Osim Seder - ARD

עושים סדר

Teammates:  
 Feliks Roizman  
 Amiel Saad

Yoav Avital

Ari Meir

Table of Contents

[Introduction 3](#_Toc157518701)

[About the costumer 3](#_Toc157518702)

[The Problem Domain 3](#_Toc157518703)

[Context 3](#_Toc157518704)

[Vision 4](#_Toc157518705)

[Stakeholders 4](#_Toc157518706)

[Software Context 5](#_Toc157518707)

[Usage Scenarios 6](#_Toc157518708)

[The Actors 6](#_Toc157518709)

[Use Cases 6](#_Toc157518710)

[Special usage considerations 19](#_Toc157518711)

[Functional requirements 22](#_Toc157518712)

[Non-functional requirements 25](#_Toc157518713)

[Implementation constraints](#_Toc157518714)

[Performance](#_Toc157518715) **25**

[Reliability and Stability](#_Toc157518716)

[Safety and Security](#_Toc157518717)

[Portability](#_Toc157518718)

[Usability](#_Toc157518719)

[Availability](#_Toc157518720)

[Platform constraints 27](#_Toc157518721)

[SE Project Constraints 27](#_Toc157518722)

[Special restrictions and limitations 3](#_Toc157518723)

[Proof of Concept 28](#_Toc157518724)

[Appendices 29](#_Toc157518725)

[I/O Format Information:](#_Toc157518726)

[User Surveys:](#_Toc157518727)

[Cost Analysis Studies:](#_Toc157518728)

[Background Information:](#_Toc157518729)

[Glossary 30](#_Toc157518730)

# Introduction

## About the costumer

The costumer we are working with is the organization "Be-Netivei Udi" (בנתיבי אודי) which was founded for the memory of Or Yehuda Elgrably, initiates many educational and social programs to strengthen and equalize the Israel youth connection. The organization acts among the different youth movements and schools and creates a community that has conversations and mutual assistance.

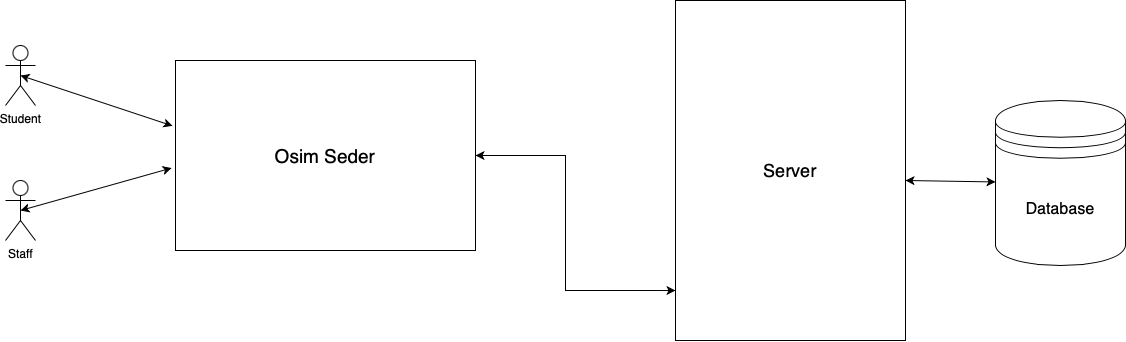
## The Problem Domain

The organization has one program called “Osim Seder” (עושים סדר) in which they gather pupils from schools and houses that need to be renovated in which elders that are known by the welfare body live in. In one of Passover days, all the signed pupils, with the organization staff, gather and go to the assigned houses (by groups) and renovating the houses. The program was embraced very well, and the number of houses and volunteers are growing each year.

In the last Passover, 1600 students helped 400 houses from Be’er Sheva and Jerusalem. The big increase in number tackled the organization with difficulties managing this complex operation, among them, keeping track of the progression of the renovation in the houses, communication with the students, discovery of shortage in supplies, correct assignment between a house that needs special attention.

## Context

The system operates within the context of an educational environment, serving both students and staff users. The primary purpose of the system is to facilitate information retrieval and data input for students, while allowing staff members to perform actions that involve accessing and manipulating a database.



## Vision

The goal of our software project is to establish a resilient and reliable system within the "Be-Netivei Udi" organization, while giving solutions to the needs of students and staff. We aim to ensure continuous operational efficiency, with a focus on security and minimal downtime and runtime. The system will feature user-friendly and convenient interfaces for both students and staff to operate on, prioritizing usability, security, and data integrity.

## Stakeholders

* **End Users:**
  + **Students:** Individuals using the system to access information and input their data.
  + **Staff:** Including team owners, area managers, and city managers who utilize the system for various administrative tasks and decision-making processes.
  + **Admin:** Generating data and managing the overview details of the program
* **Customers:**
  + **Shay Almog - Marketing Director of the Association (The Boss):** As the Marketing Director, Shay Almog represents a key customer with a high-level oversight, potentially influencing strategic decisions and overall project direction.
  + **Yarden Forman - Program Coordinator:** As the Program Coordinator, Yarden Forman is a customer with a specific role in program management. She is also the ‘city manager’ of Jerusalem.

## Software Context

The planned software system is designed to coordinate volunteer activities, focusing on improving living conditions for elders. Key features include mapping on-site needs, managing student information, assigning groups to houses based on capabilities and preferences, viewing required gear, reporting progress with visual documentation, command and control for administrators, and exporting student activity logs for schools.

**Inputs:**

1. On-site needs mapping data (elder information and house information needed).
2. Student information (school, accessibility, languages).
3. Custom capabilities and preferences for effective group assignment.
4. City/Area and their schools’ information respectively.

**Functionality:**

1. Mapping elder needs and house details.
2. Managing student information and group assignment.
3. Smart allocation considering special notices, both within houses and groups.
4. Providing a view of required gear for efficient task planning.
5. Reporting progress with visual documentation.
6. Admin oversight for comprehensive system monitoring.
7. Export information on the program schedule for the schools.

Our Relation to The Organization: We reached the organization through one of our team members that worked in one of the organization programs called “Nitzanim”. After we had a conversation with the program coordinator, we agreed on a website that includes efficient assignment algorithm of groups to different houses, realtime updates and progress report and notification on shortage of supplies. This solution would help them manage that operation, and maybe in the future it could evolve to other volunteering activities.

The Goal: The project goal is to create a convenient user interface so the students, volunteers and different managers in the organization can define tasks, track the progression of the students, refill stock of different supplies required for certain tasks, manage and control efficiently.

# Usage Scenarios

## The Actors

1. Admin: Can access all the information in the system and change it.

2. City Manager: In charge of certain city. Each city manager has a few area managers under him.

3. Area manager: In charge of ~2 schools from his area and on the houses that allocated with the students at these schools.

4. Team Owner: Communicates with the area manager, in charge of ~2 houses (2 groups), updates progress in real-time and refills shortage in gear.

5. Student: associated to a group with 2-3 more students.

## Use Cases

**Registration:**

* **Student:**

a. A student (without a user) enters the website "Osim Seder"

b. Clicks on "Register as Student".

c. Fills in his details: Name, Email, parent contact, Gender, City, School, Personal Requests, Disabilities, Languages

d. The system reads the information and registers the student in the system.

e. The system sends an email for the user for him to approve and notifies the user about it.

f. After the user approves, the user can access his account and make actions with it.

* **Staff: Team Owner/Area Manager/City Manager:**

a. Person (without a user) enters the website "Osim Seder".

b. Clicks on "Register as Staff".

c. The System presents the Staff member a new bar asking for a pre-defined code that is fixed to his role (decided by admin).

d. Fills in his details: Name, Email, City and Area (if relevant). There is also a role field that is unchangeable (defined by the previous code-word).

e. The system sends an email for the user for him to approve and notifies the user about it.

f. After the user is approved by email, the user is waiting for one of his superiors to accept his user and roles.

g. The system reads the information and registers the staff member in the system.

j. After the role is approved, the user is able to access his account and make actions with it.

* **Admin:**

a. Admin is a predefined account.

**Login**

* **All Roles:**

Precondition: user is logged out and registered in the website.

a. Person enters Email and Password and clicks on “login”.

b. If server authenticates the user:

1. Logged in.

Else:

1. an error message will appear to the user.

**Student Actions**

* **Creating a group:**

Precondition: user is logged in and not in a group.

1. User click on “My Group” tab
2. User clicks on “Create group”.
3. The system will open a new group with a unique ID that can be shared to other users.
4. The user is added to the new group.

* **Join existing group:**

Precondition: user is logged in and not in a group.

1. User click on “My Group” tab.
2. User inputs the group ID and “Join group”.
3. If group exists and not full:

user is added to the group.

Else:

a message error with be displayed

(wrong group ID / group is full)

d.  In case of success, the system will change the group view to the wanted group enlisting the other group members and (if assigned) the house details, tasks, and team owner.

* **Exit group:**

Precondition: user is logged in and in a group.

a. User clicks on “Leave group.”

b. The system will remove the user from the group.

1. If the group is empty:
   1. the system closes the group.

* **Get house info:**

Precondition: user is logged in and is in a group and a house is assigned to the group (after group-to-house allocation).

a. User clicks on “Group” tab.

c. The system will display the house information to the user:

1. Tasks for the house
2. Supplies
3. House address
4. Teammates names
5. Name and phone number of the elderly and Team Owner.

* **Edit personal details and preferences:**

Precondition: user is logged in and is a student.

a. User clicks on the “Person” icon in the header.

b. User can add/edit/remove his details besides email including:

1. Disabilities
2. Personal Requests
3. Languages

d. User clicks on “save” button to implement the changes in the system.

**Team Owner Actions**

* **View houses:**

Precondition: user is logged in and is a student.

1. User clicks on the “Houses” tab.
2. User selects the relevant house (of the houses he is owning).
3. The system will present the user with the following info:

* Tasks for the house, Supplies, House address, name and phone number of the elderly.
* For each task – status (completed or not).
* **View teams:**

Precondition: user is logged in and is a student.

a. User clicks on the “groups” tab.

b. User views the list of all the groups under his authority.

c. Each Group consist:  
 ID

Team Owner Name

Location (City/ City)  
 List of registered Students

d. Given the group has a house enlisted, he could direct himself through the house icon to get into the specific house

* **Manage team progress:**

Precondition: user is logged in, of type “Team Owner” and has at least one group assigned to him.

a. User clicks on the “My Houses” tab.

b. User selects the relevant house (of the houses he is owning), including the enlisted tasks.

c. The system will present the user with the following info:

1. Tasks for the house, Supplies, House address, name and phone number of the elderly.
2. For each task – status (completed or not).

d. If the team owner wants to change the status of a given task (if task is not done yet):

User clicks on the status color and changes it properly.

**Area Manager Actions**

* **House Management:**

Precondition: user is logged in, of type “Area Manager”.

1. User clicks on the “Houses” tab.
2. The system will view list of all the houses that relevant to the user (houses in assigned Area)
3. User selects the relevant house that he wants to access.
4. User enters the house info page, inside he can view the house basic information, the group and the group owner assigned, the house tasks, and supplies needed. Special notices including language preferences. Tasks Included.
5. The user can do the following actions:

Edit basic house information:

* 1. User clicks on the “Edit House” option inside the house page.
  2. The system will view the following information about the house: House address, name, gender and phone number of the elder, Supplies needed, and special notices including language preferences.
  3. The user clicks the relevant field he wants to change and inputs his change
  4. The user clicks on “Save Changes” to save the information in the system.

Add house task:

1. User clicks on the “Add Task” option inside the house page.
2. The system will view a page with the following fields: task type, room and free text.
3. The user will fill in the details and clicks “Create Task” to add the task.
4. The task then exports a list of needed supplies to the current task and adds it to the general supplies needed for the house.

Remove house task:

1. User clicks on the “Remove Task” option inside the house page on the relevant task.
2. The system will delete the selected task from the task list.
3. The house general supplies list updated accordingly

Edit house task:

1. User clicks on the specific task he wants to edit.
2. The system will view the following information about the task: task type, room, progress, before and after pictures, and free text.
3. The user clicks the relevant field he wants to change and inputs his change (such as description or pictures or progress).
4. The user clicks on “Save Changes” to save the information in the system.

* **Group Management:**

Precondition: user is logged in, of type “Area Manager”.

1. User clicks on the “Groups” tab.
2. The system will view a list of all the houses relevant to the user (groups in assigned Area).
3. User selects the relevant group that he wants to edit.
4. User enters the group info page, inside he can view the group basic information, The user can do the following actions:

Edit Group details:

Precondition: user is logged in, of type “Area Manager”.

1. User clicks on the “Group” tab.
2. The system will view the relevant groups that related to the Area.
3. User selects the relevant group he wants to change.

Add Group Member:

1. User clicks on the “Add Member” option.
2. User inputs the email address of the user he wants to add to the group

Remove Group Member:

1. User clicks on the “Remove Member” option on the user he wants to remove
2. the system removes the user from the chosen group

Assign Team Owner:

Precondition: user is logged in, of type “Area Manager”.

1. User clicks on the “Groups” tab.
2. The system will view the relevant groups that related to the Area.
3. User selects the relevant group he wants to assign a team owner to.
4. The system will view the selected group page.
5. User clicks on the “Assign Team Owner” option.
6. User inputs the email address of the member he want to assign as a team owner.
7. if: the requested member doesn’t exist or doesn't have the credentials of a team owner, an error message will be displayed.
8. else: the requested member added as a team owner to the selected group.

View Students Special Comments:

Precondition: user is logged in, of type “Area Manager”.

1. User clicks on “Groups” tab.
2. The system will display all the groups that are under the responsibility of the user.
3. User clicks on a group with the user he wants to view.
4. User clicks on the name of the student he wants to see his special comments.
5. The system displays the profile page of the student.
6. The user can see the student’s special comments.

* **Allocation Management:**

Precondition: user is logged in, of type “Area Manager”.

1. User clicks on the “Groups” tab.
2. The system will view a list of all the houses relevant to the user (groups in assigned Area).
3. User selects the relevant group that he wants to edit.
4. User enters the group info page, inside he can view the group basic information, The user can do the following actions:

Change group-house allocation:

Precondition: user is logged in, of type “Area Manager”.

1. User clicks on the “Allocation” tab.
2. The system will view a page with pairs of group-house (each group with a house under the group responsibility)
3. The user can change the allocation of two groups by selecting two pairs of group-house and clicking on the button “Switch”.
4. Note: he can also replace pairs with only 1 side (in case there are extra houses \ groups)

**City Manager Actions**

* **House Management:**

Precondition: user is logged in, of type “City Manager”.

1. User clicks on the “Houses” tab.
2. The system views a page with all the houses that are relevant to the user City.

Add House:

1. User clicks on the “Add House” button.
2. The system prompts a page with a list of the following fields that need to be filled: elderly person’s name, address, phone number, special notices, language preferences and gender identity.
3. User fills all the relevant details and clicks on “Create House”.

Edit House/Add Task/ Edit Task / Remove Task:

same as Area manager.

* **Group Management:**

Edit Group Details/ Add Group Member / Remove Group Member/ view student special notices:

same as Area manager.

* Exporting School Information:

Precondition: user is logged in, of type “Area Manager”.

1. User clicks on “Export” tab.
2. The system will view all the relevant data under the responsibility of the user.
3. User clicks on “Export School Information”.
4. The system will display a page with a list of schools that need to be selected from.
5. The user choose the school he wants to export information about
6. The system will create an excel document with the following information about a school: student name, email, address of the house he will volunteer at, team owner and neighborhood manager.

* **Allocation Management:**

same as Area manager.

* **Accesses Management:**

Precondition: user is logged in, of type “City Manager”.

1. User clicks on the “Accesses” tab.
2. The system will display all the requests for a role and all the staff members in the system under his authority .

Access Request:

1. User can click on “Approve” or “Decline” option under the user request

Edit Access:

1. User chooses a staff member he wants to change his role
2. User clicks on “Edit Access”.
3. The system displays a window with a dropdown menu that the user can choose a new access to give to the user.

Remove Access:

1. User clicks on “Remove Access” on the user he wants to remove his access.

**Admin Actions**

* **Admin Management:**

Precondition: user is logged in, of type “Admin”.

1. User clicks on the “Admin” tab.
2. The system will display an admin panel with specific admin options.

Limit Final Date For Registration:

1. User clicks on “Change Final Date”.
2. The system will display a calendar.
3. User chooses a valid date from the calendar.

Run Group-House Allocation:

1. User clicks on “Run Allocation”
2. The system will run an group-house allocation algorithm that allocates each house with a group
3. User can view the allocation under the “Allocation” tab.

Publish Group-House Allocation Results:

1. User clicks on “Publish Allocation”.
2. The system will publish the allocation results so all users with the appropriate authority can view the allocation

Basic System Information:

1. User clicks on “System Information”.
2. The system will display a page with the following sections: Citys, Areas.

Add New City/Area/School:

1. User clicks on “Add” in the Citys/Areas/school section.
2. The system will ask an input for the name of the new City/Area/school.
3. User inputs the desired name and clicks “Create”.

Edit Existing City/Area/School:

1. User clicks on “Edit” in the Citys/Areas/school section on the desired City/Area/school he wants to edit.
2. The system will ask an input for a new name of the selected zone.
3. User inputs the desired name and clicks “Save Changes”

Remove Existing City/Area/School:

1. User clicks on “Remove” in the Citys/Areas/schools section on the desired City/Area/school he wants to remove.
2. The system will remove the City/Area/school the user selected.

* **Accesses/Groups/House/Allocation Management:**

same as City manager.

## Special usage considerations

* **Inactivity of the user:**

Precondition: User is logged in

1. If the user is inactive for X minutes (X = 30 minutes) then the user is automatically logged out from the system.

* **Multiple logins from the same user:**

Precondition: User is logged in on computer A

* 1. If the user is trying to login on computer B, the system will log out the user on computer A and login the user on computer B.
* **Database system is down:**

Precondition: Database is running

* 1. If there is a power outage that causes the database to shut down, the system will:
     1. Option A: save all changes locally for each user logged in (for all users BUT students) and will update the database once the server and database are running. Students will be able to view their caches data as if nothing happened.
     2. Option B: all users will get a message that the database is down and the system will not be operational.
* **Server system is down:**

Precondition: Server is running

* 1. If there is a power outage that causes the Server to shut down, the system will:

1. Option A: display a 404 page for every user.

* **Notifications:**

Preconditions: all teams are allocated

* 1. Student leaving group:
     1. A notification will appear to the team manager of the student’s team.
  2. Team finished an assignment:
     1. A notification will appear to the team manager about the completed task.

# Functional requirements

**General**

* The system must allow student registration and login. The student has a unique identifier that identifies him in the system.
* The system must allow staff registration and login. Each staff member receives a pre-defined code to which they include in their registration request so a higher-authority role will approve. The staff has a unique identifier that identifies him in the system.
* The system will allow any type of user to be exposed only to information that is relevant to the city in which he is associated.
* The system must provide an option to export a summary of all the equipment required for a specific house.
* The system must provide an option to export a summary of all the equipment required for an Area/City/Day rule.
* The system must give the elderly person an option to approve publishing photos of his house. (NTH)
* The system will allow students to join and leave groups.
* The System will include in every search bar a filter option, in which different roles with many objects to sort will be able to view specific Cities/schools.
* Details of the house will contain address, city, supplies needed and the name, gender and phone number of the elder.
* In case of the server shutting down for maintenance or random server shut down, a 404 page will appear to the user stating the server is down at the moment, advising to try again later.
* In case of the data base shutting down, when the user tries to use one of CRUD operations the system will display a message apologizing for the inconvenient, advising try again later or contact an admin.
* In case of the user staying idle and not doing any operation in the system, the system will pop up a message saying the user is idle for too long and will force log out.
* The system will logout the user from device ‘A’ if the user from device ‘B’ already logged in.
* The admin will be a single user.

**Student**

* Student registration will ask for: name, email, school, parent contact and email, accessibility issues, gender, mastery of additional languages, personal preferences, and limitations.
* The system will validate a student user, only after the parent of the user will accept the form that was sent to him by email (that was provided by the student in the registration form).
* The system will allow a student who is not in the group to join the group, by selecting a group from the groups list.
* The system will allow a student who is in a group to leave the group.
* The system will allow the student to view the details of the house to which he is associated, the student's group, the Team Owner and the tasks assigned for the house.

**Team Owner**

* The system will allow a Team Owner to view the details of the houses and students assigned to him.
* The system will enable a Team Owner to view the details of the groups assigned to him.
* The system will allow a team owner to view the progress (status) of the houses associated with him.
* The system will allow a team owner to update the progress of the groups associated with him. Update contains: changing status, and taking an "after" photo and changing the status on specific tasks.
* The system will grant an area manager access to a map with pinpoints for each home listed on the renovation day. (NTH)

**Area Manager**

* The system will allow an Area Manager to edit the details of homes.
* The system will enable an Area Manager to edit a group, including adding and removing from it.
* The system will allow an area manager user to assign a team owner to a home.
* The system will provide an Area Manager with access to the school’s responsible person’s phone number (for issues/questions regarding students)
* The system will allow Area manager user to change the current group-house allocation in his Area
* The system will enable an Area Manager to view special notices entered by students and houses under their supervision.
* The system will grant an area manager access to a map with pinpoints for each home listed on the renovation day. (NTH)

**City Manager**

* The system will allow a city manager to add a home to the system, including: the elderly person’s name, address, phone number, the home needs with location, specific tasks and necessary tools, accessibility information if applicable, language preferences and gender identity.
* The system will enable a City manager to export info about a specific school students group and send it to their representative (excel doc, with list of all students -> student name, email, address of the house he will volunteer in, team owner, Area manager).
* The system will enable a city manager to edit home details.
* The system will allow a city manager to edit groups’ details such as removing students and joining students to a group.
* The system will enable a city manager user to view special comments entered by the students and houses under their supervision.
* The system will allow a city manager will be able to approve requests for different roles under his authority (Team Owner/Area Manager)
* The system will allow a city manager to edit or remove roles from staff members under his authority (Team Owner/Area Manager)
* The system will grant a city manager access to a map with pinpoints for each home listed on the renovation day. (NTH)

**Administrator**

* The system will allow an administrator to access all the information and operations of other roles, including houses, students’ elderly individuals, groups, allocations, tasks, required equipment.
* The system administrator will be able to approve requests for different roles
* The system administrator will be able to edit or remove roles from staff members
* The system administrator will be able to add and edit a limit date for which students can alter their groups until final allocations are to be made.
* The system administrator will be able to run and publish the group-house allocation algorithm.
* The system administrator will be able to add/edit/remove basic system information like: Cities, Areas. (NTH)

# 

# Non-functional requirements

## Implementation constraints

### Performance

#### Speed

The system will aim to process and respond to user interactions, such as creating a new task and progress reporting, within 2 seconds for 95% of transactions.

#### Capacity

The infrastructure must support the increasing demand, considering the annual growth in student participation. The system should be scalable to support a 15% increase in users over the next year without compromising performance.

#### Throughput

The system will be designed to handle a minimum of 2000 simultaneous users without significant degradation in performance during peak usage periods, ensuring efficient task coordination.

### Reliability and Stability

* In case of the server shutting down for maintenance or random server shut down, a 404 page will appear to the user stating the server is down at the moment, advising to try again later.
* In case of the data base shutting down, when the user tries to use one of CRUD operations the system will display a message apologizing for the inconvenient, advising try again later or contact an admin.

### Safety and Security

#### Confidentiality

User data, including elder information and student details, shall be encrypted during transmission and storage to ensure confidentiality.

#### Access Control

Different user roles, such as students, team owners, and administrators, shall be implemented with varying levels of authority to ensure secure access control. This prevents unauthorized users from accessing sensitive information.

#### Secure Communication

The system shall utilize secure communication protocols to safeguard the exchange of information between users and the server (i.e. https requests), enhancing overall system safety.

### Portability

#### Browser

#### The system shall be designed to function independently of specific browsers, ensuring compatibility with major browsers using React generated web pages. This guarantees a consistent user experience across different browsers.

#### Devices

The system shall be designed to function on various interfaces such as phones, computers etc.

### Usability

#### Students

The program is designed for effortless use by students, requiring no training. The interface is intuitive and straightforward, ensuring that students can easily comprehend and navigate the system. Additionally, student actions are intentionally limited for simplicity.

#### Staff

While staff members already engaged in the program may need a quick introduction to new features spread across different web pages, the overall interface remains user-friendly and easy to grasp. This ensures a smooth transition for staff to adopt and utilize the extended functionalities without significant training.

### Availability

#### The system is expected to maintain a high level of availability, with a minimum uptime of 99.5% during all hours, given the low interactions typically required. During the main event (Passover), the system should remain highly usable, providing a seamless experience comparable to a regular day, ensuring that users can engage with the platform without disruption.

## Platform constraints

The project is developed using Node.js and React for both the front and backends, with a relational database for data storage. Given this technology stack, there are no platform constraints imposed, providing flexibility and compatibility across various operating systems and platforms. The choice of Node.js and React ensures a versatile and platform-agnostic development environment, allowing seamless deployment and operation on different platforms without limitations.

Task status will be in 3 colors: red–not started, yellow–on the way, green–done.

## SE Project Constraints

* The program is interactive, and the inputs will come from the users – staff members and students. Students will only input personal data and staff will input information about houses, elders, location etc.
* Some information will be stored in the database as a “blob” of information – for example: areas in cities and the cities themselves.
* No special hardware is required to run the program.
* New staff members will have to get a secret code from the admin (not via the system itself) to register as a new staff member.
* The final system will be presented as a web application (hopefully with some statistics from this year’s Passover operation).

## Special restrictions and limitations

There are none.

# Proof of Concept

**Risks assessment:**

1. We need 100% coverage of testing, that includes unit, Integration, functional, end to end, security, load, and user Acceptance tests.
2. We need an online server and database which requires funds for this project.
3. We need to ensure that our Program is safe from hackers, so data like passwords can't be stolen
4. We need a logger to the system so we can keep track of functionality that is happening inside our program
5. Verify the web app's compatibility across different web browsers (chrome, Firefox, Safari, Edge, etc) addressing any functionality to ensure consistence experience for users, regardless of their browser preferences.
6. On our project time is crucial, as we need to provide the first working product on April.
7. Ensure that the user interface adapts seamlessly to various resolutions and orientations.
8. We need to insure the server is always up and doesn't crash, so with the database.

**Approach to risks:**

1. All the tests can be obtained by testing libraries and other programs for more advanced testing like acceptance tests and load tests. Also,we will ask our clients to help us test the program through a user point of view to maximize the user experience with our program and insure that our GUI is user friendly and enhance overall experience (UX testing).
2. We spoke to our marketing director, and he is on board to support us financially, so we are in the clear to keep developing.
3. In terms of security, we will be using Node.JS libraries to secure sensitive data we'll also ensure our HTTP requests are on the side using standard security libraries.
4. For real time insights into the system performance, anomaly detection and issues resolution, we will be implementing a logging mechanism so we can detect any kind of problem at any state of the program.
5. We will Test our program in all kinds of browsers and if there will be any issues we will handle them manually.
6. In our project, we are basically following the usual path of using client-side, server side, and database so all of the development team are familiar with the project model which will save time. Also since the beginning of the semester we have all started learning Node JS on Udemy and a few members of our team already built projects in React and node JS.
7. Responsiveness can be done with unique CSS coding. And when importing components make sure they are adaptable to any resolution.
8. . In our development, we will make our best effort so the server doesn't crash, but server crash can be unpredictable, so is database crash. If the server crashes, we will show a 404 page saying that the server is down. And if the database is down, we will pop up a message that we are sorry for the inconvenient and that the database is down at the moment.

# Appendices

## I/O Format Information:

The application will primarily use JSON (JavaScript Object Notation) as the standard data interchange format. This format will be utilized for communication between the client and server, ensuring a structured and efficient data exchange. Also, using this technique we ensure that only the information we need is presented to the user.

## User Surveys:

Continuous collaboration with the association's staff is an integral part of the development process. Regular meetings and discussions are held to gather feedback, implement requested features, and make informed design decisions.

## Cost Analysis Studies:

The majority of development tools and resources used are either open-source or offer free versions that meet project requirements. In cases where additional funding is needed, the Marketing Director of the Association has committed to providing financial support for future needs.

## Background Information:

The application is tailored for school students as end-users, providing a user-friendly and engaging experience. The staff members associated with the Association, comprising adults, act as administrators and facilitators in the system.

## Glossary

* **Admin** - a user created during system activation. Admin has permission to view all information and can create users with specific roles.
* **Area Manager** - A user created by the admin, or by the City manager, Responsible for ensuring the availability of required equipment and supervising core students and work status in assigned homes**.**
* **City Manager** - a user created by the admin, responsible for creating homes and overseeing the activities of city’s coordinators**.**
* **Elderly individual** - External player: students and core students go to elderly individuals to perform tasks.
* **Equipment**- Equipment required for task execution.
* Group - A group of 3-4 students, all under one home. Each group has one core student supervising them.
* **Home/Elderly Home** - Home represents the need for a “group”. Each home has a group assigned to perform required tasks.
* **School** - External player: schools can receive details from the city manager or admin regarding students associated with the school.
* **Special notices** - Every elder house \ student will have an option to write special notices that will include languages he speaks (for an elder that knows only 1 language that’s not Hebrew), disabilities, and other things that he thinks should be considered.
* **Student** - A user created through website registration. When created, the student has the option to assign to a group.
* **Task** - A specific need in a particular home. Each task generates a requirement for specific equipment displayed to core students and core coordinators.