

Attendance Management System - Carmey Gil Center

Bar Shimon Malka & Ofir Shahaf

**Supervisor:
Dr. Naomi Unkelos Shpigel**

Software Engineering Department

Project code: 24-2-D-36



Table Of Contents

Abstract.....	2
1. Introduction.....	3
2. Literature Review.....	3
2.1 Usable Design for Inclusive Population.....	3
2.2 User-Centered Design – UCD.....	4
2.3 Development Process of Assistance Technology for Users with Disabilities.....	4
2.4 Testing Process of Assistance Technology for Users with Disabilities.....	5
2.5 The Impact of Assistive Technology on the Characteristics of People with Special Needs.....	5
3. Project Review.....	6
3.1 Flow chart of daily attendance marking of patient	7
3.2 Description of the main operations in the system.....	8
3.2.1 Use Case Diagram.....	8
3.2.2 Activity Diagram.....	9
4. Description of Development Process.....	11
4.1 Engineering Process.....	11
4.1.1 Planning and Initial Research.....	12
4.1.2 System Design and Architecture.....	12
4.1.3 Prototyping and Feedback.....	12
4.1.4 Adapting the System for Tablet Use.....	13
4.1.5 Rigorous Testing and Deployment Plan.....	13
4.2 User evaluation meeting - Carmey Gil Center	14
5. Description of System Deployment and Operation.....	15
5.1 System Deployment.....	15
5.1.1 Client Side (Frontend).....	15
5.1.2 Server Side (Backend).....	15
5.1.3 Database.....	15
5.2 System Operation.....	15
5.3 Core Features.....	17
5.3.1 Attendance Logging.....	17
5.3.2 Emotional State Alerts.....	17
5.3.3 Service Recipients Management.....	17
5.3.4 Staff Management.....	17
5.3.5 Attendance Editing.....	18
5.3.6 Grade Updates for Service Recipients.....	18
5.3.7 Report Generation.....	18
5.3.8 Tablet Keyboard Integration.....	18
5.3.9 Keyboard Input Support for Other Devices.....	19
5.3.10 Multilingual Support.....	19
5.3.11 Voice Feedback.....	19
5.4 Report.....	20
5.5 Alert - Negative Emotion Selected.....	22
6. User Documentation.....	23
6.1 Home Page	23
6.2 Guide Selection	24
6.3 Service Recipient Selection	24
6.4 Emotion Selection	25
6.5 Login Screen - Staff Members.....	26
6.6 Staff Members Dashboard.....	27
6.7 Manage Attendance.....	28
6.7.1 Edit Attendance.....	29
6.7.2 Delete Attendance	30
6.7.3 Update grade.....	30
6.7.4 Add Attendance.....	31
6.8 Generate Reports.....	32
6.9 Service Recipient Management.....	33
6.10 Add Service Recipient	34
6.11 Edit Service Recipient	35
6.12 Delete Service Recipient	36
6.13 Staff Management.....	36
6.14 Edit Staff Member Screen.....	37
6.15 Delete Staff Member Screen.....	38

6.16 Add Staff Member	39
7. Description of Testing process.....	40
7.1 Testing Plan.....	40
7.1.1 Login Tests.....	40
7.1.2 Attendance Registration Tests.....	40
7.1.3 Staff Actions Tests.....	41
7.1.4 Report Tests.....	41
8. Maintenance Guide.....	42
8.1 Installation.....	42
8.1.1 Prerequisites.....	42
8.2 Setup.....	42
8.2.1 Clone the Repository.....	42
8.2.2 Option 1: Running Frontend Only (Using Render Backend).....	42
8.2.3 Option 2: Running the Complete Application Locally (Frontend + Backend).....	43
8.3 Create a Database and Collection.....	44
8.4 Deployment Diagram.....	49
8.5 Database Description.....	50
8.6 Main Code Sections.....	52
9. Conclusion and summary.....	57
9.1 Challenges and Solution	57
9.1.1 Free Translation and Voice Support.....	57
9.1.2 Adjusting the System for Platform Transition	58
9.1.3 Deployment Services for the System.....	58
9.1.4 Training the staff members.....	58
10. References.....	59

git folder :

<https://github.com/barmalka1419/Attendance-management-system-Carmey-Gil-Center-.git>

Abstract

The Carmey Gil Center, a rehabilitative employment facility, supports individuals with intellectual, physical, and sensory disabilities. The center currently faces challenges in manually tracking attendance and lacks an effective system for monitoring the emotional well-being of its participants. These issues hinder operational efficiency and the personalized care provided at the center.

This project focuses on enhancing accessibility and self-efficacy for individuals with intellectual, physical, and sensory disabilities at the Carmey Gil Center. We developed an interactive attendance and emotional state tracking system tailored to the unique needs of the center's users.

Key features include simple attendance marking, multilingual support (Hebrew, English, Arabic, and Russian), and emotional state tracking through an intuitive interface. The system also provides voice feedback, repeating users' selections aloud to ensure clarity and accessibility for all users.

By replacing manual processes with a digital solution, the platform helps users and staff manage attendance more effectively and provides a practical alternative to manual attendance records.

1. Introduction

Managing the attendance process at the Carmey Gil Center was traditionally performed using paper records, making it challenging to maintain accuracy and consistency, which can lead to inefficiencies and errors. In addition This method created additional workload for the staff.

The goal of this project was to develop a digital system tailored to the needs of the center's users. The system provides a simple and accessible way for users to mark their attendance and indicate their emotional state through an emoji and pictures based interface. To ensure inclusivity for all the users, the platform supports multiple languages: Hebrew, English, Arabic, and Russian and features voice feedback for selections according to the selected language. This design not only empowers users with intellectual disabilities to interact with the system independently but also helps staff track attendance and emotional states more effectively.

This book explores the motivation behind the project, the technologies used, and the challenges addressed during development. It provides a comprehensive guide to understanding how the system was designed to improve attendance management process and support the needs of both users and staff at the Carmey Gil Center.

2. Literature Review

2.1 Usable Design for Inclusive Population

Developing user-friendly interfaces for individuals with disabilities demands a profound understanding of their unique needs and preferences. Research indicates that mobile applications featuring simple, customizable interfaces such as those allowing adjustments to font size and language selection significantly enhance user engagement and satisfaction [2]. Key design principles play a crucial role in promoting independent task execution. These include implementing large, easily identifiable buttons, providing clear, concise instructions, and creating intuitive navigation paths. Adherence to these principles substantially aids users in their interactions with the system [4]. In essence, the thoughtful design of system interfaces, guided by principles tailored for individuals with special needs, helps users overcome physical or intellectual barriers. This approach enables them to utilize the system with the same efficacy as any other user.

2.2 User-Centered Design - UCD

User-centered design (UCD) is an approach that focuses on involving users at all stages of development to ensure that the system meets their needs optimally. This approach emphasizes a deep understanding of the users, including their needs, abilities and limitations, and involves them in the product or system development process. In the UCD method, extensive research is carried out on the users, data is collected regarding their preferences and challenges, and tests are carried out with the users to check the suitability of the product for their needs. This process is found to be particularly effective when developing applications for people with cognitive disabilities, because it allows us to examine how the interfaces affect the users, to improve the user experience and make it more intuitive [2]. Technologies developed using a user-centered approach can be seen. In certain gaze-based technology, the unique needs of people with cerebral palsy are used as a main driver for the development of technologies that allow them to communicate and act more effectively [5]. The direct involvement of users with intellectual disabilities in the design process is an important thing in order to create solutions that fit their real needs [8]. The integration of users throughout the development process increases the effectiveness of the product and ensures that it will meet the needs and be adapted to the disabilities of the specific users for whom it is intended.

2.3 Development Process of Assistance Technology for Users with Disabilities

The development of applications for users with disabilities demands a comprehensive, user-centered approach. This process emphasizes the critical importance of thoroughly understanding users' specific needs, while incorporating iterative development cycles and ongoing user feedback [1]. The approach extends to adapting existing technologies for simple use by those with special needs. A study on a technological medicine system illustrates this point, where users struggled to access medical information due to inadequate adaptation, necessitating researcher intervention [3]. Creating specialized interfaces for individuals with special needs requires careful attention to accessibility. A study presenting a system teaching route-learning to people with special needs highlights key accessibility design principles. These include limiting the number of required actions, providing text alternatives for images, and ensuring clear, user-friendly navigation [5].

2.4 Testing Process of Assistance Technology for Users with Disabilities

Rigorous testing methods are essential when evaluating software for adults with intellectual disabilities. One research compares various evaluation techniques, such as heuristic evaluation, user testing, and automated readability evaluation. During heuristic evaluation, experts evaluate the software based on predefined usability principles and ensure that it meets certain accessibility standards. User testing, on the other hand, focuses on direct interaction with the target audience, where participants complete tasks using the software and allows evaluators to collect qualitative and quantitative data on usability, comprehension, and satisfaction. By combining these methods, the study provides a comprehensive understanding of the effectiveness of this software and highlights the importance of different evaluation methods for meeting the unique needs of adults with intellectual disabilities [6]. Furthermore, Incorporating accessibility considerations at the beginning of development ensures that accessibility is a fundamental aspect of design, not an added feature later on. Identifying and addressing potential issues from the start can prevent them from becoming more complicated and expensive to resolve later on. This method enables the creation of interfaces and user experiences that meet the varied needs of all users, leading to a more uniform and fair user experience. Furthermore, consistent testing with individuals who have disabilities during the development stages offers important insights, allowing for more precise customization of the product[9]. In light of these findings, our project will incorporate observations to gather data on system usage, conduct thorough testing of each component to preempt complex issues, and involve users with disabilities throughout the development process. This approach will ensure that our attendance tracking system not only meets accessibility standards but also truly enhances the experience for all users at Carmey Gil Center.

2.5 The Impact of Assistive Technology on the Characteristics of People with Special Needs

The impact of assistive technology on the independence and responsibility of individuals with special needs is significant. Research underscores its crucial role in enhancing perceived independence in daily activities [7]. Various technologies such as communication software, organizational and control devices, and physical tools such as writing aids and fine motor skill development tools empower users to perform tasks independently, boosting their self-reliance and confidence. The effectiveness of assistive technology in promoting adaptive skills is demonstrated in a study involving children with cerebral palsy. Tailored interventions can markedly improve children's ability to perform everyday activities such as writing, drawing to express themselves, and holding small objects. [2]. These technological solutions enable us to assume more responsibilities and engage more actively.

3. Project Review

The Carmey Gil Center's system is designed to make daily activities easier for individuals with disabilities and the staff who support them. It combines advanced technology with a simple and accessible interface, helping users express their emotions, track attendance, and manage administrative tasks more efficiently. The system focuses on creating a user-friendly experience that is both practical and adaptable to the needs of the center.

At the heart of the system lie three strategically designed modules that work in concert to revolutionize center operations. The emotional tracking module empowers users to independently express their daily emotional states through an intuitive emoji interface, while simultaneously providing staff with valuable psychological insights. The attendance management tool replaces cumbersome manual processes with a streamlined digital solution, enabling precise and effortless attendance tracking. The administrative module further enhances operational effectiveness by simplifying user management and report generation, allowing staff to redirect their focus from paperwork to direct care.

To ensure inclusivity for users with various abilities, the platform incorporates voice feedback, enabling users to hear their selections read aloud immediately after making them. This feature prevents errors, especially for users with cognitive challenges. Additionally, the system supports multiple languages, including Hebrew, English, Arabic, and Russian. Once a language is selected, all text and voice feedback adapt seamlessly to the chosen language, ensuring a fully localized and accessible experience for diverse user groups.

The platform's architectural design embodies a profound commitment to accessibility, scalability, and continuous improvement. By creating an interconnected system that prioritizes user-friendly interactions and comprehensive data management, the solution not only addresses current operational challenges but also establishes a flexible framework for future enhancements. This approach reflects an assistive technology for individuals with disabilities, demonstrating how thoughtfully designed digital tools can significantly improve communication, independence, and overall quality of life within rehabilitative settings.

3.1 Flow chart of daily attendance marking of patient

The attendance marking process for the Carmey Gil Center's digital platform represents a transformative approach to user engagement and operational efficiency. By integrating automated attendance tracking with an intuitive emoji-based emotional selection feature, the system creates a seamless, user-centric entry point that prioritizes independence for individuals with special needs. This innovative design not only simplifies the login process but also provides staff with valuable insights into users' daily psychological states. Through this approach, a manual administrative task becomes a meaningful interaction that supports both operational requirements and personalized support.

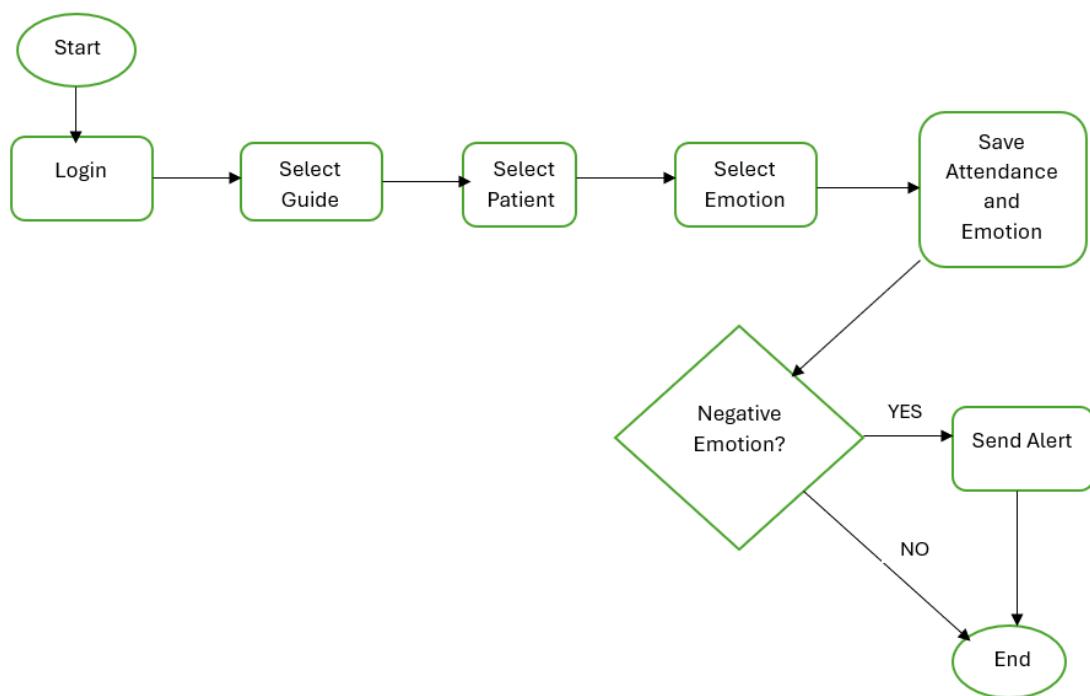


fig 1. Flow chart of daily attendance marking

3.2 Description of the main operations in the system:

3.2.1 Use Case Diagram:

Service Recipient Actions

A service recipient can perform two simple actions:

- **Attendance Registration** – This includes selecting an instructor, selecting the service recipient, and choosing an emotion.
- **Language Selection** – Choosing the preferred language for the system interface.

Staff Member Actions

A staff member can perform the following actions:

- **System Login** – Requires entering a valid username and password.
- **Report Generation** – Includes exporting reports to an Excel file.
- **Managing Service Recipients** – Allows the staff member to add, delete, edit service recipients.
- **Managing Staff Members** – Allows the staff member to add, delete, or edit staff members.
- **Attendance Management** – Allows staff to manage attendance records and update grades for a specific service recipient.

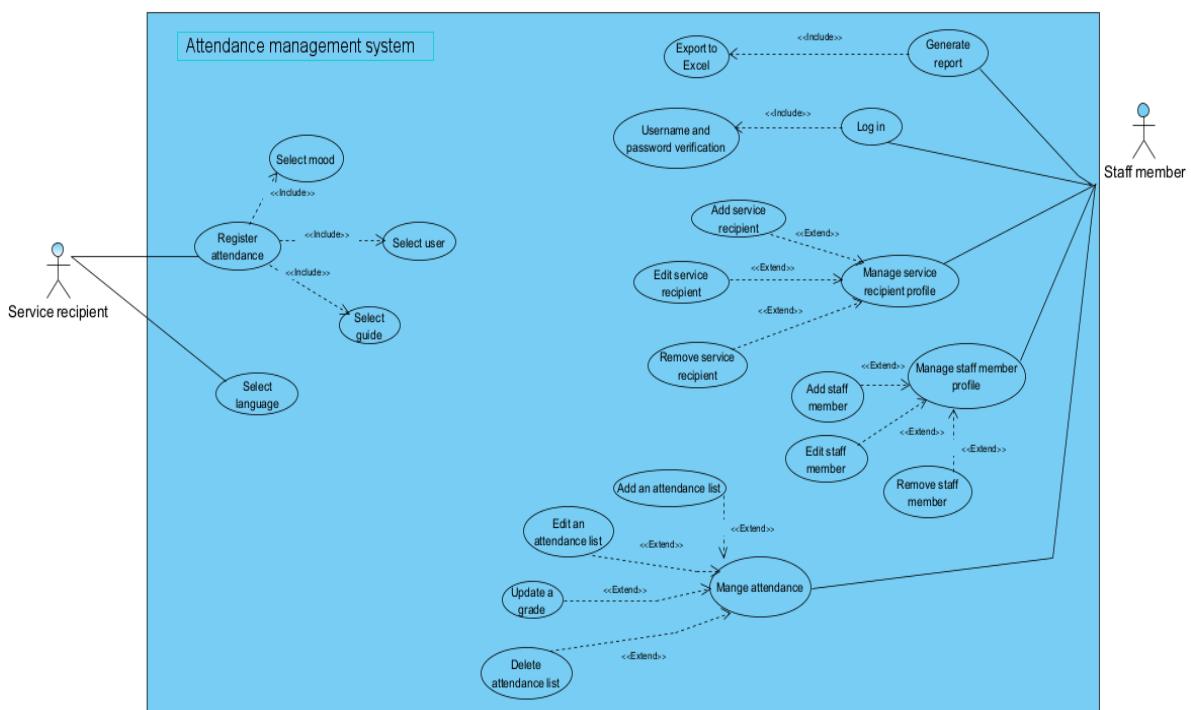


fig 2. Use Case diagram

3.2.2 Activity Diagram

This diagram maps the workflow and processes within the system, showing the sequence of activities of each user and how they communicate with each other and with the system. The first diagram is for the actions of the service recipient, and the second diagram for the actions of a staff member.

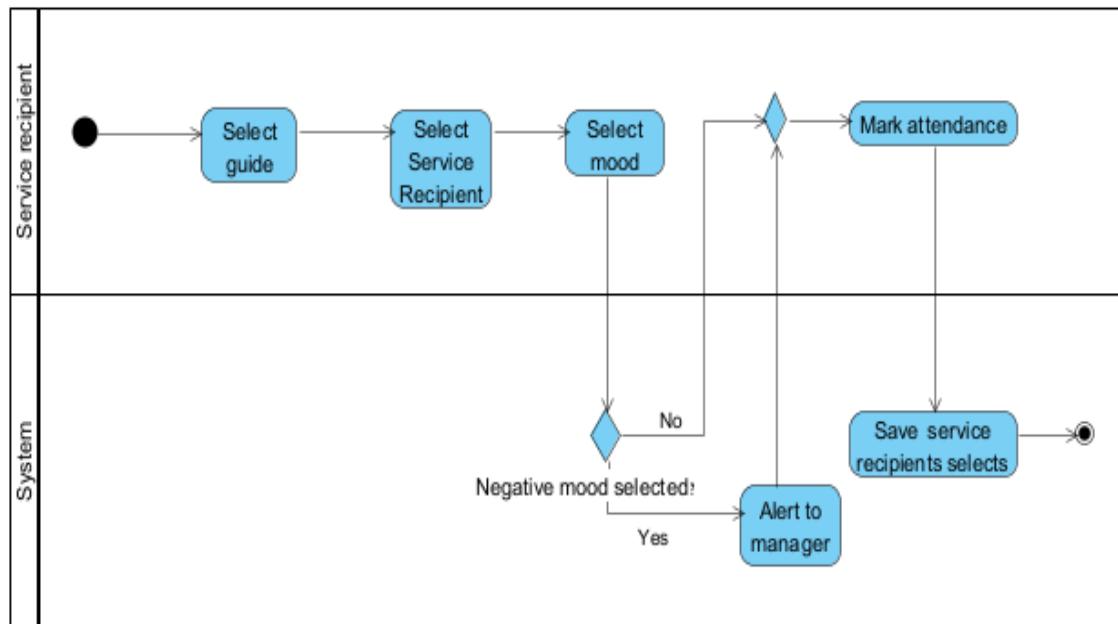


fig 3. Activity diagram - Attendance registration

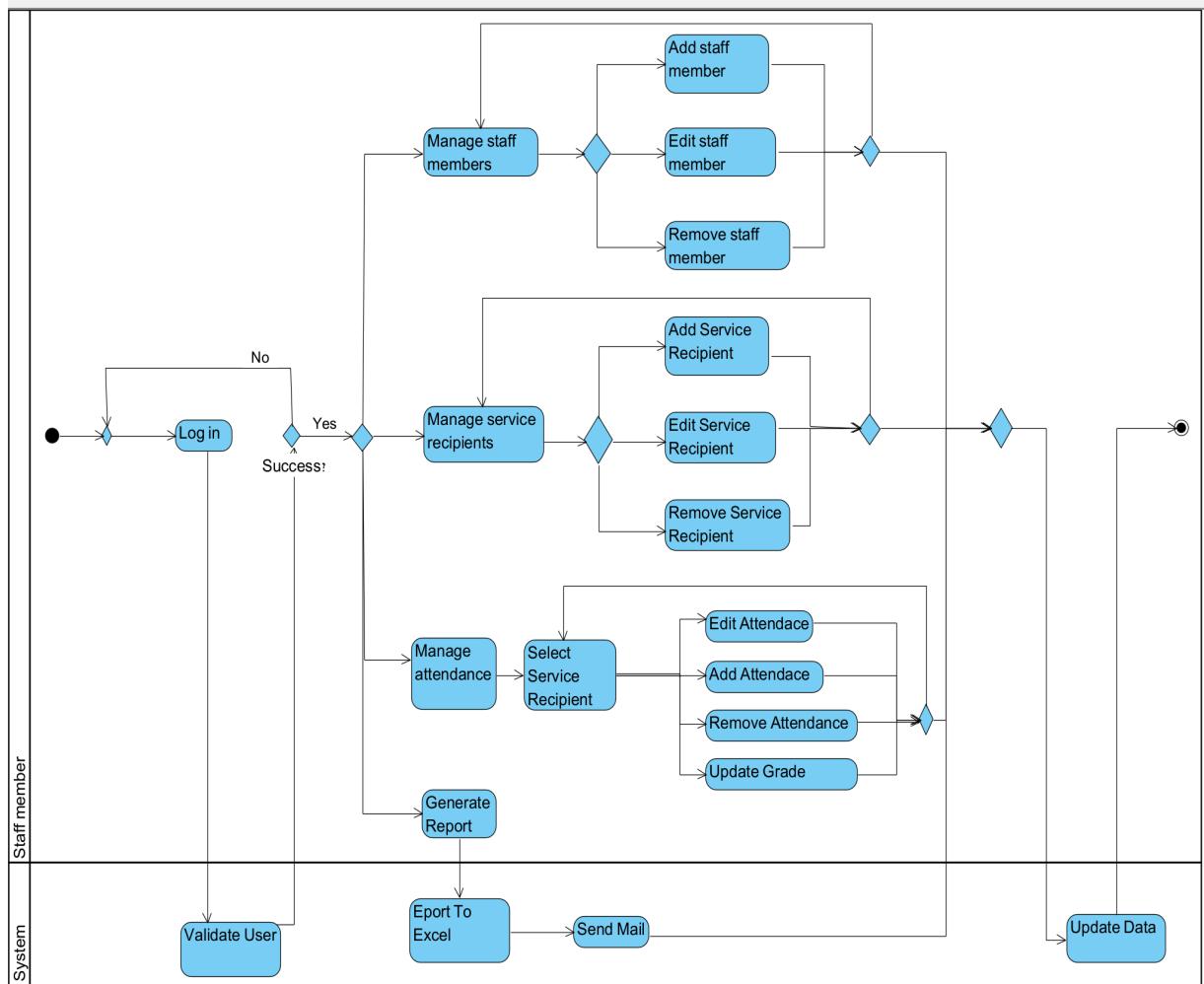


fig 4. Activity diagram - Staff actions in the system

4. Description of Development Process

4.1 Engineering Process

The Carmey Gil Center's digital platform was developed to meet the specific needs of individuals with intellectual disabilities. Using a client-server architecture the system simplifies daily processes for users and staff. Its primary goals are to replace manual attendance tracking with an automated, time-saving solution and provide an accessible tool for service recipients to share their emotional states.

Designed with accessibility in mind, the platform features large buttons, clear icons, multilingual support and voice feedback in the chosen language. These elements ensure ease of use for individuals with diverse abilities, promoting greater independence. Built with React.js for the interface, Node.js and Express.js for backend management, and MongoDB for secure data storage, the system enables real-time interaction and generates reports for daily operations.

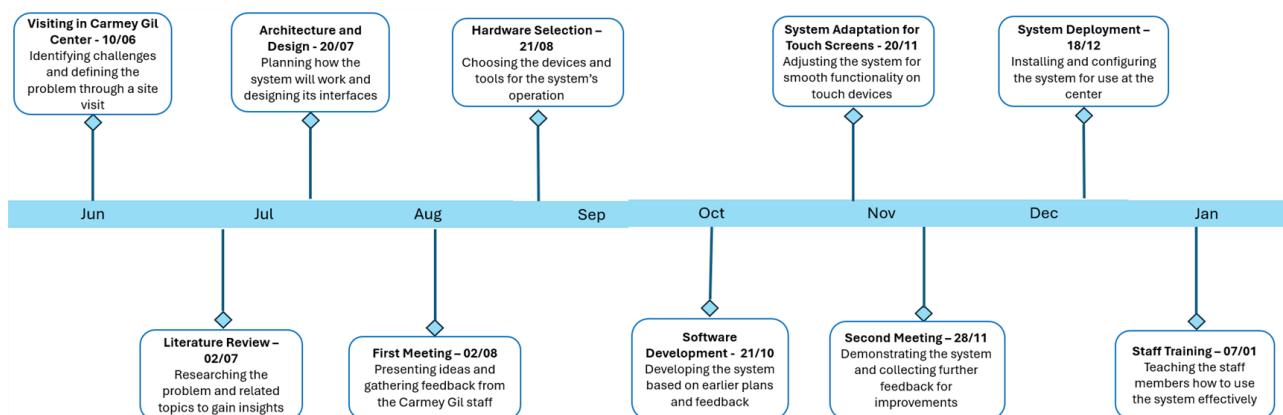


fig 5. Working Flow diagram

The flowchart above gives an overview of the steps we took to develop the system for the Carmey Gil Center. In the following sections, we will elaborate on the key parts of the process, from the early design and research stages to testing and deployment. These sections explain how each step helped us design and build a system that truly fits the center's needs.

4.1.1 Planning and Initial Research

The development of the Carmey Gil Center's digital platform began with thorough planning and research, including staff meetings, facility tours, and daily activity observations to address challenges like manual attendance tracking and emotional state monitoring. Communication with the center's manager via a dedicated WhatsApp group allowed for exchanging ideas, sharing prototypes, and refining the system. A literature review guided the creation of an accessible platform with simplified navigation, multilingual support, voice feedback, and clear visuals. This collaborative process ensured a user-friendly system that streamlined processes, reduced staff workload, and met the center's needs.

4.1.2 System Design and Architecture

The system's architecture was carefully designed to meet Carmey Gil specific needs. It employs a client-server model with a React.js frontend optimized for both touchscreens and desktop computers. The backend, built with Node.js and Express.js, efficiently handles API requests and business logic. MongoDB was chosen for its ability to manage complex data like attendance logs. To make the system easier to understand, detailed diagrams and flowcharts were created, presenting the workflows and data interactions in a clear and accessible way for the Carmey Gil staff.

4.1.3 Prototyping and Feedback

The development process was a collaborative effort, with Carmey Gil staff playing a key role in shaping the system. Early prototypes were shared to gather feedback, allowing us to improve functionality and enhance user experience. Staff opinions also guided the decision to use tablets for each group, ensuring simple and accessible attendance logging. Regular meetings and open communication ensured the system addressed the center's unique needs, creating a tool that effectively supports both staff and service recipients.

4.1.4 Adapting the System for Tablet Use

The system was initially developed as a web-based application for use on computers. However, to make it accessible for service recipients, we needed to adapt it for tablets as well. This required adjustments to ensure compatibility and usability across both platforms.

The adaptation process included redesigning the user interface to support touch gestures like taps and swipes, with large, user-friendly interactive elements. The layout was tested and optimized for various tablet screen sizes, ensuring a clean and accessible design.

Despite focusing on tablet compatibility, the system remains fully functional on computers, giving Carmey Gil the flexibility to use either platform based on their preferences. This dual-platform approach ensures that the system is both practical and accessible, providing a seamless experience for service recipients and staff alike.

4.1.5 Rigorous Testing and Deployment Plan

To ensure that the system was fully ready for use at the Carmi Gil Center, we conducted extensive testing throughout the development process.. Functional and usability tests ensured features like attendance tracking, multilingual support, and voice feedback worked seamlessly. The app was tested on a few different tablets and computers to ensure accessibility and smooth operation. A planned launch day included staff training sessions, with support provided to address any challenges. This thorough process ensured a smooth transition and successful implementation at the center.

4.2 User evaluation meeting - Carmey Gil Center

After several weeks of development, during which we had enough material to present, we met with the staff at Carmey Gil to present the system and gather their feedback. The goal was to make iterative adjustments based on their direct input. During the demo, we presented the various features that had been developed in accordance with the principles of user-centered design (UCD), and received comprehensive insights into the user experience, along with various suggestions for better adapting the system to their specific needs.

In addition to feedback on the user experience, the staff provided critical perspectives on the screen design to ensure the system was accessible and suitable for both the Carmey Gil staff and the service recipients with special needs. During the discussion, the staff decided that it would be better to use tablets for the purpose of recording the attendance of the service recipients, with one device designated for each service group.

An important point of discussion was the functionality of the system for recording performance scores. While the main focus remained on attendance management, the center expressed interest in incorporating daily performance tracking for each service recipient.

The meeting ended with a noteworthy discussion on technical expertise among the staff. Recognizing a lack in technological familiarity, the manager suggested bringing in a more technically skilled colleague. We mutually agreed to schedule a comprehensive training session with the initial deployment of the system, to ensure smooth integration and user adoption.

The meeting left both parties with positive momentum and a shared commitment to completing the development of the digital platform, while also being enthusiastic about bringing the innovative solution to full operational status at Carmey Gil.

5. Description of System Deployment and operation

5.1 System Deployment

The system is designed to function as a web-based application accessible through tablets for service recipients, staff and computers for staff Figure 6 presents the main components of the system:

5.1.1 Client Side (Frontend):

- The user interface is developed using React.js, providing an intuitive and responsive design tailored for both tablets and desktop environments.
- Each tablet is pre-configured with a web link to the system, allowing service recipients to easily log their attendance and emotional state.

5.1.2 Server Side (Backend):

- Managed using Node.js with Express.js to handle API requests efficiently.
- The server is responsible for processing data from the frontend, such as attendance record.
- Processes data about attendance records, generates detailed Excel reports, and handles their automatic delivery via email.
- Supports sending notifications and alerts through email to the manager.

5.1.3 Database:

- MongoDB serves as the database for the system, storing all service recipients data, staff member data, and attendance logs.
- The database is designed to support real-time updates, ensuring that data is always up-to-date.

5.2 System Operation

Service Recipients features:

- Tablets are used to access the system.
- Each recipient selects their guide, chooses their profile, and logs their emotional state.
- Attendance time is recorded and sent to the server for storage, the emotion selected requires additional processing, and if the emotion selected is negative, an email is sent to the administrator.

Staff features:

- Staff members access the system via desktop computers or tablets.
- Staff members manage service recipient profiles, manage staff members profiles, manage attendance of service recipients, generate attendance reports. The manager receives alerts for unusual emotional states.
- Manage service recipient profiles allows staff members to add, edit, and remove service recipients.
- Managing staff profiles allows staff members to add, edit, and remove staff members.
- Manage Attendance of Service Recipients Allows staff members to manually add attendance, update an attendance record, delete an attendance record, and update a service recipient's grade.

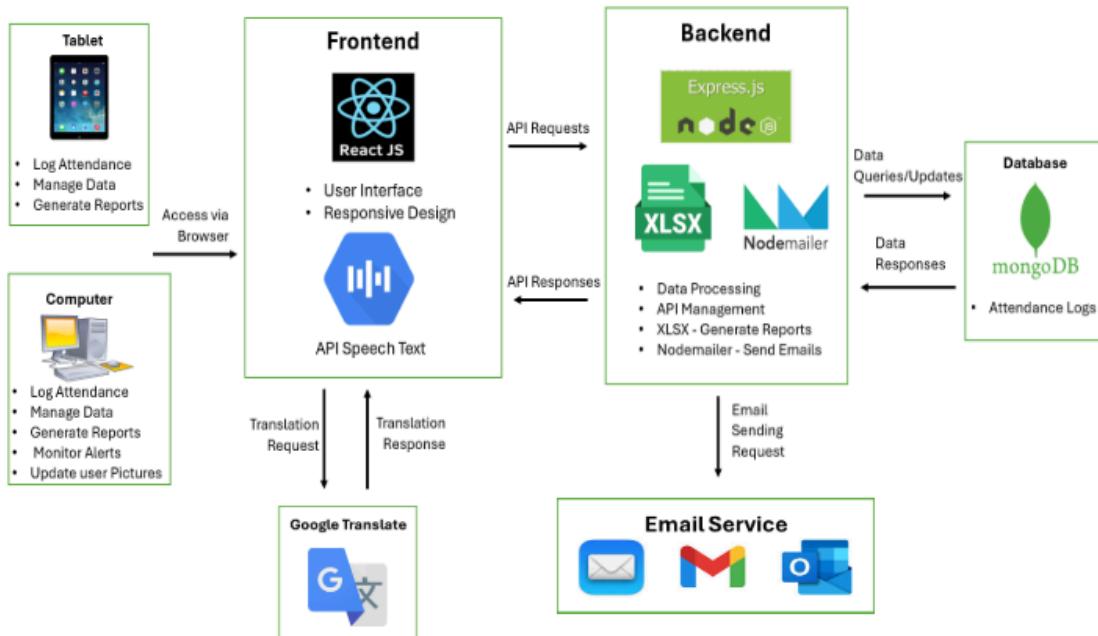


fig 6. system architecture diagram

5.3 Core Features

The development of the system's core features was carried out iteratively, ensuring functionality, accessibility, and ease of use. Each feature was carefully designed to address the specific needs of service recipients and staff.

5.3.1 Attendance Logging

Service recipients check in by selecting a photo of their guide, selecting their own photo, and selecting an emotional state that reflects their current mood, using an intuitive emoji-based interface designed for tablets. The system incorporates large, visually appealing icons to accommodate those with visual or motor impairments. Voice feedback confirms each action, providing reassurance and reducing errors for users who may have difficulty reading or cognitive processing. This feature ensures that checking in is a seamless and inclusive experience.

5.3.2 Emotional State Alerts

If a user selects a negative emotional state, such as sadness or anger, the system instantly generates an alert to notify the manager. This feature ensures that staff are aware of users' emotional needs in real-time, enabling timely interventions. By prioritizing users' emotional well-being, the system helps staff provide personalized support and fosters a safe, supportive environment.

5.3.3 Service Recipients Management

Staff can easily manage client profiles, using a central interface to add, remove, and update client details. This feature ensures that client details remain accurate and organized, which is critical to managing their information and operating the entire system.

5.3.4 Staff Management

The system allows staff members to easily manage staff profiles. They can add, remove or update staff member details. Similar to managing service recipients. This feature allows staff members to manage all staff members for the smooth operation of the attendance registration system.

5.3.5 Attendance Editing

To account for unforeseen circumstances, such as missed entries or errors, the system allows staff to manually edit attendance records. This feature provides flexibility for handling special cases, such as users who forgot to log their attendance or instances where corrections are needed for data accuracy. By enabling such adjustments, the system ensures that attendance tracking remains reliable and up-to-date.

5.3.6 Grade Updates for Service Recipients

The team can assign and update performance scores for service recipients, which are integrated into attendance reports. These grades provide a comprehensive overview of each user's participation and engagement over time. These scores are important for the Carmey Gil staff member in assessing the productivity of each service recipient throughout the day.

5.3.7 Report Generation

The system generates detailed attendance reports in Excel format, incorporating user scores. Staff can select an email address to receive these reports directly, streamlining record management and enabling efficient data sharing.

5.3.8 Tablet Keyboard Integration

The system includes an on-screen keyboard specifically designed for tablets, allowing for easy data entry directly on the device. The keyboard design is suitable for touch interactions, ensuring smooth functionality even on smaller screens. This feature improves usability, especially with compact and preferred devices.

5.3.9 Keyboard Input Support for Other Devices

Beyond tablets, the system supports standard keyboard input for use on desktop computers and laptops. This ensures compatibility with a wide range of devices and platforms, allowing staff to choose the setup that best suits their workflow. The system's flexibility enables seamless transitions between devices, ensuring a consistent user experience.

5.3.10 Multilingual Support

The system supports multiple languages, including Hebrew, English, Arabic, and Russian. Users can select their preferred language on the main screen, and the system automatically adjusts all subsequent pages to the chosen language. This feature ensures accessibility for a linguistically diverse user base . Combined with voice feedback, the multilingual support simplifies interactions for users with language barriers.

5.3.11 Voice Feedback

Voice feedback is integrated into the system to confirm every action performed by service recipients. For example, when a user selects their name or emotional state, the system reads out the selection, providing reassurance and reducing errors. This feature is essential for individuals with visual impairments or cognitive challenges, as it helps bridge communication gaps and ensures a smoother, more intuitive user experience.

5.4 Report

The attendance report is an Excel file that allows instructors to track their group's attendance data. Each instructor has a dedicated worksheet that displays the data for the service recipients belonging to their group. The leftmost column displays the report dates, followed by two columns for each recipient: one for attendance times (in and out) and one for grades set by the instructor. The report accurately reflects the system data and presents the data as it appears in it. These reports provide instructors with a clear and efficient overview of attendance and performance for the service recipients belonging to their group.

A	B	C	D	E
1 תאריך		צ'יון עברו יוסף דגניה	Յօסֵף דָגְנִיה	צ'יון עברו משה אוריאל משה אוריאל
2 1.12.2024	אין נוכחות			אין נוכחות
3 2.12.2024	אין נוכחות			אין נוכחות
4 3.12.2024	נכח : 03:03-04:04	5		אין נוכחות
5 4.12.2024	אין נוכחות			אין נוכחות
6 5.12.2024	אין נוכחות			אין נוכחות
7 6.12.2024	אין נוכחות			אין נוכחות
8 7.12.2024	אין נוכחות			אין נוכחות
9 8.12.2024	אין נוכחות			אין נוכחות
10 9.12.2024	אין נוכחות			אין נוכחות
11 10.12.2024	נכח : 08:10---:--	5	12:48-15:48	67
12 11.12.2024	נכח : 06:24-11:02	10	12:46-15:46	63
13 12.12.2024	אין נוכחות			אין נוכחות
14 13.12.2024	אין נוכחות		08:51-15:51	29
15 14.12.2024	אין נוכחות			אין נוכחות
16 15.12.2024	אין נוכחות			אין נוכחות
17 16.12.2024	אין נוכחות			אין נוכחות
18 17.12.2024	אין נוכחות			אין נוכחות
19 18.12.2024	נכח : 18:04-22:48	77	12:25-17:27	1
20 19.12.2024	אין נוכחות		--:11:48	15
21 20.12.2024	אין נוכחות			אין נוכחות
22 21.12.2024	אין נוכחות			אין נוכחות
23 22.12.2024	אין נוכחות			אין נוכחות
24 23.12.2024	אין נוכחות			אין נוכחות
25 24.12.2024	אין נוכחות		---:17:06	68
26 25.12.2024	אין נוכחות			אין נוכחות
27 26.12.2024	אין נוכחות			אין נוכחות
28 27.12.2024	אין נוכחות			אין נוכחות
29 28.12.2024	אין נוכחות			אין נוכחות
30 29.12.2024	אין נוכחות			אין נוכחות

fig 7. exemple for attendance report in excel

The attendance system stores detailed records of service recipient's attendance, including dates, times, and grades. The image below displays a sample of the attendance records as they are stored in the system. These records were used to generate the monthly report for the guide, ensuring accuracy and consistency. By presenting both the stored data and the generated report, we demonstrate the system's reliability in processing and summarizing attendance information.

רשומות נוכחות		
18.12.2024	צ'ק אין: 12:25 צ'ק-אנווט: 17:27	זיהוי: 1
19.12.2024	צ'ק אין: -- צ'ק-אנווט: 11:48	זיהוי: 15
24.12.2024	צ'ק אין: -- צ'ק-אנווט: 17:06	זיהוי: 68
11.12.2024	צ'ק אין: 12:46 צ'ק-אנווט: 15:46	זיהוי: 63
10.12.2024	צ'ק אין: 12:48 צ'ק-אנווט: 15:48	זיהוי: 67
13.12.2024	צ'ק אין: 8:51 צ'ק-אנווט: 15:51	זיהוי: 29

fig 8. attendance log from the system - Moshe Uriel

רשומות נוכחות		
3.12.2024	צ'ק אין: 3:03 צ'ק-אנווט: 4:04	זיהוי: 5
10.12.2024	צ'ק אין: 8:10 צ'ק-אנווט: --	זיהוי: 5
11.12.2024	צ'ק אין: 6:24 צ'ק-אנווט: 11:02	זיהוי: 10
18.12.2024	צ'ק אין: 18:04 צ'ק-אנווט: 22:48	זיהוי: 77

fig 9. attendance log from the system - Yosef Degania

5.5 Alert - Negative Emotion selected

When completing the attendance registration process, the service recipient selects their current emotional state. To monitor the emotional well-being of service recipients at the Carmey Gil Center, the system sends an alert to the manager whenever a negative emotion is selected. This alert is sent directly to the manager's email, allowing them to respond quickly and effectively to the needs of the service recipient.

Here is an example of an alert received by the manager after a service recipient selected a negative emotion. The alert includes key details such as the service recipient's name, ID number, and the specific emotion they selected. This structured format ensures the manager has all the necessary information to address the situation promptly and accurately.



הודעה זו נשלחה בעקבות בחירת אימוג' חשור על ידי המטופל
שם: אופיר שחר
תעודת זהות: 314764283
הרגש שנבחר: כועס

fig 10. Example of an email due to choosing a bad emotional

6. User Documentation:

6.1 Home Page

This screen allows users to log in either as a service recipient or as staff. Additionally, the screen supports multiple languages. Language selection buttons are located at the bottom right of the screen. Clicking on a specific language button changes all system text to the selected language. From this screen, users can choose one of two paths based on their login type: attendance registration for service recipients or staff login for staff members. The text will be displayed in the selected language throughout the use of the system.

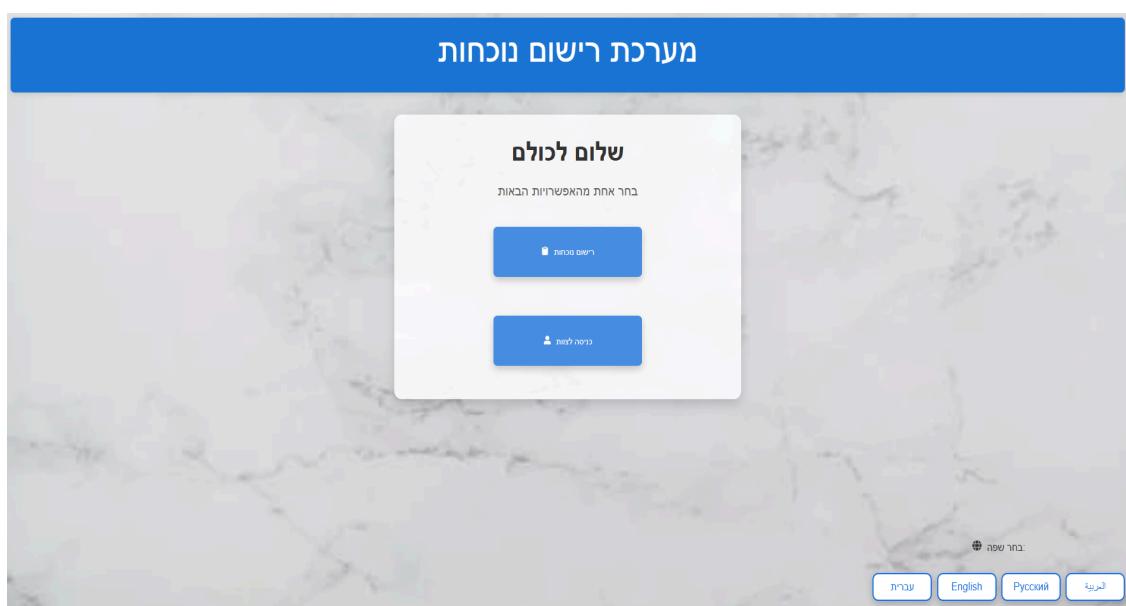


fig 11. Login screen

6.2 Guide Selection

After selecting "Attendance Registration" on the main screen, the service recipient proceeds to the "Guide Selection" screen. Here, photos of all group guides at the Carmey Gil center are displayed, with the name of each guide appearing below their photo. To continue the registration process, the service recipient selects their guide. When the user clicks on the guide, the system provides voice feedback, the name of the guide.

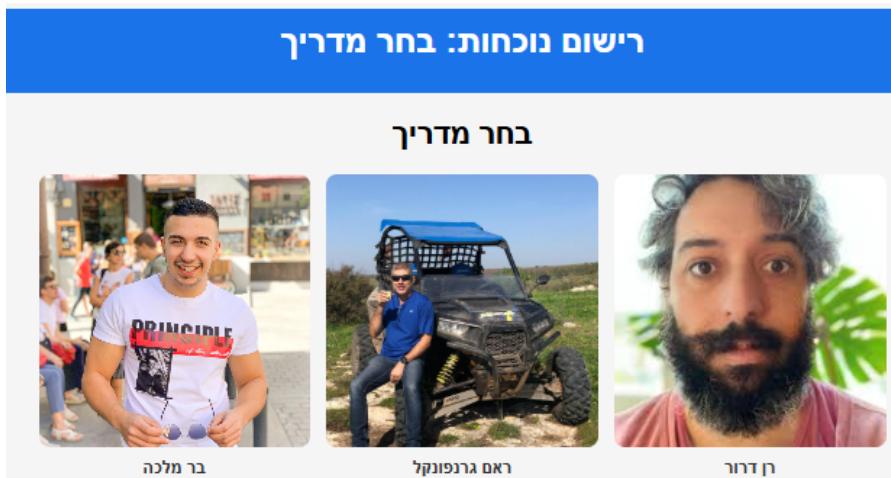


fig 12.Guide Selection screen

6.3 Service Recipient Selection

Once the service recipient selects their guide, they proceed to the "Service Recipient Selection" screen. This screen displays photos of all service recipients belonging to the selected guide's group. Similar to the previous screen, each photo is labeled with the recipient's name. The recipient selects their photo to proceed. When the user clicks on receive the service, the system provides voice feedback, the name of the service recipient.

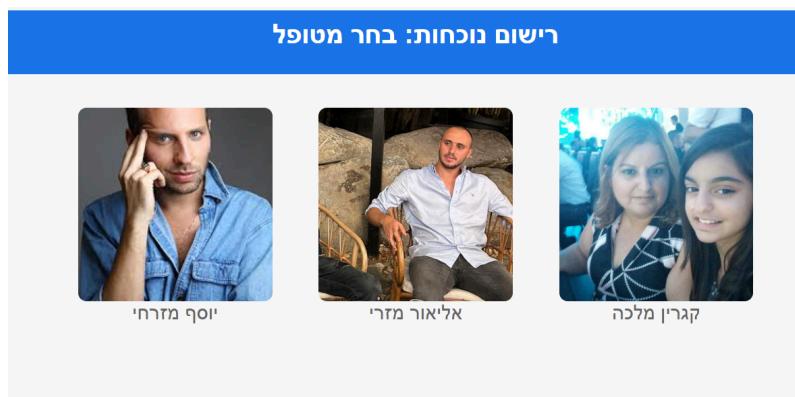


fig 13.Service Recipient Selection screen

6.4 Emotion Selection

After selecting their photo, the service recipient moves to the "Emotion Selection" screen. This screen displays emojis representing various emotions. The recipient selects the emotion that reflects how they feel at the time of registration. At this point, the attendance registration process is complete, and the information is saved in the system. If the service recipient chooses an angry or sad emoticon, an email will be sent to the manager with the details of the service recipient.

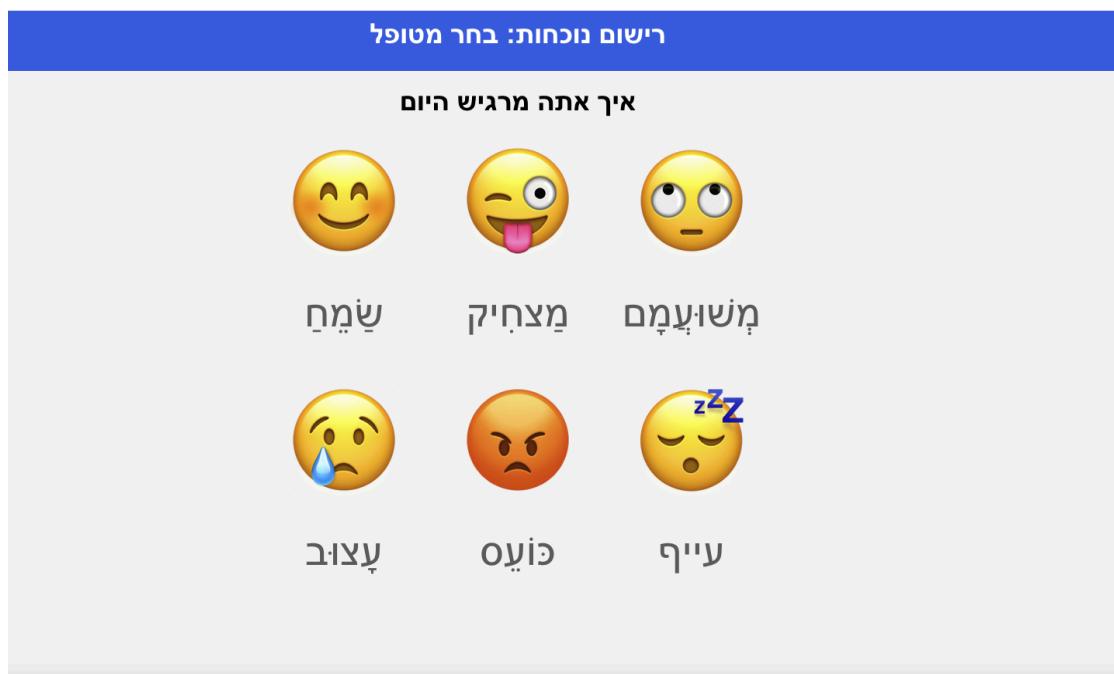


fig 14. Emotion Selection screen

6.5 Login Screen - Staff Members

In this screen, staff members are required to enter their username in the top field and their password in the bottom field. The system verifies that the username and password are correct. Upon successful login, the system directs the user to the staff member dashboard.

The on-screen keyboard allows users to enter data into the system. It is designed to support platforms such as tablets, ensuring accessibility for devices without a physical keyboard. This keyboard appears on any screen where data entry is possible.

Login details for check:

username - barmalka1419@gmail.com

password - 123456

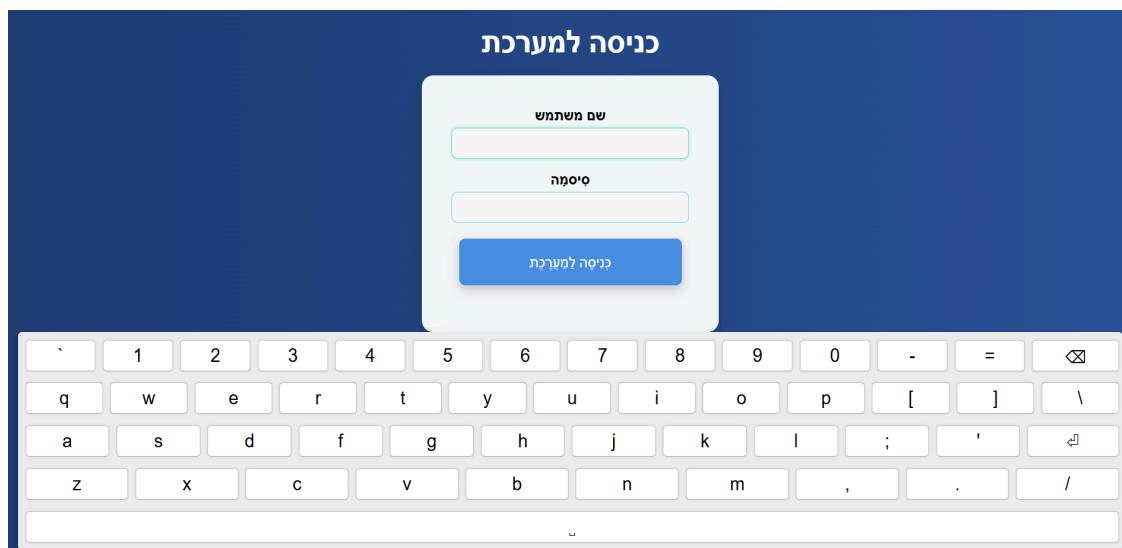


fig 15.Login Screen - Staff Members

6.6 Staff Members Dashboard

After logging in as a guide, the main dashboard for staff members appears. This dashboard provides four options: Manage Attendance, Manage Service Recipients, Manage Staff Members, and Generate Reports.

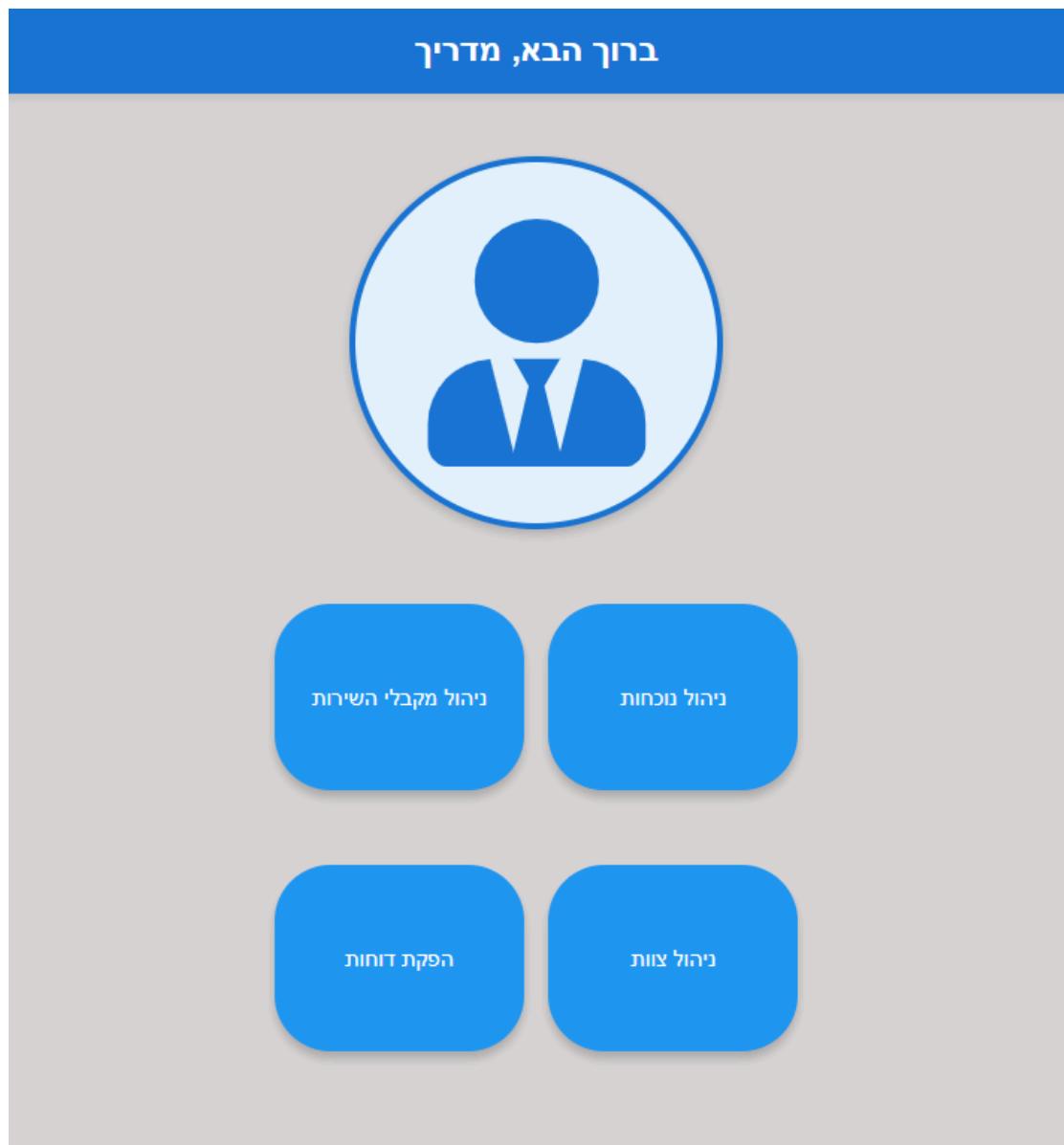


fig 16. Staff Members Dashboard

6.7 Manage Attendance

In this screen, the guide can manage the attendance records of the service recipients. A list of all service recipients is displayed on the left side of the screen. To update a specific recipient's attendance, the guide selects their name. Attendance records, including entry and exit times and four additional options: Edit Attendance, Delete Attendance, Update Score, and add attendance manually.

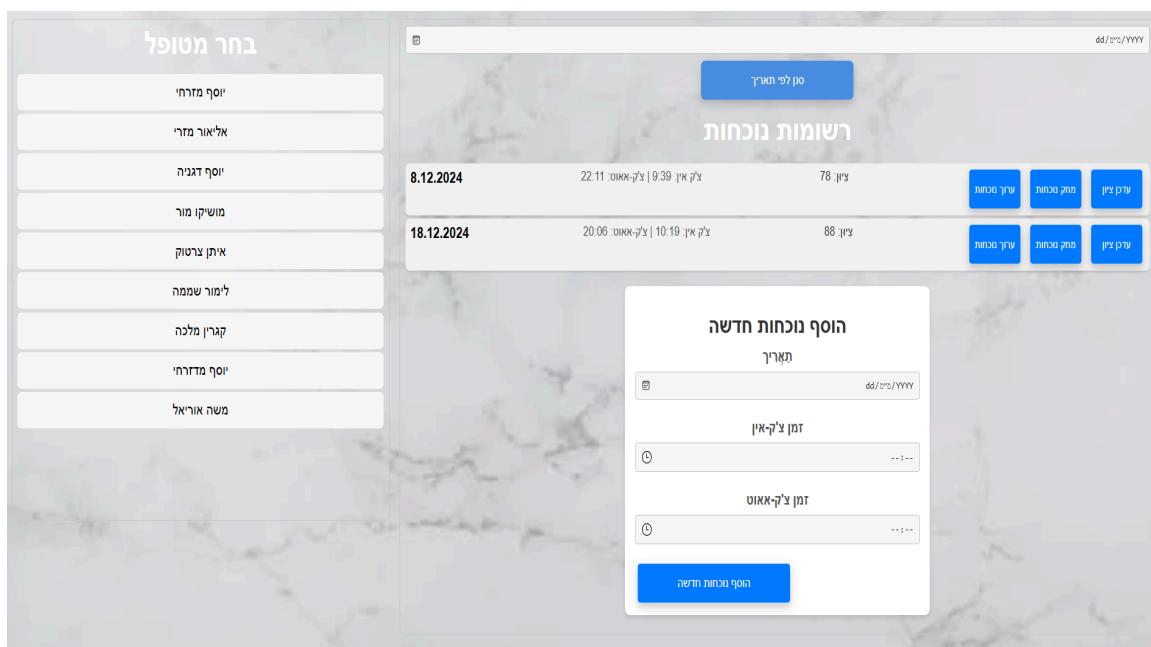


fig 17. Manage Attendance screen

6.7.1 Edit Attendance

When we click on the option to edit attendance that appears next to each of the attendance records, we will reach this screen. On this screen, we can edit the date and time of the record we want. The guide can click a filter and choose a date he wants to focus on. The screen has three fields: Date, Entry Time, Exit Time. These fields display the times that are registered in the system. To update the date, we click on the small calendar on the left side of the field and use it to select the date that we want to enter into the record. Similarly, we will update the attendance hours using the small clock that appears in each of the fields: Entry Time and Exit Time. To update the attendance and save it in the system, we click on the blue "Update" button. To cancel the operation and not make any changes, we click on the red "Cancel" button.

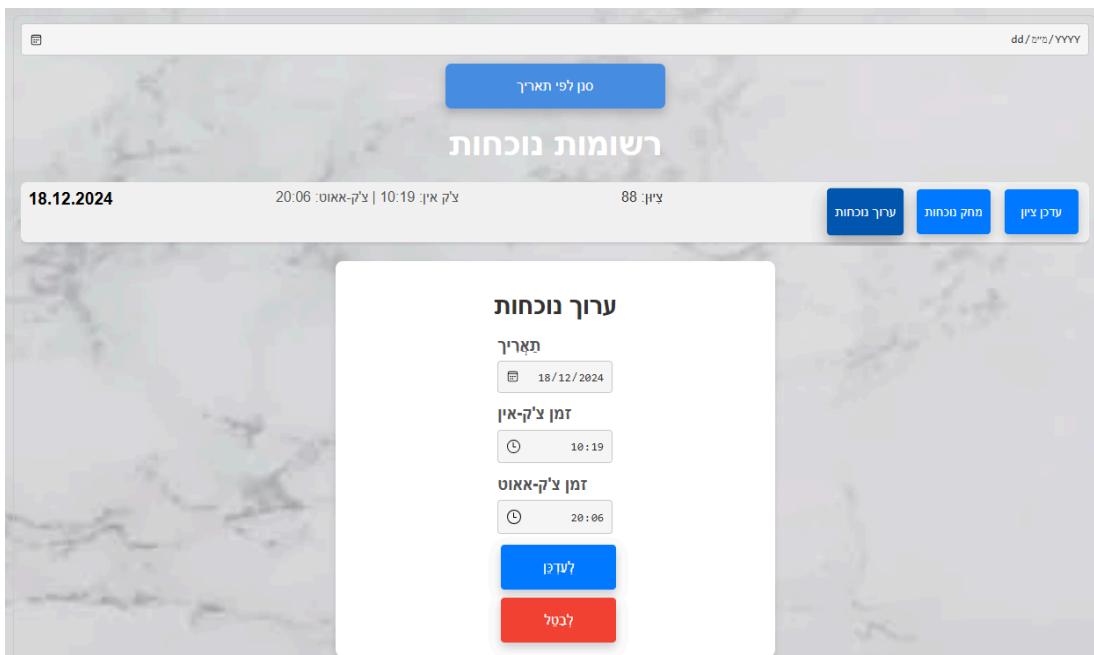


fig 18.Edit Attendance screen

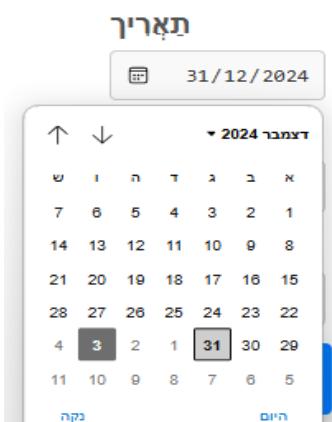


fig 19.Calendar to select a attendance date

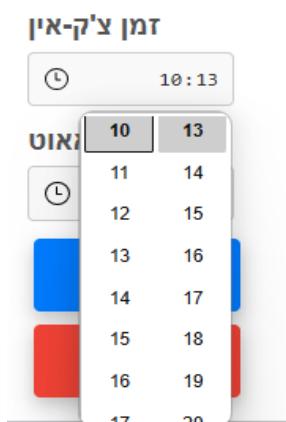


fig 20.Clock to select attendance time

6.7.2 Delete Attendance

To delete an attendance record, click the Delete Attendance button which appears next to it.

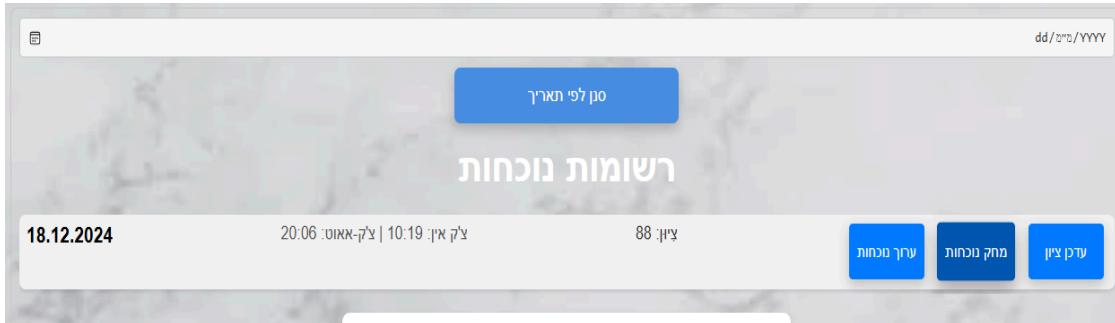


fig 21.Delete Attendance screen

6.7.3 Update grade

When we want to update a grade for a particular record, we click on the "Update Grade" button that appears next to it. When we click on the button, no grade will appear for a new record and for a record that already has a grade, a window will open with the number with the grade. When we stand on the right part of the field where the grade appears, we will have two arrows (up and down) when clicking on the up arrow will increase the grade by 1 and clicking on the down arrow will decrease the grade by 1. In addition, the guide can update through the keyboard by entering a number. When we reach the desired grade, we click on the left blue button "Update Grade" and the grade will be updated in the attendance record. To cancel the operation, we click on the left red button "Cancel Update".

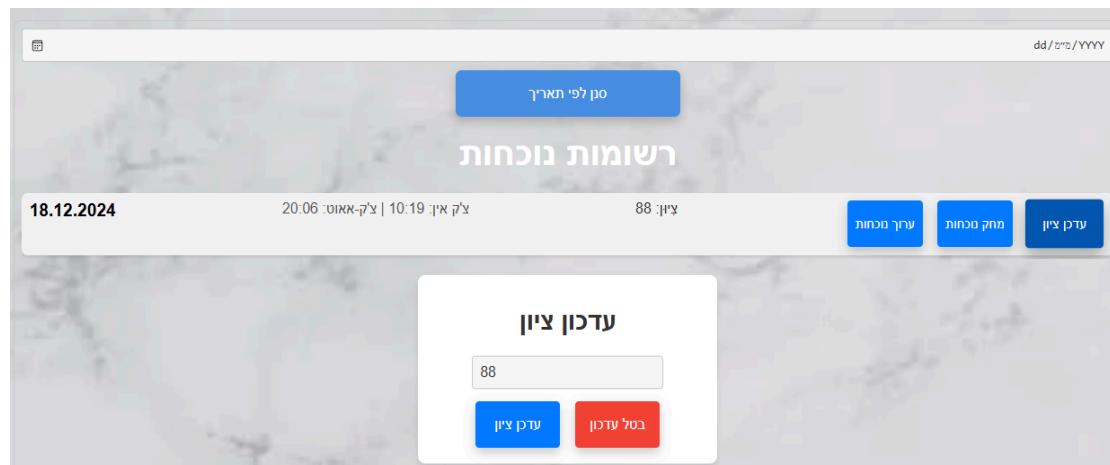


fig 22.Update grade screen

6.7.4 Add Attendance

To add a new attendance record, we will fill in the following fields: date, check-in time, check-out time. To enter the date, we click on the small calendar on the left side of the field and use it to select the date that we want to enter into the record. Similarly, we will enter the attendance hours using the small clock that appears in each of the fields: Entry Time and Exit Time. The guide can enter these details using the keyboard. To add the attendance record with the data we filled in, we click the "Add new attendance" button.

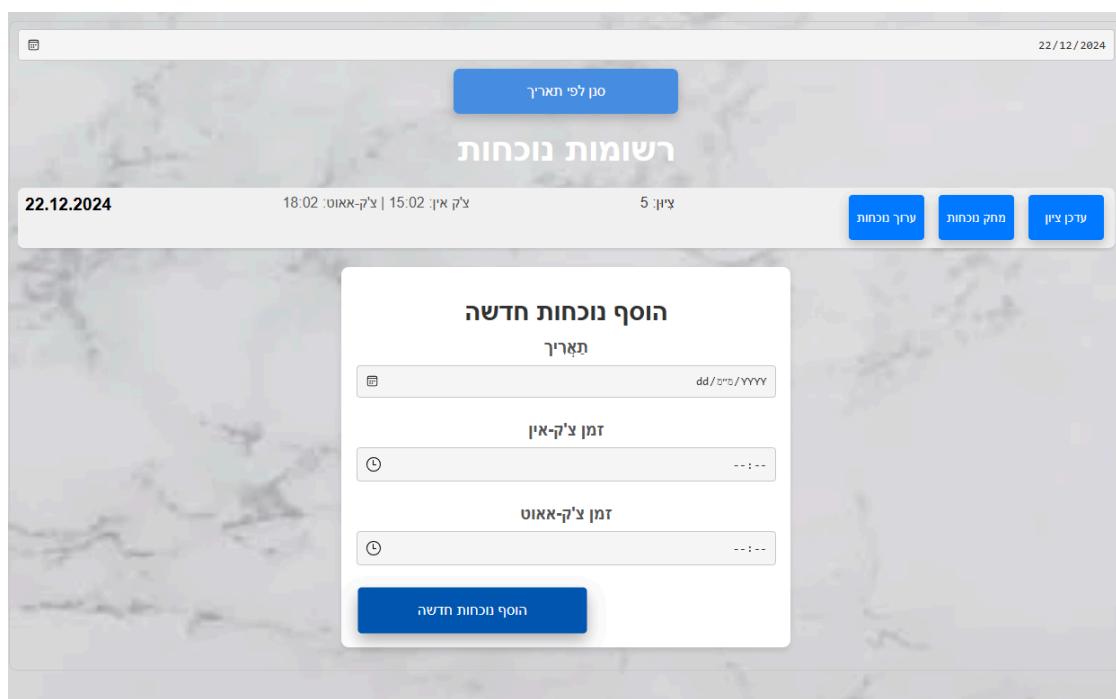


fig 23.Add Attendance screen

6.8 Generate Reports

On this screen, the staff member selects a month at the top of the page for which the report will be generated. To select the date, they use the calendar that appears on the left side of the top field. At the bottom of the screen, the staff member enters an email address to which the report should be sent. They can type the email address using an intuitive keyboard that opens at the bottom of the screen. After filling in the required details, the staff member clicks the "Generate Report" button, and a report containing all attendance data stored in the system for the selected month is sent to the specified email address.

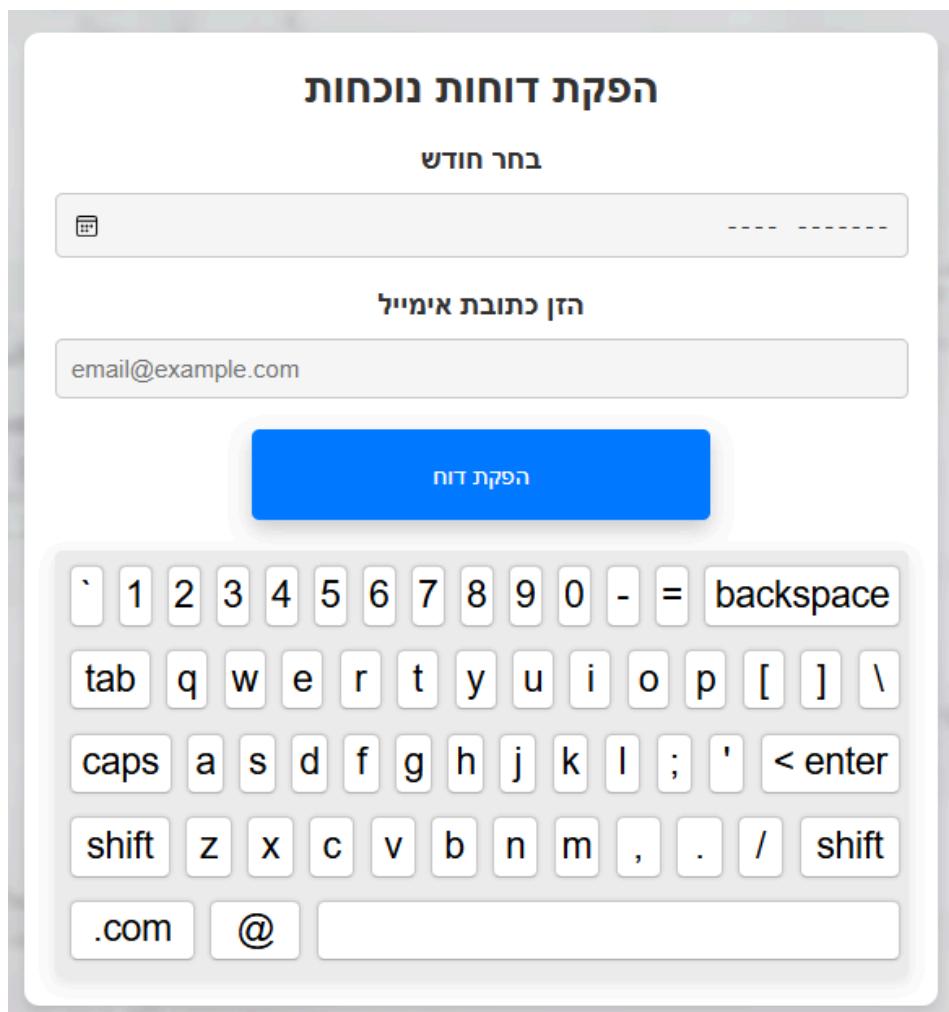


fig 24. Generate Reports screen

6.9 Service Recipient Management

In this screen, three options are available: Edit Service Recipient, Delete Service Recipient, and Add Service Recipient.

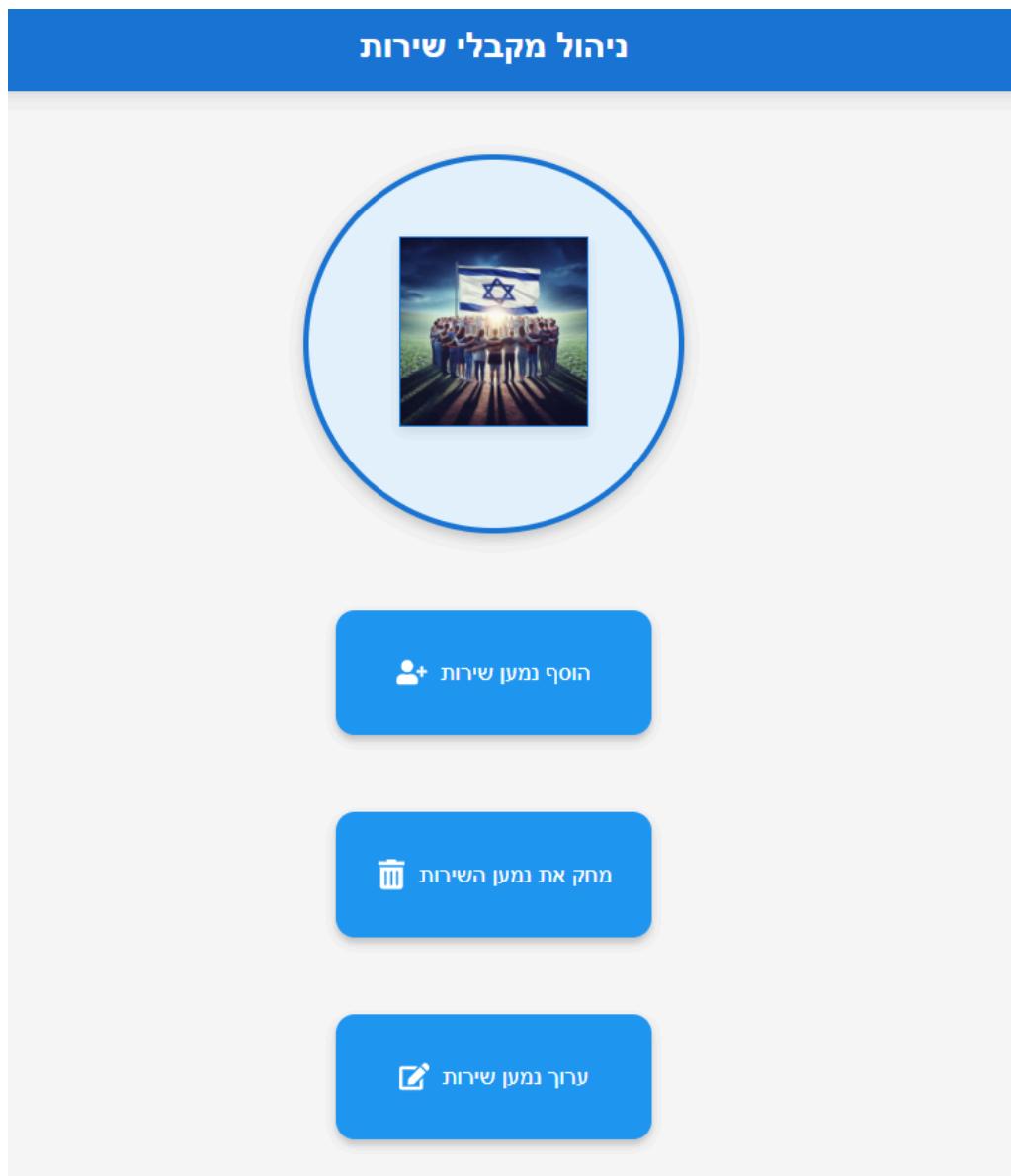


fig 25. Service Recipient Management screen

6.10 Add Service Recipient

Clicking on this option opens a screen for entering all the required details for the new service recipient: ID, name, photo, and group. To select the guide assigned to the service recipient, the user clicks on a dropdown menu, which displays a list of all the guides in the center, allowing them to choose the desired guide. The guide then enters the remaining details using the keyboard and clicks the "Add" button at the bottom to save the new service recipient.

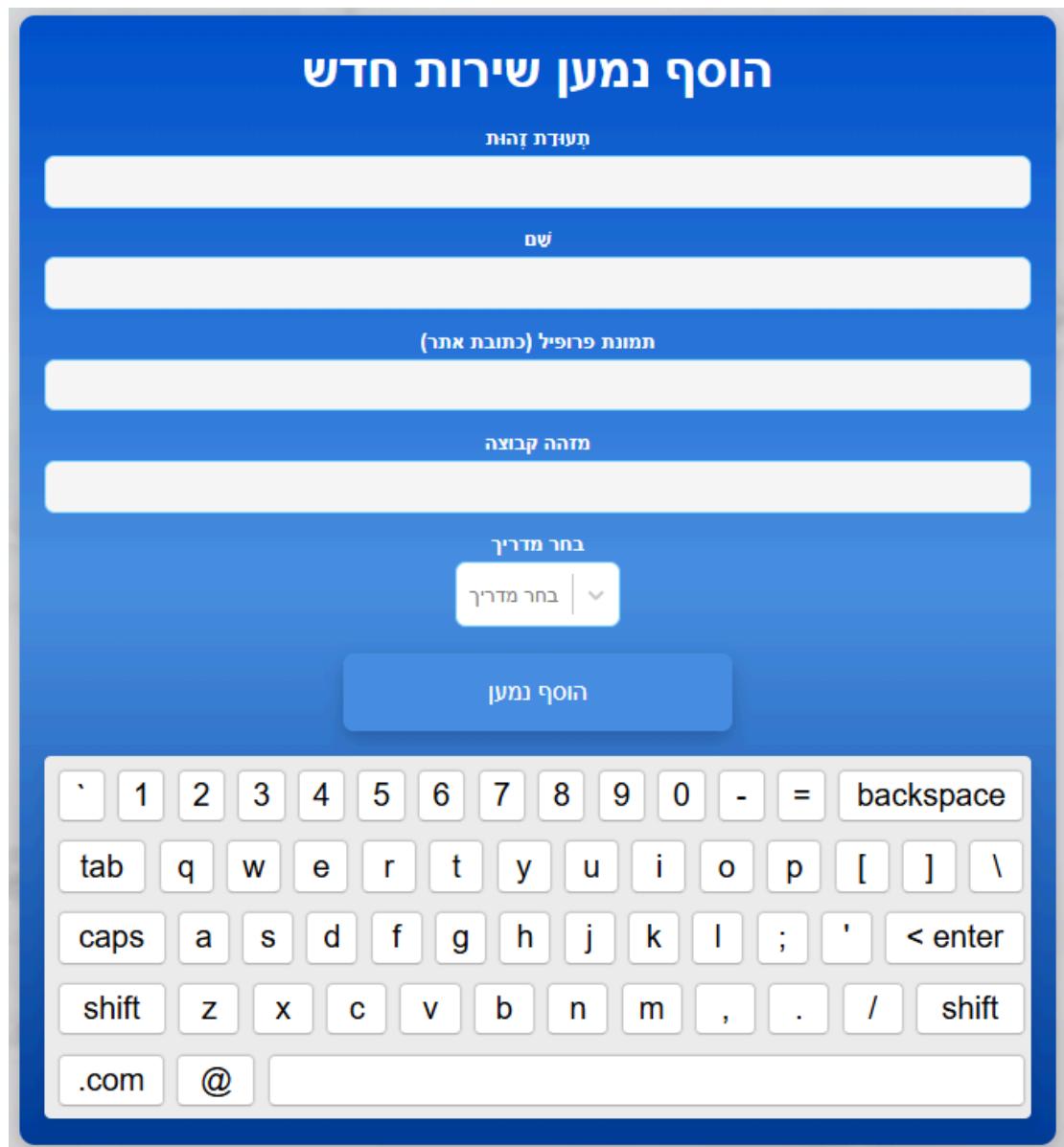


fig 26.Add Service Recipient screen

6.11 Edit Service Recipient

This screen displays all service recipients in the system. To edit a recipient, the guide clicks the photo of the recipient, the system loads the data about that recipient. Once selected, all fields: ID, name, email, photo, group, and guide appear for editing. The guide changes the details that he wants to change. A keyboard pops up to allow changes. After the updates, the guide clicks the "Save Changes" button to save the edits.

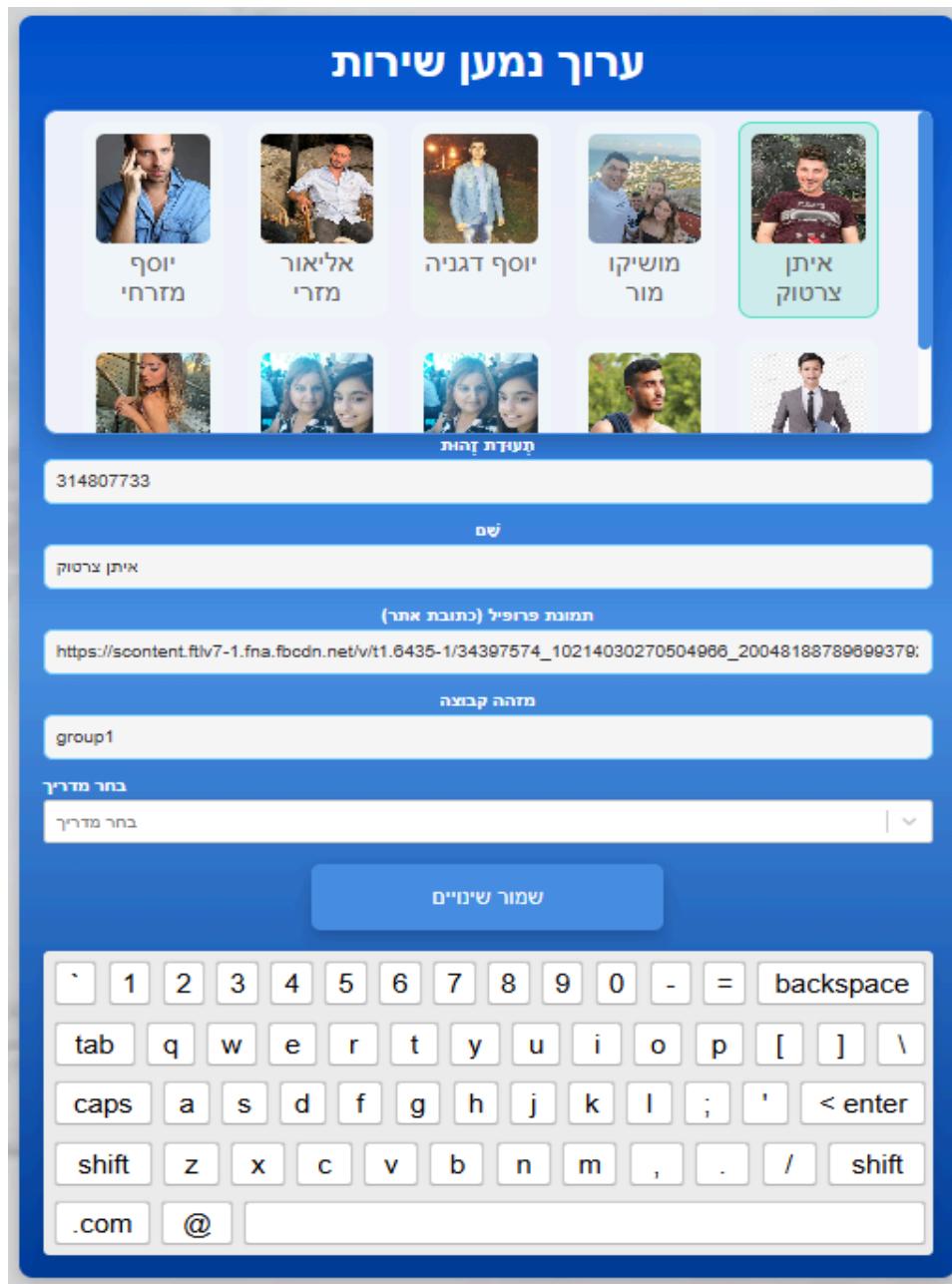


fig 27. Edit Service Recipient screen

6.12 Delete Service Recipient

This screen displays all the service recipients. To delete a recipient, the manager clicks their photo and then the "Delete" button at the bottom.

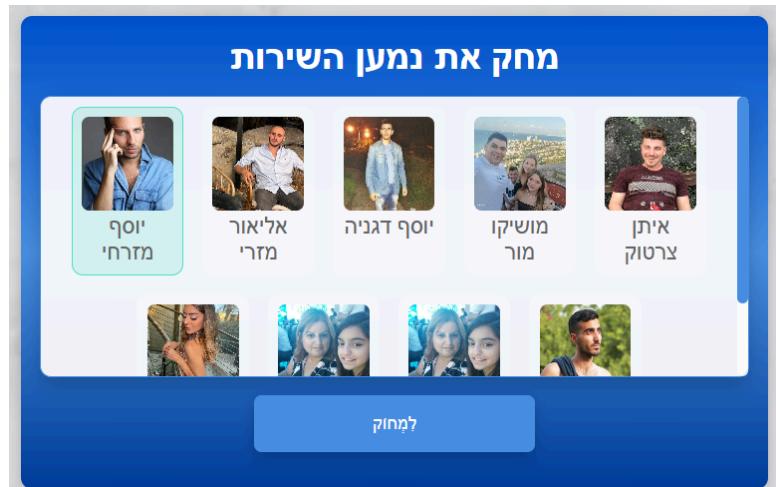


fig 28.Delete Service Recipient screen

6.13 Staff Management

This screen, similar to the Service Recipient Management screen, has the same staff member management options: Edit staff Member, Delete staff Member, and Add staff Member.



fig 29.Staff Management screen

6.14 Edit Staff Member

This screen displays all guides with their photos and names. To edit a guide, click their photo. the system loads the data about that guide. Once selected, all fields (ID, name, email, password, and image link) appear for editing. A keyboard pops up to make changes.

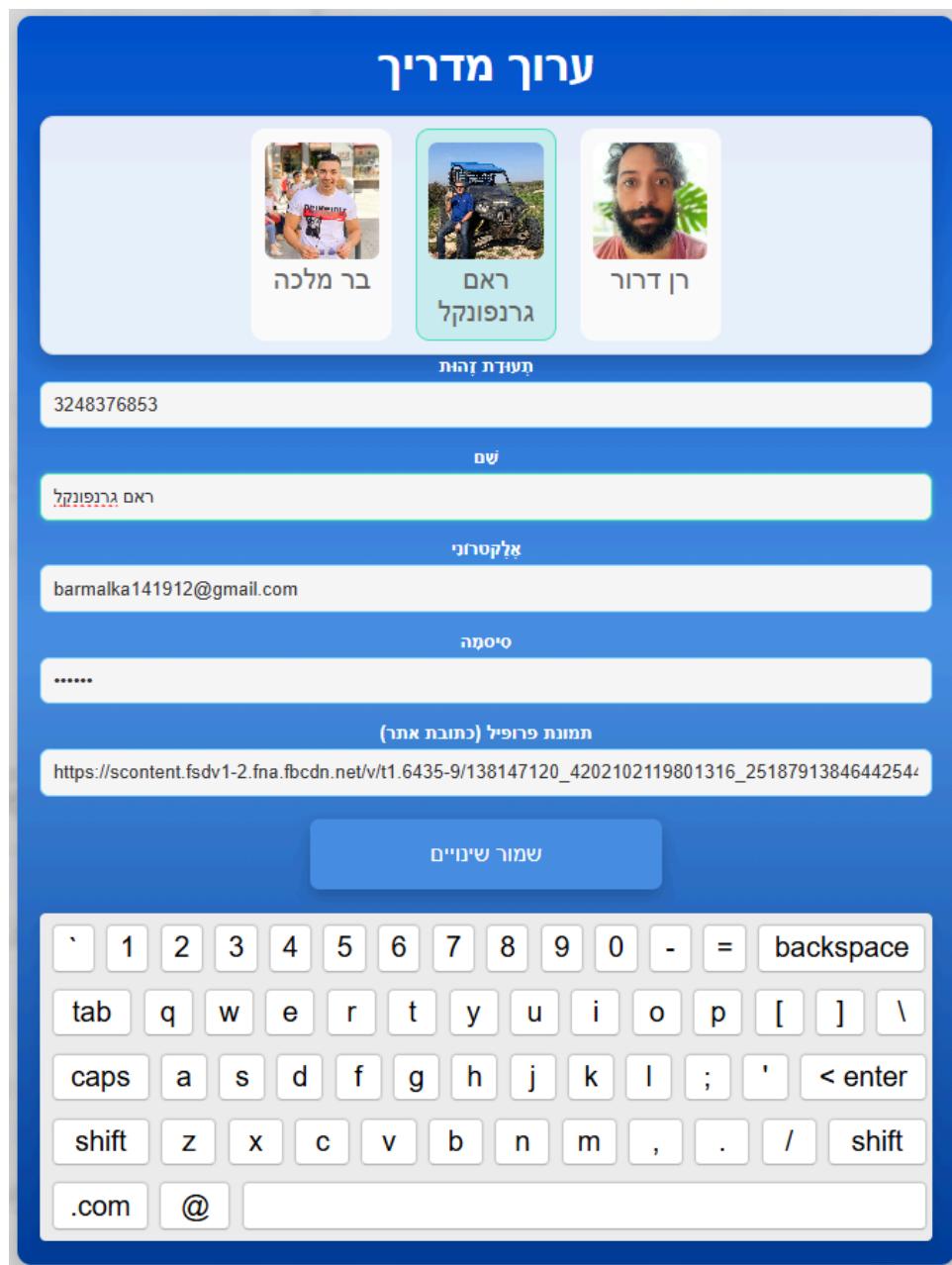


fig 30.Edit Staff Member screen

6.15 Delete Staff Member

This screen displays all guides. To delete a guide, the manager clicks their photo and then the "Delete" button at the bottom.

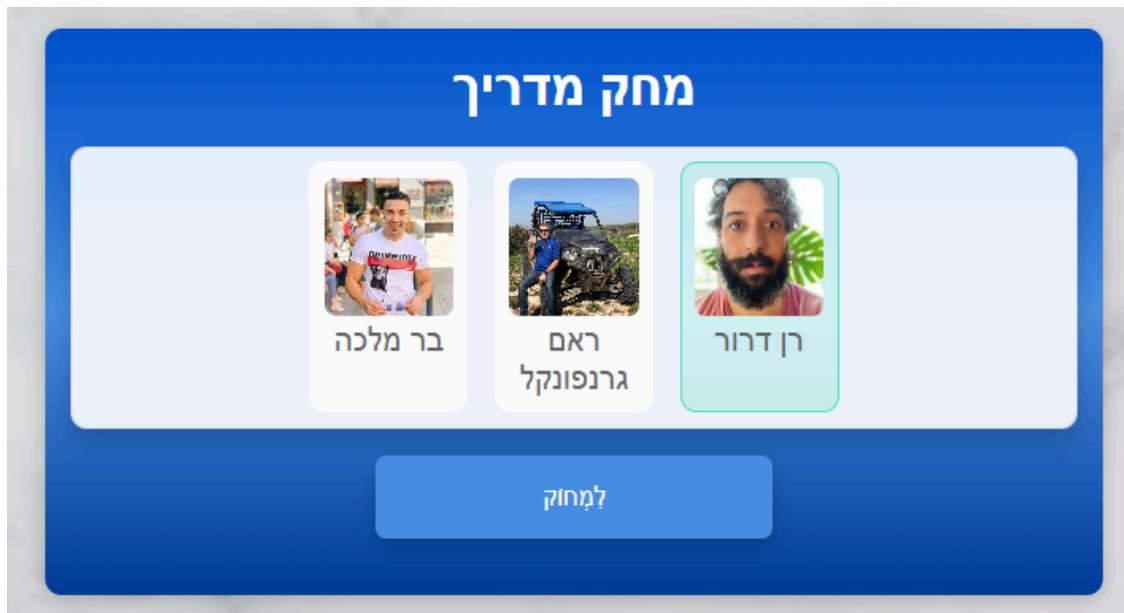


fig 31. Delete Staff Member screen

6.16 Add Staff Member

Clicking this option opens a screen to input details for new staff members: ID, name, email, password, and image url. The manager enters the details using the keyboard and clicks the "Add Staff Member" button at the bottom to save the new staff member.

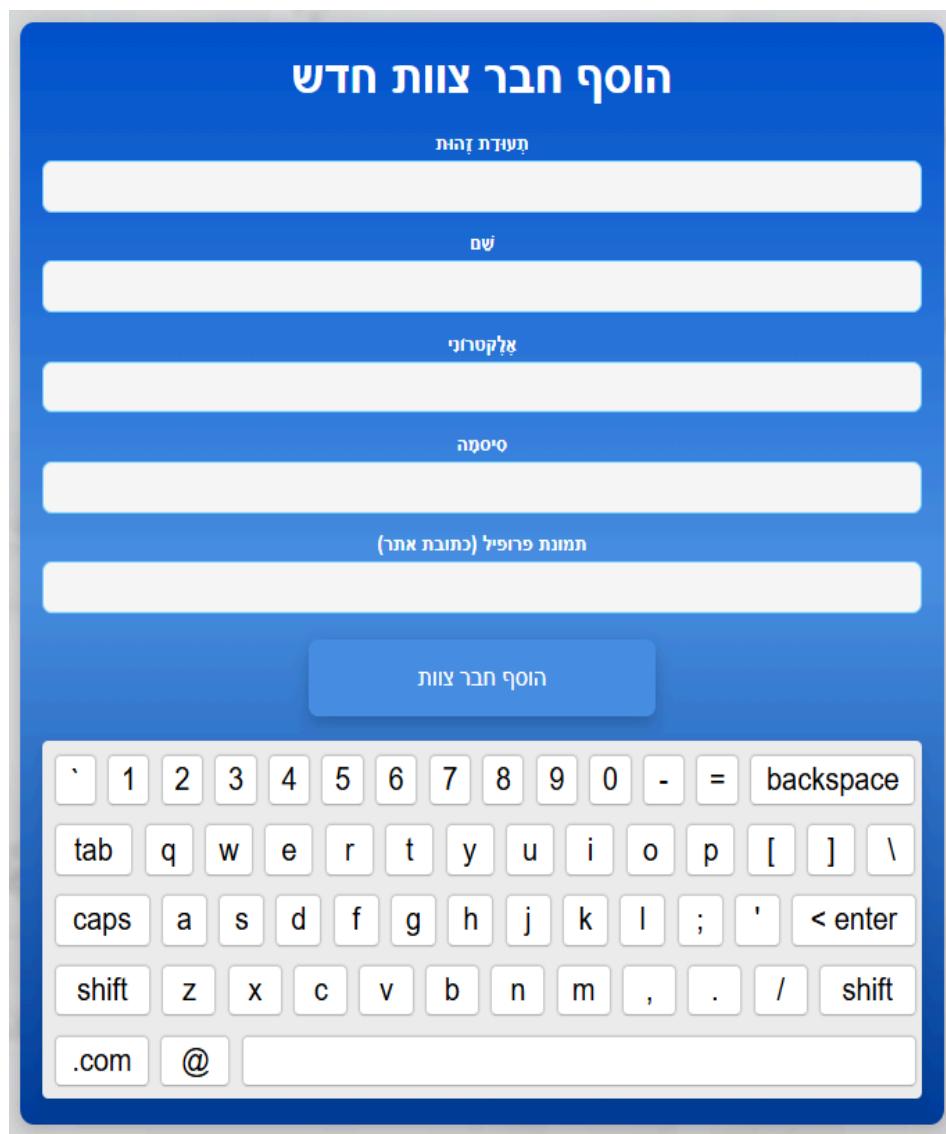


fig 32.Add Staff Member screen

7. Description of Testing process

7.1 Testing Plan

To ensure the reliability and functionality of our system, we conducted a series of tests throughout the development process, tailored to the specific features we implemented. After completing the development of each feature, we thoroughly tested it along with all previously developed features to ensure the new edition did not interfere with the existing functionality. This process was repeated for every new feature, allowing us to identify and resolve any potential issues early on. Once all features were fully developed, we performed comprehensive testing of the entire system to confirm that all features worked seamlessly together and that the platform operated smoothly as a whole.

7.1.1 Login Tests

Test	Test Description	Expected Result	Result
1	Successfully login with valid username and password as staff	The system opens a staff member's screen.	Pass
2	Fail login with invalid username or password as staff	The system displays: 'Guide not found in the system.'	Pass
3	Simultaneous attendance registration	The registration process will be successful.	Pass

table 1. Login Tests

7.1.2 Attendance Registration Tests

Test	Test Description	Expected Result	Result
4	Service recipient selects a guide, then their picture, and finally an emotion emoji	The system logs attendance and displays: 'registration success.'	Pass
5	Service recipient selects a sad emotion emoji	The system sends mail to manager	Pass
6	A registered service recipient is trying to register again	The system displays: 'Attendance already complete for today'	Pass

table 2. Attendance Registration Tests

7.1.3 Staff Actions Tests

Test	Test Description	Expected Result	Result
7	Add a new service recipient	The system displays: 'Service recipient successfully added.'	Pass
8	Edit service recipient	The system displays: 'Service recipient successfully edited.'	Pass
9	Remove an existing service recipient	The system displays: 'Service recipient successfully removed.'	Pass
10	Add a new staff member	The system displays: 'Staff member successfully added.'	Pass
11	Edit staff member	The system displays: 'Changes have been saved'	Pass
12	Remove an existing staff member	The system displays: 'Staff member successfully removed.'	Pass
13	Add a new attendance list	The system displays: 'The action completed successfully.'	Pass
14	Edit attendance list	The system displays: 'The action completed successfully.'	Pass
15	Remove a attendance list	The system displays: 'The action completed successfully.'	Pass
16	Update grade	The system displays: 'The action completed successfully.'	Pass

table 3. Staff Actions Tests

7.1.4. Report Tests

Test	Test Description	Expected Result	Result
17	Create report	The system displays: 'Report sent successfully.'	Pass
18	Testing Sending an Empty Report	The system displays: "Please fill in all fields"	Pass

table 4. Report Tests

8. Maintenance Guide

8.1 Installation

8.1.1 Prerequisites

- Visual Studio Code - for editing and running code.
- Node.js and npm - for both server-side and client-side JavaScript.
- MongoDB - Used as the database for the project.
- Git - For version control and cloning the repository.
- Browser - Any modern browser to access the frontend.

8.2 Setup

8.2.1 Clone the Repository

- Clone the project repository from GitHub:

<https://github.com/barmalka1419/Attendance-management-system-Carmey-Gil-Center-/tree/main>

- Deployed Application:

The system is also deployed and accessible at the following URL:

[Attendance Management System on Render](#)

- If you choose to use the deployed version, please note the following:
Wait for the Backend Server

8.2.2 Option 1: Running Frontend Only (Using Render Backend)

In the current setup, the backend server is hosted on **Render**, and all requests from the frontend are configured to point to the Render server's URL.

Steps:

1. Run the Frontend:

Open a terminal in the frontend directory.

Install dependencies: **npm install**

Start the application: **npm start**

Wait for the Backend Server:

The Render server may take up to 5 minutes to wake up from idle mode. Once the server has fully started, it will function smoothly without delays. To confirm that the server is ready, click on the "Attendance Registration" and wait for the guide images to load. This indicates that the server is fully operational. Once the images are displayed, you can begin using the system, and it will work seamlessly.

Testing the Login Feature:

If you choose to test the **Login** functionality, you can use the following credentials of an existing user in the system:

- **Username:** barmalk1419@gmail.com
- **Password:** 123456

Enter these details to log in as a guide and explore the available options within the system.

8.2.3 Option 2: Running the Complete Application Locally (Frontend + Backend)

This option involves running both the **frontend** and **backend** locally. You will need to configure MongoDB, set up the backend server, and modify the frontend to point to the local backend.

Update the Connection String:

- Open the `server.js` file in the `backend` directory.
- Replace the placeholder connection string with your MongoDB connection string.

```
const mongoose = require('mongoose');

mongoose.connect('Your MongoDB Connection String Here', {
```

fig 33. The section of the code where you need to paste the Connection String.

· 8.3 Create a Database and Collection:

- In MongoDB Atlas:
 1. Click **Create Database** in the control panel.
 2. Enter the database name: CarmeyGil.
 3. Create a collection named: Guides.

Add an Initial Object:

- Add an object to the Guides collection with all necessary details.
- Once the object is added, you will be able to use its credentials to log in to the system and test the guide's functionality.

CarmyGil.Guids

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 112KB TOTAL DOCUMENTS: 3 INDEXES TOTAL SIZE: 72KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes **INSERT DOCUMENT**

Generate queries from natural language in Compass **Filter** Type a query: { field: 'value' } **Reset** **Apply** **Options**

QUERY RESULTS: 1-3 OF 3

1	_id: ObjectId('674dd2eabf0fc67da1825a07')	ObjectId
2	id: "2313393324"	String
3	name: "ניר גיל"	String
4	email: "barmalka1419@gmail.com"	String
5	password: "123456"	String
6	imageUrl: "https://scontent.ftlv21-1.fna.fbcdn.net/v/t39.30808-6/474534836_957586"	String
7	__v: 0	Int32

CANCEL **UPDATE**

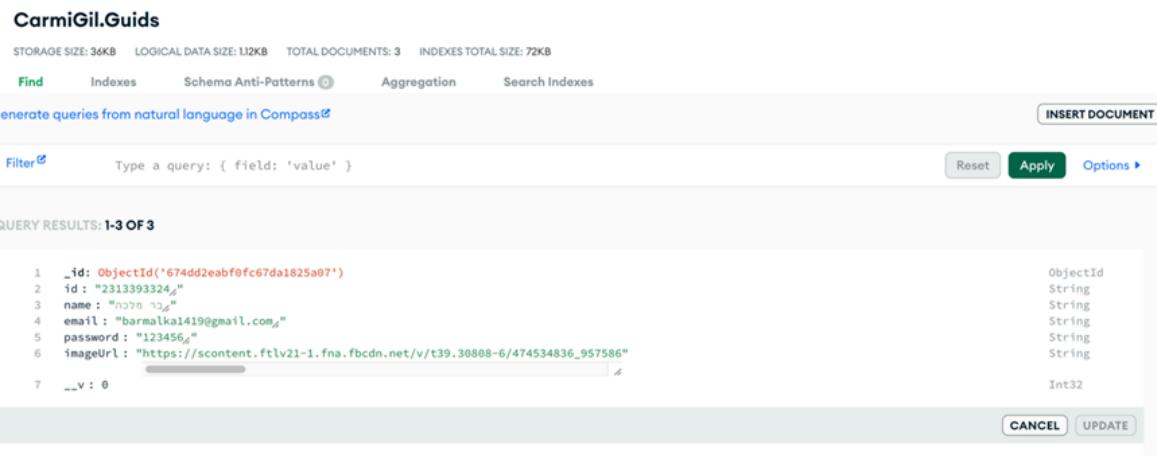


fig 34. Database view after creation.

Running the Backend Locally:

Open a terminal in the backend directory.

Install dependencies: `npm install`

Start the backend server: `npm start`

Updating Frontend to Point to Local Backend

If you choose to run the backend locally, you must update the frontend code to point to your local server.

1. Open the frontend codebase (typically in the `src` directory).
2. Search for all instances of the current backend URL (Render server URL).
3. Replace them with the local backend URL: <http://localhost:5000/>

Running the Frontend Locally

Open **another** terminal in the frontend directory.

Install dependencies: `npm install`

Start the React application: `npm start`

Access the Application.

Summary of Options:

- Option 1 (Frontend Only):
Use this option if you do not want to configure MongoDB or the backend locally. All requests will go to the Render server.

- Option 2 (Complete Local Setup):
Use this option for full local control, including running your own MongoDB instance and backend server. This requires updating the frontend to point to the local backend.

Explanation how to add an image link via Facebook:

This operation can only be done via computer. Log in to Facebook and select the picture of the user you want to add to the system. After you have selected the picture, hover over it and right-click. Then click "Open the image in a new tab" as shown in the example.

At the Carmey Gil Center, a dedicated Facebook account will be created, containing photos of guides and service recipients. This setup will allow the center to easily manage and update images. By using Facebook, the process of adding or updating images becomes simple and efficient, ensuring that all relevant pictures are accessible.

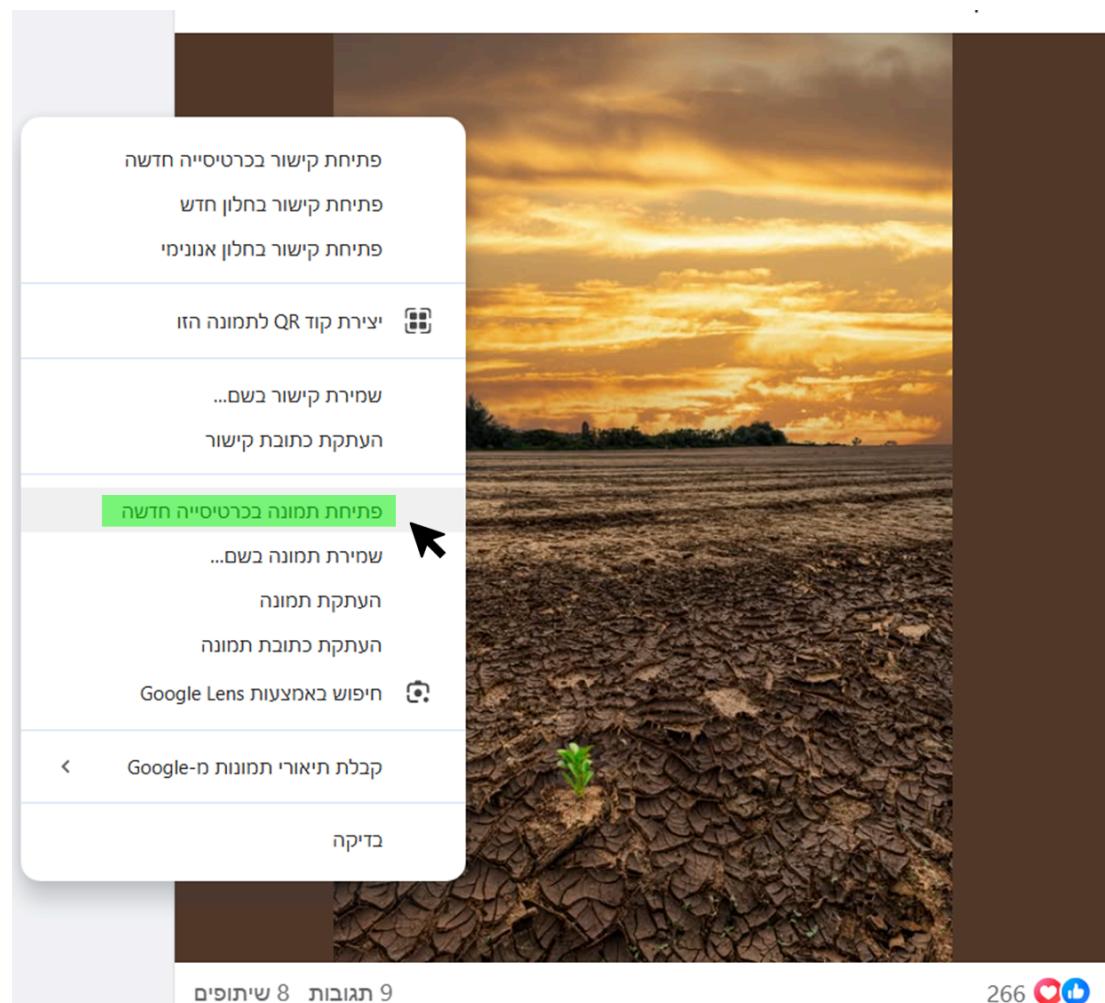


fig 35. Demonstration of adding a photo to the system from Facebook.

Now we will take the image URL from the new tab that opened for us.

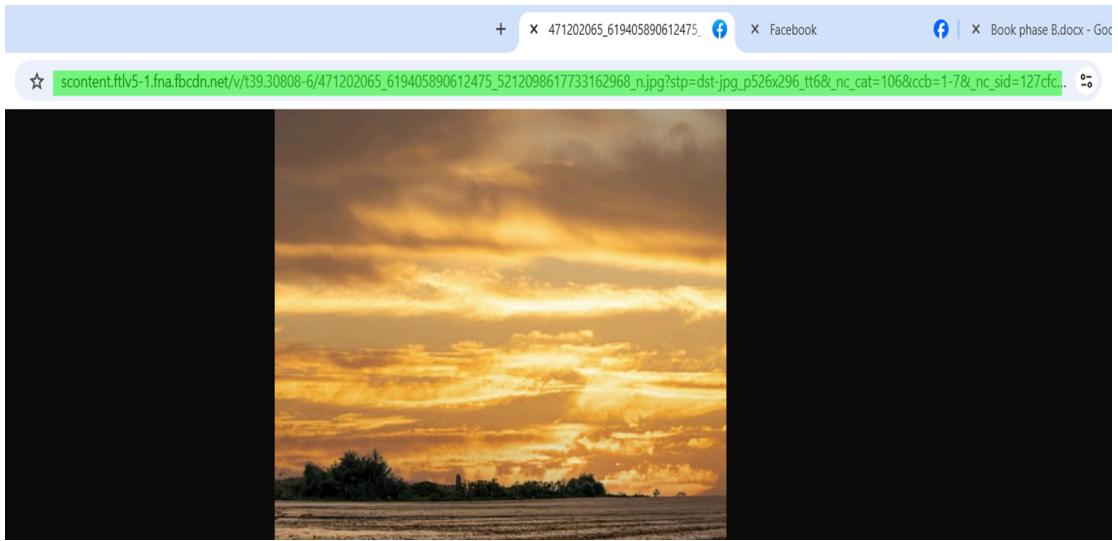


fig 36. Demonstration of copy url of the photo.

To create an initial object, paste the image address in the "imageUrl" field.

```
id : "3248376853"
name : "ראם נטנקל"
email : "barmalka141912@gmail.com"
password : "123456"
imageUrl : "https://scontent.fsdv1-2.fna.fbcdn.net/v/t1.6435-9/138147120_420210211..."
```



fig 37. Explain where in the database to paste the url link

To add an image as part of the process of adding a service recipient/guide, paste the link in the "profile photo" field on the Add Service Recipient/Guide page.

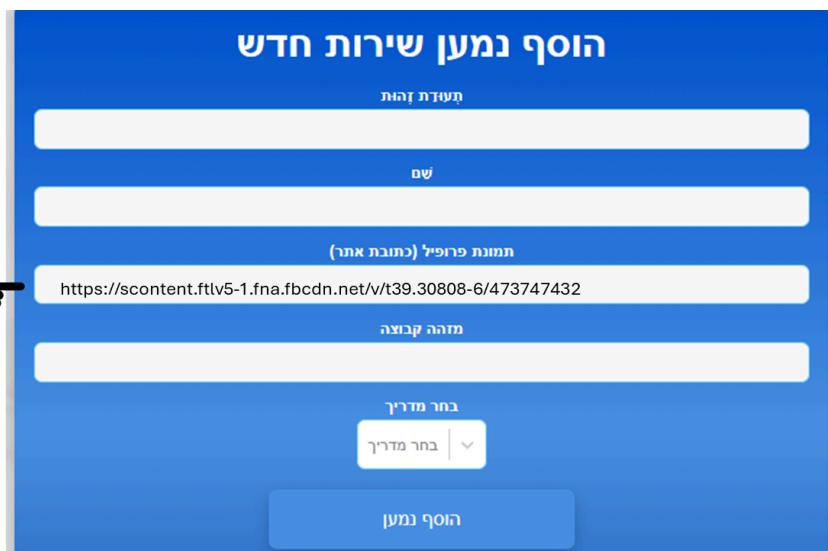


fig 38. Explain where in the system to paste the url link.

Explain how to set up an email for notifications:

Changing the Sender Email

To change the sender email (the email from which alerts are sent), follow these steps:

1. Replace the value in the **user field** – Update the email address under auth: { user: 'example@gmail.com' } to the new sender email.
2. Generate an App Password – If using a Gmail account, generate an App Password through the account security settings instead of using the regular password. Enter the new password in the **pass field**.

Changing the Recipient Email for Alerts

To change the email address that receives the alerts (defined in the **to field**), replace the existing email with the desired recipient's email.

Once updated, alerts will be sent to the new email whenever a user reports a negative emotional state.

The following code snippet is located in `backend/server.js`

```
// Email sending endpoint used in PatientSelectionPage
app.post('/api/sendEmail', (req, res) => {
  const { patientName, patientId, emotion } = req.body; // Extracting patient data and emotion from request body

  // Configuring the email transporter
  const transporter = nodemailer.createTransport({
    service: 'gmail', // Email service provider
    auth: {
      user: 'barmalka1419@gmail.com', // Sender email address
      pass: 'zlxxfjowctpmzfay', // App-specific password for Gmail
    },
  });
  // Email details
  const mailOptions = {
    from: 'barmalka1419@gmail.com', // Sender email
    to: 'barmalka1419@gmail.com', // report emotion mail
    subject: 'חוויות על מנת גמיש חשור',
    text: `ההודעה זו נשלחה בעקבות בחירת אינטוג', חזוך על ידי מטוטל
          שם: ${patientName}
         -ID: ${patientId}
          הרגש שנבחר: ${emotion}`}
```

fig 39. Explanation on Changing Email Addresses in the code

8.4 Deployment Diagram

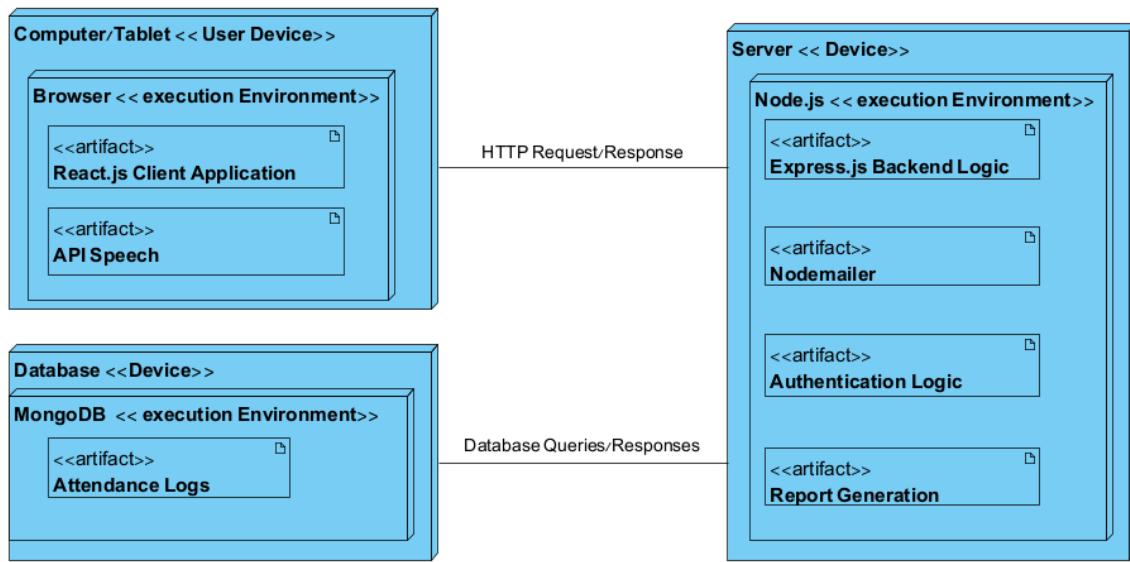


fig 40. Deployment diagram - Physical layout of the system

8.5 Database Description

Table Name: Guids

Fields: id, name, email, password, imageUrl

Table Description: In this table, we store the details of the guides at the Carmey Gil Center. This information is used during various operations within the system. The guide's image is saved using a URL link.

```
const mongoose = require('mongoose');

const guideSchema = new mongoose.Schema({
  id: { type: String, required: true },
  name: { type: String, required: true },
  email: { type: String, required: true, unique: true },
  password: { type: String, required: true },
  imageUrl: { type: String, required: true }
}, { collection: 'Guids' });

module.exports = mongoose.model('Guide', guideSchema);
```

fig 41. "Guids" Database table structure

```
id : "3248376853"
name : "ברנאליק ברמן"
email : "barmalka141912@gmail.com"
password : "123456"
imageUrl : "https://scontent.fsdv1-2.fna.fbcdn.net/v/t1.6435-9/138147120_420210211..."
```

fig 42. Example of guide data stored in the database

Table Name: patients

Fields: id, name, imageUrl, groupID, guideID, attendance

Table Description: In this table, we store the details of the service recipients at the Carmey Gil Center. This information is utilized during various operations within the system. Each service recipient has a field that stores the MongoDB ID of their assigned guide, which links them to a specific guide. When a guide is selected on the main page, the system displays all service recipients associated with that guide based on the stored MongoDB ID. The recipient's image is saved using a URL link.

```
const mongoose = require('mongoose');

const patientSchema = new mongoose.Schema({
    id: { type: String, required: true, unique: true }, // מזהה מותאם אישית ויחודי
    name: { type: String, required: true },
    imageUrl: { type: String, required: true },
    groupID: { type: String, required: true },
    guideId: { type: mongoose.Schema.Types.ObjectId, ref: 'Guide', required: true },
    attendance: [
        {
            date: { type: Date, default: Date.now },
            checkInTime: { type: String },
            checkOutTime: { type: String },
            score: { type: Number, default: null }, // שדה חדש לציון
        },
    ],
});

module.exports = mongoose.model('Patient', patientSchema);
```

fig 43. "patients" Database table structure

```
id : "31333933245"
name : "טולקן מורה"
imageUrl : "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRIwSpkIuM_VMj90W..."
groupID : "gorup1"
guideId : ObjectId('675c70240ddf88fcf1df184a')
attendance : Array (2)
```

fig 44. Example of patient data stored in the database

8.6 Main Code Sections

```
// Function to translate text to a specified target language using the Google Translate API
export async function translateText(text, targetLang = null) {
    // Determine the target language:
    // If `targetLang` is provided, use it.
    // Otherwise, check for a language stored in localStorage under the key 'selectedLanguage'.
    // If no language is set, default to Hebrew ('he').
    const language = targetLang || localStorage.getItem('selectedLanguage') || 'he';

    // Construct the URL for the Google Translate API request
    const url = `https://translate.googleapis.com/translate_a/single?client=gtx&sl=auto&tl=${language}&dt=t&q=${encodeURIComponent(
        text // Encode the text to ensure special characters do not break the URL
    )})`;

    try {
        // Send a GET request to the constructed URL
        const response = await fetch(url);

        // Parse the response into JSON format
        const data = await response.json();

        // Extract and return the translated text from the API response
        return data[0][0][0];
    }
}
```

fig 45. part of the code - translateText function

The code, from translation.js, is an asynchronous function that uses the Google Translate API to translate text into a user-specified language or defaults to Hebrew if none is provided. It dynamically constructs the request URL, sends a GET request, parses the response, and returns the translated text. This enhances accessibility by adapting content to the user's language preferences.

```

export function speakText(text, language = 'he') {
  const synth = window.speechSynthesis; // Access the browser's speech synthesis API. Web Speech API

  // Function to load available voices
  const loadVoices = () => {
    return new Promise((resolve) => {
      let voices = synth.getVoices(); // Retrieve the list of available voices.

      if (voices.length > 0) {
        resolve(voices); // If voices are already loaded, resolve the promise.
      } else {
        // Wait for the voices to be loaded, in case they're not yet available.
        const handleVoicesChanged = () => {
          voices = synth.getVoices(); // Reload the voices after the event triggers.
          resolve(voices); // Resolve the promise with the loaded voices.
          synth.onvoiceschanged = null; // Clear the event listener to prevent it from firing again.
        };

        synth.onvoiceschanged = handleVoicesChanged; // Set an event listener for when voices are loaded.

        // Backup: Call getVoices again after a short delay, in case the event doesn't trigger.
        setTimeout(() => {
          voices = synth.getVoices(); // Attempt to load voices again.
          if (voices.length > 0) {
            resolve(voices); // Resolve the promise if voices are found.
            synth.onvoiceschanged = null; // Clear the event listener.
          }
        }, 500); // Wait for 500 milliseconds.
      }
    });
  };
}

```

fig 46. part of the code - Loading Available Voices

The following two code segments are actually part of a single larger code that generates voice feedback in the system by reading aloud text. The code is presented here in two separate parts, each serving an important role.

The first part of the code is responsible for retrieving and loading the available voices for speech. The function `loadVoices` ensures that the voices are available before they are used. If the voices are already loaded, the function resolves immediately. Otherwise, an event listener is set on `synth.onvoiceschanged` to detect when voices are available. Additionally, a backup using `setTimeout` is implemented to retry loading voices after a short delay in case the event does not trigger automatically. This ensures that the system reliably retrieves available voices before proceeding with speech.

```

// Start the process of speaking the text
loadVoices().then((voices) => {
  const utterance = new SpeechSynthesisUtterance(text); // Create a new speech synthesis utterance for the text.

  // Set the language of the utterance based on the provided language code.
  if (language === 'ar') {
    utterance.lang = 'ar-SA'; // Arabic
  } else if (language === 'en') {
    utterance.lang = 'en-US'; // English
  } else if (language === 'ru') {
    utterance.lang = 'ru-RU'; // Russian
  } else {
    utterance.lang = 'he-IL'; // Hebrew (default)
  }

  // Configure the voice parameters for pitch, rate, and volume.
  utterance.rate = 0.95; // Set the speed of the speech.
  utterance.pitch = 1.1; // Set the pitch of the speech.
  utterance.volume = 1; // Set the volume of the speech.

  // Find a high-quality voice for the selected language, preferring "Google" voices.
  const voice = voices.find((v) => v.lang === utterance.lang && v.name.includes('Google'));
  if (voice) {
    utterance.voice = voice; // Assign the found voice to the utterance.
  }

  synth.speak(utterance); // Speak the utterance using the selected voice and settings.
}).catch((error) => {
  console.error('Error loading voices or speaking text:', error); // Log any errors encountered during the process.
});
}

```

fig 47. part of the code - Loading Available Voices

The second part of the code handles the actual text-to-speech conversion. Once the voices are loaded, the function speakText creates a SpeechSynthesisUtterance object containing the text to be spoken. The language is determined based on the provided parameter, supporting Hebrew (default), Arabic, English, and Russian. The utterance is then configured with specific rate, pitch, and volume settings to improve clarity. A high-quality voice is selected, preferring voices that include "Google" in their name for better performance. Finally, the synth.speak(utterance) function is called to vocalize the text.

```

const sendEmailWithAttachment = async (filePath, email) => {
  const transporter = nodemailer.createTransport({
    service: 'gmail', // Using Gmail service
    auth: {
      user: 'barmalka1419@gmail.com', // Sender email address
      pass: 'zlxxfjowctpmzfay', // App-specific password
    },
  });

  const mailOptions = {
    from: 'barmalka1419@gmail.com', // Sender email
    to: email,
    subject: 'הה נוכחות',
    text: 'מזרף כואן מודול זהה.',
    attachments: [
      {
        filename: 'attendance-report.xlsx',
        path: filePath,
      },
    ],
  };

  await transporter.sendMail(mailOptions); // Send the email
};

// Route to generate and send an attendance report
// Endpoint: /api/reports/send-report
// Used in AttendanceReport
router.post('/send-report', async (req, res) => {
  try {
    const { selectedMonth, email } = req.body; // Extract the selected month and email from the request.
    const wb = await generateExcelReport(selectedMonth);

    const filePath = `./attendance-report.xlsx`;
    XLSX.writeFile(wb, filePath); // Write the workbook to a file.
  }
});

```

fig 48. part of the code - Sending a report via mail

This code defines an API route for generating and sending attendance reports via email as part of the Carmey Gil Center's digital platform. The route accepts a request containing the selected month and an email address, generates an Excel report using the generateExcelReport function, saves it as a file named attendance-report.xlsx, and sends it as an email attachment to the specified address. This feature enables guides and administrators to receive organized attendance reports directly in their inbox, streamlining the process of managing attendance data and improving accessibility and efficiency.

```

// Email sending endpoint used in PatientSelectionPage
app.post('/api/sendEmail', (req, res) => {
  const { patientName, patientId, emotion } = req.body; // Extracting patient data and emotion from request body

  // Configuring the email transporter
  const transporter = nodemailer.createTransport({
    service: 'gmail', // Email service provider
    auth: {
      user: 'barmalka1419@gmail.com', // Sender email address
      pass: 'zlxxfjowctpmzfay', // App-specific password for Gmail
    },
  });
  // Email details
  const mailOptions = {
    from: 'barmalka1419@gmail.com', // Sender email
    to: 'barmalka1419@gmail.com', // report emotion mail
    subject: 'דווח על מצב רגשי חשוך',
    text: `הדווח זו נשלחה בעקבות בחירת אינזוג'. השוד על ידי המטולפ
      שם: ${patientName}
     -ID: ${patientId}
      הרגש שנבחר: ${emotion}
    `,
  };
  // Sending the email
  transporter.sendMail(mailOptions, (error, info) => {
    if (error) {
      console.log('Error sending email:', error);
      res.status(500).send('Error sending email');
    } else {
      console.log('Email sent:', info.response);
      res.send('Email sent successfully');
    }
  });
});

```

fig 49. part of the code from - Sending a mail about negative mood selection

The code defines an API endpoint that sends an email whenever a patient selects a suspicious emotional state. The endpoint is part of a system designed to monitor emotional states in the Carmey Gil Center. If a different email recipient is needed, the existing parameters should be replaced accordingly, and a valid token should be generated and set as the pass parameter to ensure proper authentication.

9. Conclusion and summary

In conclusion, our project provided the Carmey Gil Center with a practical and accessible solution to manage attendance and monitor emotional states more effectively. The system simplifies daily processes, giving service recipients greater independence while helping staff focus on care rather than administration.

Developing this system allowed us to create a tool that combines technology with thoughtful design, ensuring it meets the unique needs of the center. By implementing this platform, we have improved the way the center operates and positively impacted both staff and service recipients

9.1 Challenges and Solution

9.1.1 Free Translation and Voice Support

At the Carmey Gil Center, many service recipients face challenges with reading and understanding written text. To enhance accessibility, we integrated solutions that address these difficulties. We implemented both a translation service and text-to-speech functionality to help users confirm their choices during actions like selecting a guide or service recipient. Finding a solution that was accurate, seamlessly integrated with our system, and entirely free presented a significant challenge, as most available tools required payment or lacked essential features. We selected Google Translate for text translation, which is free, effective, and integrates well with our system. Additionally, we incorporated a speech API to provide voice feedback, allowing the system to read out text for users, further improving accessibility. Based on the language selected on the main screen, the system automatically translates all subsequent pages into the chosen language, ensuring a consistent multilingual experience. Combined with voice support, these features greatly enhance the usability of the system for service recipients.

9.1.2 Adjusting the System for Platform Transition

At the end of the first phase of the project, in which we planned the hardware together with the Carmey Gil staff, the original decision was to use a TV screen with touch functionality for user interactions. However, due to financial and technical constraints at the Carmey Gil center, we decided during development to switch the hardware to Tablets and computers. This change required us to adapt the system to ensure compatibility with both types of devices, allowing all actions to be performed seamlessly. For the tablets, we optimized the system for use with the built-in keyboard, while for computers, we ensured full functionality with external peripherals like keyboards and mice. These adjustments ensured that users could perform all tasks smoothly, whether on a tablet or a computer.

9.1.3 Deployment Services for the System

When it came to deploying the system and setting up all necessary services, including cloud hosting and servers, we faced challenges finding suitable free solutions. Most of the relevant options we encountered required payment or offered free trials for a limited time. Short-term free solutions were not suitable for our system, as we aim for long-term stability and continuous use by the Carmey Gil staff. This constraint posed a significant challenge, as we needed a reliable and sustainable solution to support the system's deployment and ensure its ongoing functionality.

9.1.4 Training the staff members

Training the staff at the Carmey Gil Center presented a challenge due to scheduling difficulties and varying levels of technical knowledge. We initially planned to train the manager, who would then train the staff, but it became clear during the session that some technical aspects were unclear to her. At her suggestion, a more tech-savvy guide joined the training, allowing for a deeper understanding of the system. To further support the process, we added the guide to the project's WhatsApp group to address technical questions and assist the manager in training the rest of the staff. This approach helped us overcome scheduling and knowledge gaps, ensuring effective training tailored to the staff needs.

10. References

1. Alshammari, M., Doody, O., & Richardson, I. (2020, August). Software Engineering Issues: An exploratory study into the development of Health Information Systems for people with Mild Intellectual and Developmental Disability. In 2020 IEEE First International Workshop on Requirements Engineering for Well-Being, Aging, and Health (REWBAH) (pp. 67-76). IEEE.
2. Borblik, J., Shabalina, O., Kultsova, M., Pidoprigora, A., & Romanenko, R. (2015, July). Assistive technology software for people with intellectual or development disabilities: Design of user interfaces for mobile applications. In 2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA) (pp. 1-6). IEEE.
3. Cabrera, R., Molina, A., Gómez, I., & García-Heras, J. (2017). Kinect as an access device for people with cerebral palsy: A preliminary study. International Journal of Human-Computer Studies, 108, 62-69.
4. Dekelver, J., Kultsova, M., Shabalina, O., Borblik, J., Pidoprigora, A., & Romanenko, R. (2015). Design of mobile applications for people with intellectual disabilities. Communications in Computer and Information Science, 535, 823-836
5. Galante, A., & Menezes, P. (2012). A gaze-based interaction system for people with cerebral palsy. Procedia Technology, 5, 895-902.
6. Huenerfauth, M., Feng, L., & Elhadad, N. (2009, October). Comparing evaluation techniques for text readability software for adults with intellectual disabilities. In Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility (pp. 3-10)
7. van Dam, K., Gielissen, M., Bles, R., van der Poel, A., & Boon, B. (2024). The impact of assistive living technology on perceived independence of people with a physical disability in executing daily activities: a systematic literature review. Disability and Rehabilitation: Assistive Technology, 19(4), 1262-1271.
8. Wilson, C., Sitbon, L., Brereton, M., Johnson, D., & Koplick, S. (2016, November). 'Put yourself in the picture' designing for futures with young adults with intellectual disability. In Proceedings of the 28th Australian Conference on computer-human interaction (pp. 271-281).
9. Zimmermann, G., & Vanderheiden, G. (2008). Accessible design and testing in the application development process: considerations for an integrated approach. Universal Access in the Information Society, 7, 117-128.

10. <https://react.dev/>
11. <https://www.mongodb.com/docs/>
12. <https://nodejs.org/en>
13. <https://expressjs.com>
14. <https://www.nodemailer.com/>
15. <https://chatgpt.com/>
16. <https://claude.ai/>

