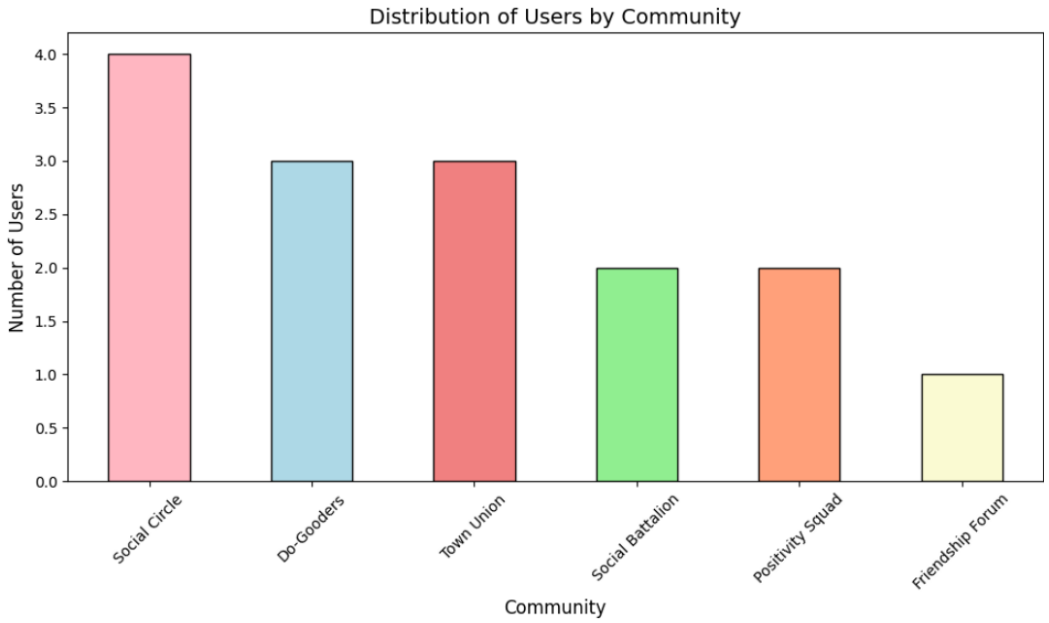


Figure 4: simulation results -Distribution of Hobbies

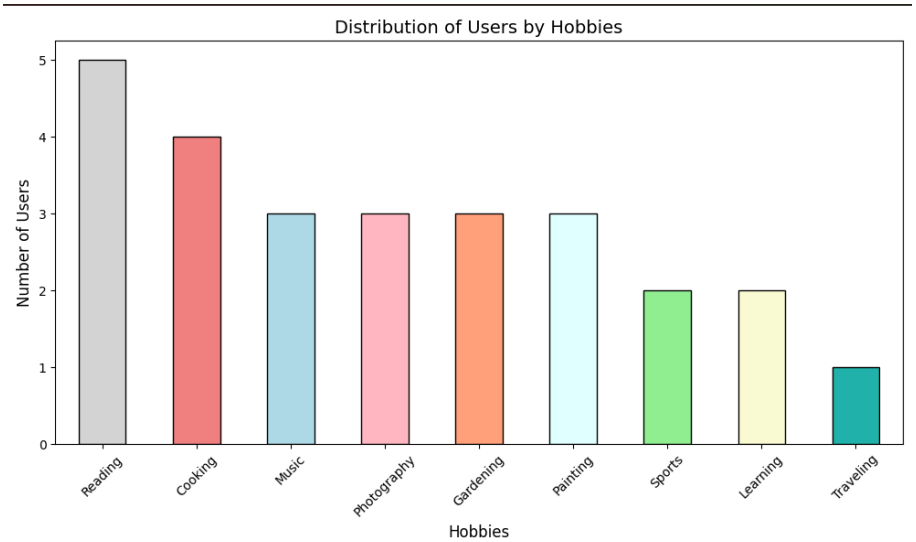
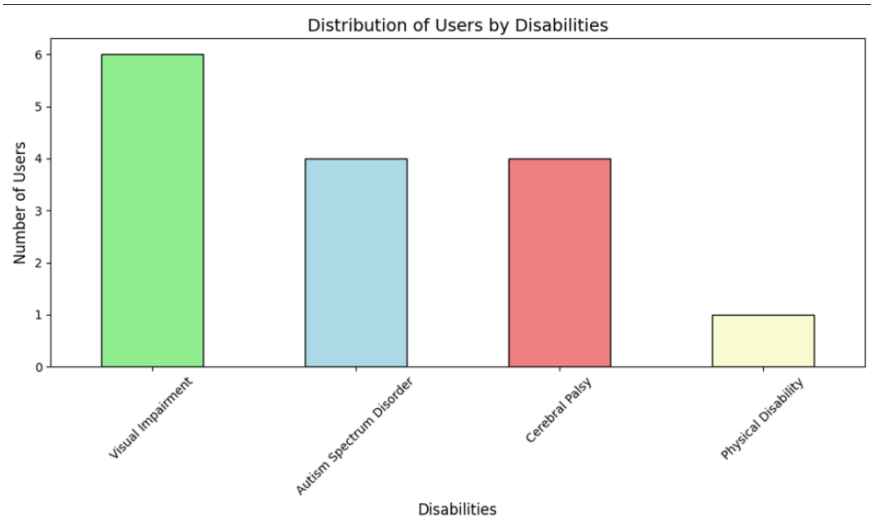


Figure 3: simulation results -Distribution of Disabilities



6. Results, Conclusions & Lessons Learned:

In summary, our efforts focused on maximizing the efficiency and usability of the application. We emphasized user feedback and iterative improvements to create a product that is both functional and user-friendly. While there is always room for further enhancements, such as improving the community-building algorithm, the current version of the application successfully addresses the primary needs of our target audience and the goals we set for ourselves when designing the application in the first part of the project. We hope that this application will assist its intended users, provide solutions, and help ease, even slightly, the management of their complex lifestyles.

During the development of our application, we gained valuable insights that will guide us in future projects. Here are some key lessons we learned:

1. **Flexibility and Adaptability**: We learned the importance of being open to changes and willing to adjust plans when necessary. This was crucial in overcoming unexpected challenges, such as hardware limitations and algorithmic complexity.

2. **User-Centric Design:** Engaging with potential users and incorporating their feedback were essential in creating a user-friendly interface. This iterative process helped us refine the design to better meet user needs.

3. **Simplification of Complex Processes:** Simplifying algorithms and processes can enhance efficiency and ensure functionality within resource constraints. This approach allowed us to deliver a functional product despite time limitations.

4. **Collaboration and Communication**: Effective communication within the team and with external stakeholders was vital for addressing issues promptly and efficiently. Encouraging open dialogue helped us identify problems early and find solutions collaboratively.

5. **Continuous Improvement:** Recognizing that there is always room for improvement, we plan to enhance certain features, such as the community-building algorithm, in future versions. This commitment to continuous improvement will help us better serve our users.

Additionally, we realized that we could have managed our time more efficiently, which might have alleviated some of the challenges we faced. These lessons underscore the importance of flexibility, user engagement, simplicity, collaboration, and continuous improvement in successful project development.