Final project Phase B

Paz Fayer, Eden Bar

Supervisor: Dr. Naomi Unkelos Shpigel

September 22, 2024

Project code: 24-1-D-41

IIoT Inclusive Design Management Software

Figure 1: logo

Software Engineering Department Braude College of Engineering

Contents

1 Introduction **4**

2 Background and Related Work **4**
2.1 Manpower Shift Scheduling Problem . . . 4
2.2 Inclusive Design . . . 5
2.2.1 User Centered Design . . . 5
2.3 Industrial Internet of Things (IIoT) and MQTT Protocol . . . 6
2.4 Genetic Algorithm (GA) . . . 6
2.5 Client-Side Technologies . . . 6
2.5.1 React Framework . . . 6
2.5.2 JavaScript . . . 6
2.5.3 TypeScript . . . 7
2.5.4 Cascading Style Sheets . . . 7
2.5.5 Hypertext Markup Language . . . 7
2.6 Server-Side Technologies . . . 7
2.6.1 Node js . . . 7
2.6.2 MongoDB . . . 7
2.6.3 AWS . . . 7

3 Software engineering process **8**
3.1 Genetic Algorithm . . . 9
3.1.1 Genetic Algorithm (GA) . . . 9
3.1.2 Genetic algorithm in our system . . . 9
3.1.3 Genetic algorithm flow . . . 9
3.1.4 Using Algorithm Output . . . 11
3.2 Interviews and meetings with 'Migdal-Or' Managers . . . 11
3.3 Requirements . . . 12
3.3.1 Non-Functional Requirements (NFR) . . . 12
3.3.2 Functional Requirements (FR) . . . 12
3.4 Use Case Diagram . . . 13
3.5 Activity Diagram . . . 14
3.6 Class Diagram . . . 15
3.6.1 Model . . . 16
3.6.2 View . . . 17
3.6.3 Controller . . . 18

4 User Manual **19**
4.0.1 The application . . . 19

5 Developer Manual **27**
5.1 Product Design . . . 27
5.1.1 System Architecture . . . 27
5.1.2 Database Schema . . . 29
5.2 Backend . . . 29
5.2.1 Directory Structure . . . 29
5.2.2 Key Components . . . 30
5.3 Frontend . . . 30
5.3.1 Frontend directory . . . 30
5.3.2 Key Components . . . 30
5.4 Deployment Process . . . 30
5.4.1 Deployment . . . 30
5.5 Testing Plan . . . 31

6 Achievements **32**
6.1 Success Criteria . . . 32

7 Users Evaluation **32**

8 Future Work **33**

References **34**

9 AI Tools and Prompts **36**

10 Appendices **36**
10.1 First Interview with Gabi ('Migdal-Or' manager) 36
10.2 Second Interview with Gabi ('Migdal-Or' manager) 36
10.3 Third Interview with Gabi and Shmuel(department manager) 38

Abstract

This project addresses the Manpower Shift Scheduling Problem of Migdal-Or, a factory employing visually impaired workers, and is part of the flagship project at the college with Mechanical engineering faculty. The primary objective was to transition from manual management methods to an advanced technological solution that optimizes worker allocation and enhances operational efficiency. A genetic algorithm was developed for worker placement optimization, integrated with Industrial Internet of Things (IIoT) technology for real-time productivity data collection. Custom software was created for data analysis and report generation, focusing on inclusive design principles to accommodate the unique needs of visually impaired employees. The project demonstrates how modern industrial technologies, including IIoT, can be adapted to address specific challenges in specialized manufacturing environments, particularly for workers with visual impairments. By automating and optimizing productivity and worker placement, the system aims to promote dignity, independence, and economic inclusion for visually impaired workers. This approach not only improves operational processes but also fosters a supportive work environment, showcasing the potential of inclusive design in contemporary industrial settings. The findings contribute to a broader understanding of how advanced scheduling and management systems, enhanced by IIoT capabilities, can be leveraged to promote workplace inclusion and efficiency in specialized manufacturing contexts.

1 Introduction

"Migdal-Or" is a factory focusing on inclusive population that offers employment to around 120 individuals with visual impairments. Operating under business principles that emphasize quality and efficiency, the factory empowers these individuals to contribute meaningfully to society. More than just assembling products, Migdal-Or provides valuable vocational training and social connection, fostering a supportive environment where employees can thrive. This unique model promotes dignity, independence, and economic integration, allowing individuals with visual impairments to shine both at work and in their lives. Today, most of the information about workers output and assigning process in positions managed manually in Excel files. This literature review aims to provide a comprehensive understanding of IIoT's contribution to solving the "Migdal-Or" problem by analyzing relevant academic research and industry publications.(https://www.migdalor.org.il/wpcontent/uploads/2024/03 n.d.)

2 Background and Related Work

2.1 Manpower Shift Scheduling Problem

The manpower shift scheduling problem (MSSP) is a complex optimization problem encountered in various industries. It involves assigning workers to tasks while considering multiple factors to achieve an optimal outcome. These factors can include worker skills and experience, task requirements, workload balancing, and minimizing costs or maximizing productivity (Brucker, Burke, Curtois, Qu, & Vanden Berghe, 2010). MSSP is an NP-HARD problem, which means that we cannot find the most effective solution in polynomial time. (Liu & Zhang, 2021). Traditional methods for solving MSSP often rely on mathematical programming techniques like linear programming or integer linear programming. These approaches seek to optimize an objective function, such as total labor cost, by formulating constraints as equations and inequalities (Ravindran, 2007). However, these methods have limitations. They can struggle with large-scale scheduling problems with many complex real-world constraints, leading to computationally expensive solutions that might not be optimal (Van den Bergh, Beliën, De Bruecker, Demeulemeester, & De Boeck, 2013). Recent research explores alternative approaches to MSSP. Heuristics and metaheuristics offer efficient solutions for finding near-optimal placements, particularly for larger or more intricate problems (Blum & Roli, 2003). These methods involve iterative search algorithms that explore potential solutions and converge towards high-quality assignments. Additionally, machine learning techniques are gaining traction for MSSP. Supervised learning algorithms can be trained on historical data to predict worker performance on specific tasks, aiding in more informed placement decisions. Genetic algorithms (GAs) have emerged as a powerful alternative for tackling MSSP. GAs mimic the process of natural selection, where a population of candidate solutions (schedules) evolves over generations. Each solution is represented by a chromosome, often a binary string encoding employee assignments to shifts. In the context of MSSP, the chromosome might define which employees are assigned to morning, afternoon, or night shifts (Nearchou & Lagodimos, 2013). The GA iteratively performs the following steps:

- Selection: Fitter solutions (schedules with fewer violations or lower costs) are chosen for reproduction.
- Crossover: Selected solutions are combined to generate new offspring (new schedules) by exchanging parts of their chromosomes.
- Mutation: Random changes are introduced in the offspring to introduce diversity and explore new solution spaces.
- Evaluation: The fitness of the new offspring is evaluated based on the objective function and constraints.

This process continues for a predefined number of generations, aiming to converge on a near-optimal solution that satisfies all constraints and achieves the desired objective. GAs offer several advantages for MSSP. They can handle complex constraints effectively and are well-suited for large-scale problems. Additionally, GAs are stochastic, meaning they can escape local optima and explore different solution spaces, potentially leading to better results compared to deterministic methods. (Voß, 2001). However, GAs also have limitations. They require careful tuning of parameters like population size, crossover rate, and mutation rate to achieve optimal results. Additionally, they might be computationally expensive for very large problems with numerous employees and shifts. Despite their limitations, GAs remain a valuable tool for tackling the MSSP. Research continues to explore advanced GA techniques like hybrid approaches that combine GAs with other optimization methods for improved efficiency and solution quality (Burke et al., 2013).

2.2 Inclusive Design

The digitalization of our world has significantly impacted everyday life, shaping public and commercial services. However, despite its advantages, a substantial portion of the population remains excluded from the digital realm (Rodríguez-Pérez, Nadri, & Nagappan, 2021).Inclusive design, which aims to develop services and products that cater to diverse backgrounds and abilities, plays a crucial role in fostering digital inclusion. In particular, it accommodates the needs of users with varying abilities, ensuring equitable access to technology. A recent review by Li, Li, and Tang (2023) explored the field of design for digital inclusion. Their study analyzed 721 relevant articles from Scopus, revealing four main topics: information technology, online education, assistive technology, and digital health. The review also highlighted the distinctive features of design for digital inclusion compared to inclusive design and identified research gaps and future directions (Li, Li, & Tang, 2023). Visually impaired individuals face unique challenges when interacting with digital interfaces. Inclusive design principles play a pivotal role in ensuring equitable access to software applications for this user group. By considering their needs, software developers can create more accessible and user-friendly systems. A comprehensive approach involves addressing aspects such as screen reader compatibility, keyboard navigation, and alternative text for images. Researches emphasizes the importance of designing software interfaces that accommodate various visual impairments, including low vision and blindness. Their work highlights the significance of collaboration between designers, developers, and visually impaired users to create inclusive software experiences (Johnson & Abumeeiz, 2023). As we strive for a more accessible digital landscape, integrating these principles becomes essential for promoting equal participation and usability for all.

2.2.1 User Centered Design

User-Centered Design (UCD) is a crucial component of inclusive design, placing users at the heart of the development process. UCD involves understanding and addressing user needs, preferences, and limitations throughout the design lifecycle (Norman, 2013). This iterative approach encompasses early user research, empirical measurement of product usage, and continuous refinement based on user feedback (Abras, Maloney-Krichmar, & Preece, 2004). By employing methods such as interviews, surveys, and usability testing, UCD ensures that products are not only functional but also intuitive and satisfying for a diverse range of users (Vredenburg, Mao, Smith, & Carey, 2002). In the context of inclusive design, UCD helps identify and address potential barriers to accessibility, usability, and user experience for individuals with varying abilities and backgrounds. This user-focused approach ultimately leads to products that better serve a wider audience, aligning closely with the goals of inclusive design.

2.3 Industrial Internet of Things (IIoT) and MQTT Protocol

The Industrial Internet of Things (IIoT) and Message Queuing Telemetry Transport (MQTT) are transforming modern industrial landscapes. IIoT facilitates real-time data exchange, analysis, and decision-making across diverse industries through a three-layer architecture: connectivity, intelligence, and application (Lee & Lee, 2015). This architecture enables applications such as predictive maintenance, real-time monitoring, and process optimization in sectors like manufacturing, energy, and healthcare (Boyes, Hallaq, Cunningham, & Watson, 2018). Complementing IIoT, serves as a dominant messaging protocol for IoT, featuring a lightweight design and publish-subscribe model that enables efficient data exchange on resource-constrained devices (Quincozes, Emilio, & Kazienko, 2019). MQTT's scalability allows it to manage large numbers of devices and messages, while its topic-based messaging facilitates efficient data categorization and targeted distribution (*AWS IoT Core for MQTT*, n.d.). Despite the numerous benefits, IIoT faces challenges in security, interoperability, and scalability (Mineraud, Mazhelis, Su, & Tarkoma, 2016). MQTT addresses some of these concerns with its support for encryption and different Quality of Service (QoS) levels, ensuring reliable message delivery even in unstable network conditions (*HiveMQ - MQTT Essentials - All Core Concepts Explained*, n.d.). As these technologies continue to evolve, they drive efficiency and innovation across industrial sectors, while simultaneously raising important considerations regarding data privacy, standardization, and ethical implications. Future research and development efforts will need to focus on advancing technological solutions, fostering industry collaboration, and addressing societal concerns to fully realize the transformative potential of IIoT and MQTT in industrial settings.

2.4 Genetic Algorithm (GA)

Genetic algorithms are a type of optimization technique inspired by natural evolution and genetics. They work by maintaining a population of candidate solutions and iteratively modifying them through processes like selection, crossover, and mutation. Solutions with better fitness values are more likely to be selected for reproduction and pass on their characteristics to the next generation. Through this evolutionary process, genetic algorithms can find near-optimal solutions to complex problems. Key concepts include initialization, fitness evaluation, selection, crossover, mutation, and replacement of the population over multiple generations until a stopping criterion is met. (Goldberg, 1989). Amindoust (Amindoust, Asadpour, & Shirmohammadi, 2021) developed a hybrid genetic algorithm for multi-skill manpower shift scheduling, incorporating cross-training and reassignment of employees to optimize staffing costs while meeting demand and skill requirements. Wang (Wang, Zhang, Shi, Duan, & Liu, 2018) introduced a hybrid swarm intelligence algorithm that combines genetic algorithms and particle swarm optimization for solving multi-skill resource-constrained project scheduling problems, which share similarities with manpower shift scheduling. Their genetic algorithm considers multiple shifts and criteria, such as fairness and workload balance. These recent studies demonstrate the ongoing efforts to improve and adapt genetic algorithms for various aspects of manpower shift scheduling, including shift preferences, multi-period costs, multi-skill requirements, cross-training, reassignment, and rescheduling scenarios.

2.5 Client-Side Technologies

2.5.1 React Framework

React, a JavaScript library, is predominantly utilized for the construction of user interfaces, specifically within the realm of single-page applications. Its primary function is to manage the view layer in both web and mobile applications. This innovative library facilitates the design of simplistic views for each state within an application. A key feature of React is its ability to efficiently update and render the appropriate components when there are alterations in your data. This dynamic capability contributes to the creation of highly responsive and user-friendly applications (*React – A JavaScript library for building user interfaces*, n.d.).

2.5.2 JavaScript

JavaScript (JS) a high-level, interpreted programming language that conforms to the ECMAScript specification, is characterized as dynamic, weakly typed, prototype-based, and multi-paradigm. It enhances user-web-page interaction, making web pages more lively and interactive. (*JavaScript - MDN Web Docs*, n.d.)

2.5.3 TypeScript

TypeScript (TS), an open-source language, builds on JavaScript, one of the world's most used tools, by adding static type definitions. These types provide a way to describe the shape of an object and offer better documentation. (*TypeScript - MDN Web Docs Glossary: Definitions of Web-related terms*, n.d.)

2.5.4 Cascading Style Sheets

Cascading Style Sheets (CSS), a style sheet language, is used for describing the look and formatting of a document written in HTML. It primarily enables the separation of document content from document presentation, including elements such as layout, colors, and fonts. (MDN Contributors, n.d.-a)

2.5.5 Hypertext Markup Language

HTML (HyperText Markup Language), the standard markup language for documents designed to be displayed in a web browser, can be assisted by technologies such as CSS and scripting languages such as JavaScript. (MDN Contributors, n.d.-b)

2.6 Server-Side Technologies

2.6.1 Node js

Node.js is an open-source, cross-platform JavaScript runtime environment that executes JavaScript code outside a web browser. It's designed to build scalable network applications and is used by numerous major companies worldwide. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices. It comes with a rich library of various JavaScript modules, simplifying the development of web applications. Node.js applications can run on various platforms including Windows, Linux, Unix, Mac OS X, and more. It also provides a rich set of features including HTTP server libraries, making it excellent for developing microservices and APIs. The Node.js package ecosystem, npm, is the largest ecosystem of open source libraries in the world. (*Node.js - About Node js*, n.d.)

2.6.2 MongoDB

MongoDB is a source-available cross-platform document-oriented database program. Classified as a No-SQL database program, MongoDB uses JSON-like documents with optional schema. (*MongoDB*, n.d.)

2.6.3 AWS

Amazon Web Services (AWS) has emerged as a leading cloud computing platform, offering a wide array of scalable and flexible services for businesses and developers. Since its launch in 2006, AWS has continuously expanded its portfolio, now providing over 200 fully featured services from data centers globally (*About AWS*, n.d.). Key offerings include compute power through Amazon EC2, storage solutions like Amazon S3, and database services such as Amazon RDS (*AWS Products*, n.d.). AWS's pay-as-you-go pricing model and ability to scale resources on-demand have made it particularly attractive for startups and enterprises alike, enabling cost-effective infrastructure management (*AWS Pricing*, n.d.). The platform's robust security features, compliance certifications, and global infrastructure have contributed to its adoption across various industries, from healthcare to finance (*AWS Security*, n.d.). As cloud computing continues to evolve, AWS remains at the forefront, consistently introducing new services and technologies to meet the changing needs of its customers.

3 Software engineering process

The software engineering process for the Migdal-Or factory project commenced with a comprehensive on-site assessment and stakeholder interviews. Initial contact was established with Mr. Gabi Cohen, the facility director, who provided an overview of operations and personnel. This preliminary engagement revealed critical challenges in employee scheduling and real-time production data acquisition. Subsequent interviews and detailed analysis of provided data sheets facilitated a thorough understanding of work processes and system objectives, forming the foundation for requirement elicitation and project scope definition. System development proceeded iteratively, with a significant milestone being the coordination with the mechanical engineering team to establish protocols for sensor-based data collection from workstations, ensuring seamless integration with the database infrastructure. Throughout the development lifecycle, continuous collaboration with factory management was maintained to align the system with evolving requirements. This iterative approach allowed for adaptive refinement based on feedback from periodic review meetings, ensuring the final solution remained closely tailored to the factory's operational needs. The integration phase presented significant challenges, particularly due to the interdependence with the mechanical engineering team. This cross-functional collaboration necessitated careful coordination and communication to align software interfaces with hardware sensors, often resulting in unforeseen complexities and occasional delays. Furthermore, the dynamic nature of the project led to the emergence of new requirements and changes throughout the development process. These evolving demands necessitated frequent adjustments to the system architecture and functionality, requiring agile adaptation of the software design. Such modifications, while enhancing the final product's relevance and effectiveness, introduced additional complexity to the development timeline and resource allocation.

Current system - OvdimNet:

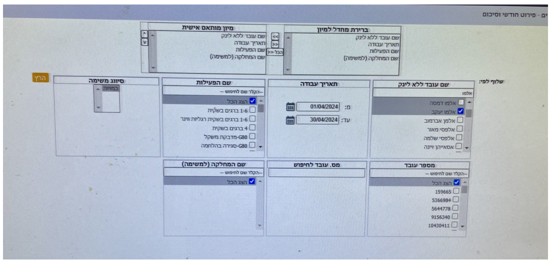


Figure 2: Selecting data to generate a report

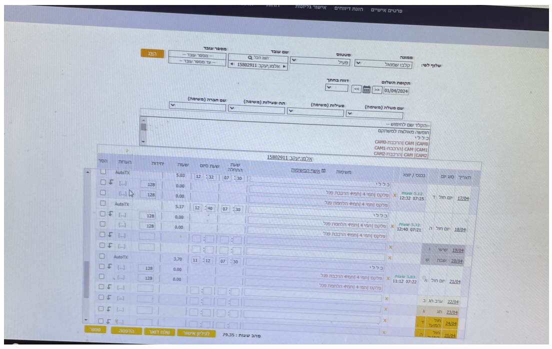


Figure 3: Recording employee productivity

Figure 4: Old system interfaces for report generation and productivity recording

Throughout the software engineering process, we maintained a focus on simplicity and clarity. We broke down complex concepts into easily understandable terms and used straightforward language to communicate our ideas and solutions effectively.

3.1 Genetic Algorithm

3.1.1 Genetic Algorithm (GA)

Genetic algorithm (GA) is an adaptive heuristic search algorithm that is popular for solving complex optimization problems such as scheduling algorithm. This algorithm uses historical information to direct the search. The GA schedules shifts to different categories such as the time of the shift that has to be scheduled equally between the workers. Scheduling shifts to the workers of the 'Migdal-Or' has some constraints that need to take into account such as:

- All the workers must be assigned.
- All active workstations must be staffed.
- The expected products are as close as possible to the necessary weekly production quantity.

In recent years, researchers have proposed various genetic algorithm approaches to tackle different aspects and variations of the manpower shift scheduling problem. These approaches often involve encoding potential solutions as chromosomes, evaluating their fitness based on the problem constraints and objectives, and iteratively evolving the population of solutions through genetic operations like crossover and mutation.

3.1.2 Genetic algorithm in our system

Our system aims to efficiently match tasks with employees whose skills and abilities best fit the requirements. Using a genetic algorithm, the system analyzes task data from the database to create a ranked list of employees for each job position. Variations are introduced through crossover and mutation to explore new combinations of employee-task pairings. Over successive generations, the population converges toward optimal worker assignments for each task. Upon convergence, the algorithm generates a ranked list of employees, ordered by their suitability for each specific station based on their unique skill profiles.

3.1.3 Genetic algorithm flow

The genetic algorithm starts by initializing a population of task assignments, with each assignment representing a potential solution to the matching problem. Each assignment is a list of employees for a specific station. The algorithm evaluates each assignment's fitness by calculating the average match percentage between the station's requirements and the skills of the assigned employees. It then enters a loop, selecting two parent assignments from the population based on their fitness scores using a roulette wheel selection. A point crossover is performed between the two parents to produce two offspring, combining information from the parents to create new assignments. After crossover, the algorithm randomly changes one of the offspring's employee assignments (mutation). This process continues for a specified number of generations, maintaining a predetermined population size. In each generation, the top two assignments from the previous generation are preserved (elitism), and the rest of the population is replaced by new assignments created through crossover and mutation. After the specified number of generations, the algorithm evaluates the final population and extracts the top three unique employee assignments for each station from the sorted final population, identifying the best matches based on the employees' skills and the station's requirements.

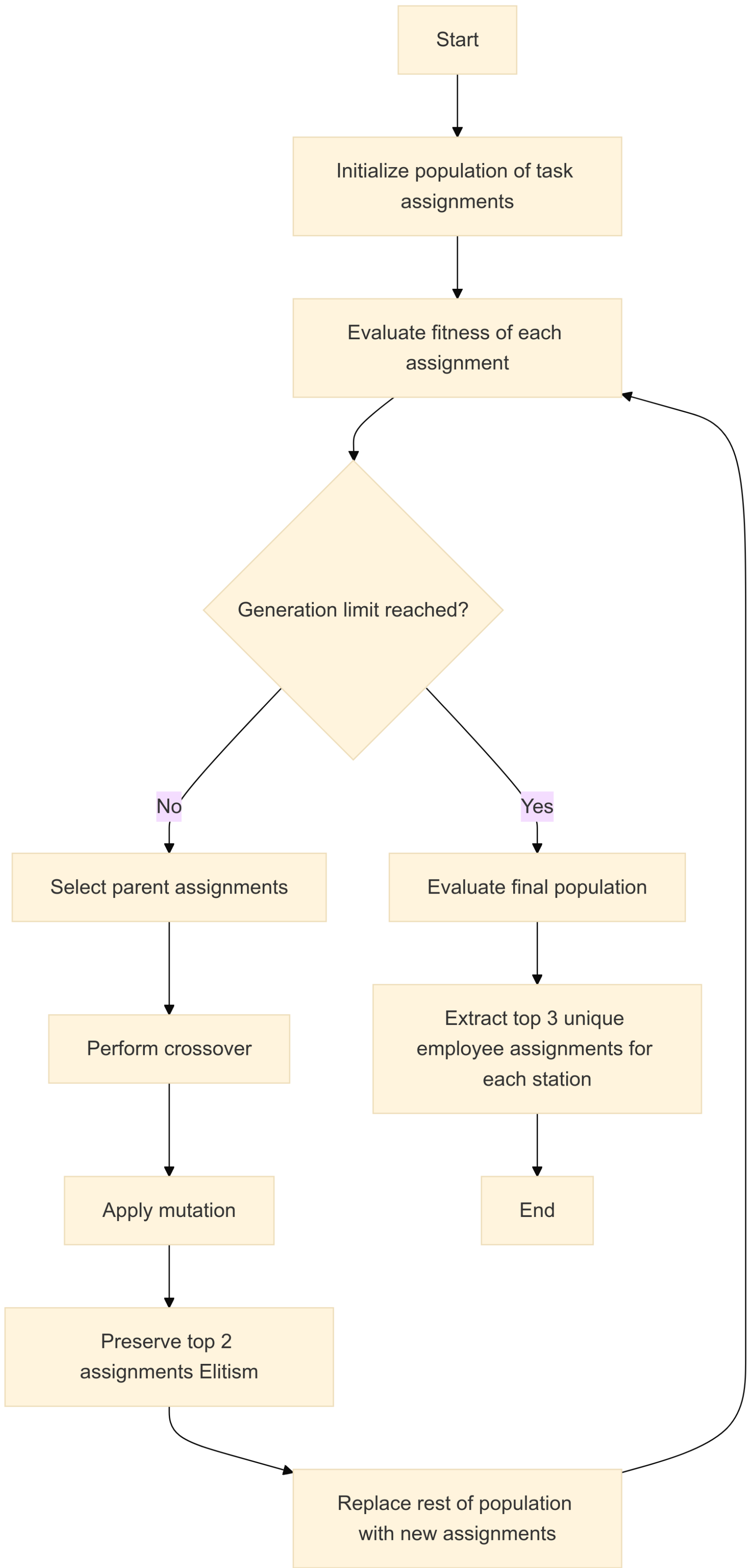


Figure 5: GA flow

3.1.4 Using Algorithm Output

Once the genetic algorithm outputs the ranked list of top employees for each station, the manager can use this information to make informed decisions about staffing. By considering the top three employees suggested by the algorithm for each station, the manager can evaluate the specific skills and suitability of these workers for the station's requirements. The manager might also consider additional factors such as current project needs, team dynamics, and employee availability or preferences. Ultimately, the manager selects the most suitable employee for each station, ensuring that the workers' skills are effectively utilized, and the organization benefits from an optimized workforce allocation. This approach allows for flexible and efficient staffing decisions, matching the best-suited employees to each station based on both algorithmic recommendations and managerial insight.

3.2 Interviews and meetings with 'Migdal-Or' Managers

The research process involved a series of structured engagements with plant management personnel. These interactions served multiple critical functions: establishing mutual familiarity, elucidating requirements, and validating the alignment of our research objectives with the facility's operational needs.

The initial site visit focused primarily on familiarization with the manufacturing environment, including workspace layouts and the challenges encountered by both operational staff and management in their daily functions. During a comprehensive tour of various departments, we observed practical work processes, including a prototype air pressure testing station at Shluker, which employs sensor technology to extract data on worker productivity.

Subsequent to this initial assessment, we identified two paramount issues faced by management: employee allocation and real-time production data acquisition. Follow-up meetings were conducted with the aim of refining system requirements, emphasizing relevant data metrics, determining optimal presentation methodologies, ensuring display visibility, and addressing protocols for data entry and exportation.

A separate consultation was held with representatives from the mechanical engineering department, during which they presented their existing system for information collection and transmission from the testing station. We established mutual agreements regarding the requisite information and the protocol for data transfer.

Following this, an additional on-site meeting was convened with factory management to demonstrate the system and verify communication protocols. This meeting also provided an opportunity to discuss the employee placement process and system visibility considerations with plant managers.

A final meeting was initiated with plant management and corporate information systems personnel to reach consensus on cost structures and deployment strategies. This session culminated in a summary of the procedures for ongoing system maintenance and management.

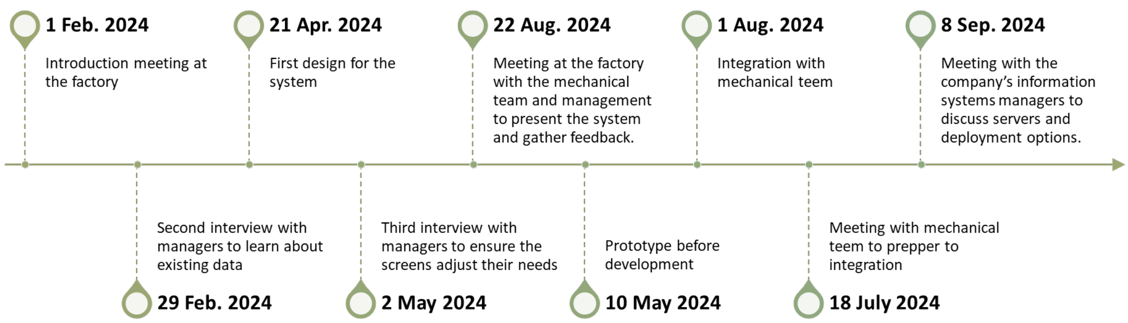


Figure 6: Interviews timeline

3.3 Requirements

3.3.1 Non-Functional Requirements (NFR)

1. Only factory manager can add new employees - **security**
2. An employee can be assigned maximum of 2 workstations per day. - **Capacity**
3. All department manager able to assign employees from every department. - **Access**
4. The assignment determines according to the criteria: an assignment for each employee, the output is as optimal as possible (not necessarily the best, but not the lowest either). - **Constraints**
5. A placement offer is given to any working station that a manager choose. - **Management**
6. The reports can be filtered by: working station, employees and date. - **Usability**
7. Assignment in a position is possible only if an employee is qualified to work in the job position. - **Data integrity**
8. Department managers can add qualification to job positions for employees that registered under them - **Modifiability**
9. For the air pressure test station in Shluker - receiving the information is done through a sensor directly into the system. - **Dependency on other parties**
10. The system response to each action is up to 2 seconds - **Performance**

3.3.2 Functional Requirements (FR)

1. The system allows adding new employees.
2. The system allows view employee profile.
3. The system allows edit employee profile.
4. The system allows assignment of workers on working stations.
5. The system allows the saving of an employee's average outputs.
6. The system allows export employees list to Excel file.
7. The system allows export daily assignments table to Excel file.
8. The system displays Shluker production status on real-time.
9. The system produce a daily report.
10. The system suggests a possible assignment of an employee for working positions.

3.4 Use Case Diagram

The system have two types of users - the plant manager and department managers. Both have the same actions to perform, but upon request - only the plant manager have permission to add a new employee to the system and edit the details of an existing employee.

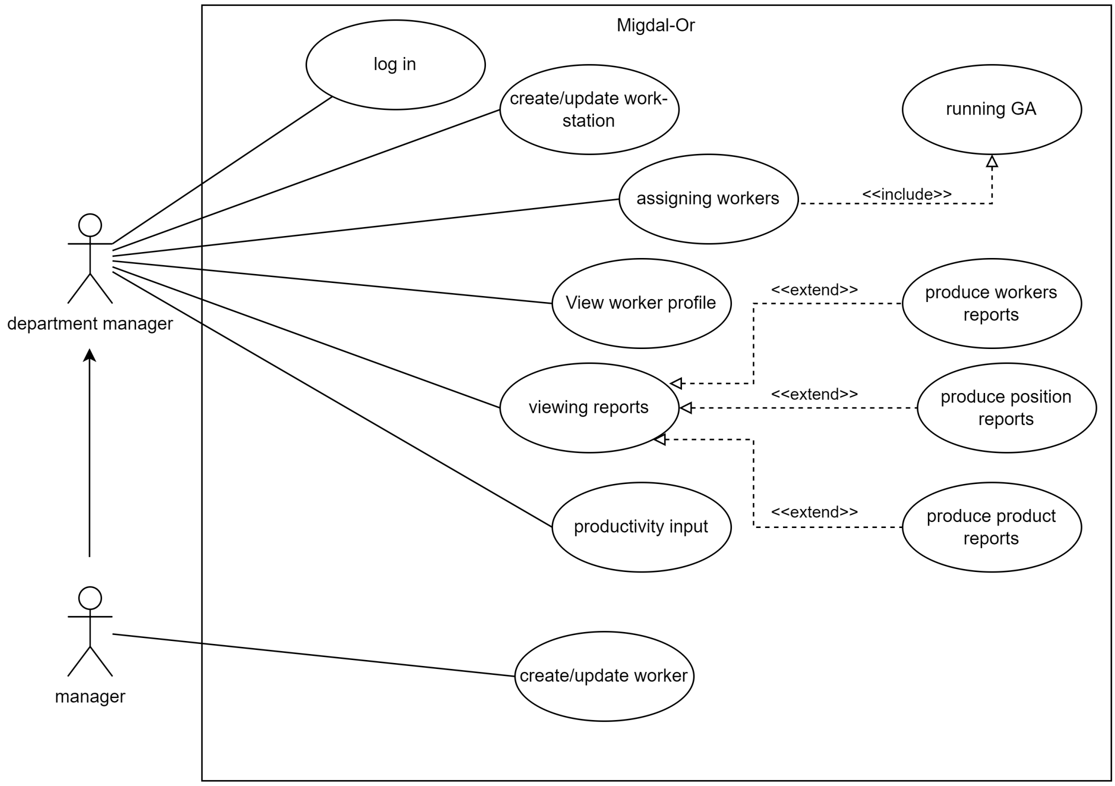


Figure 7: Use-Case diagram

3.5 Activity Diagram

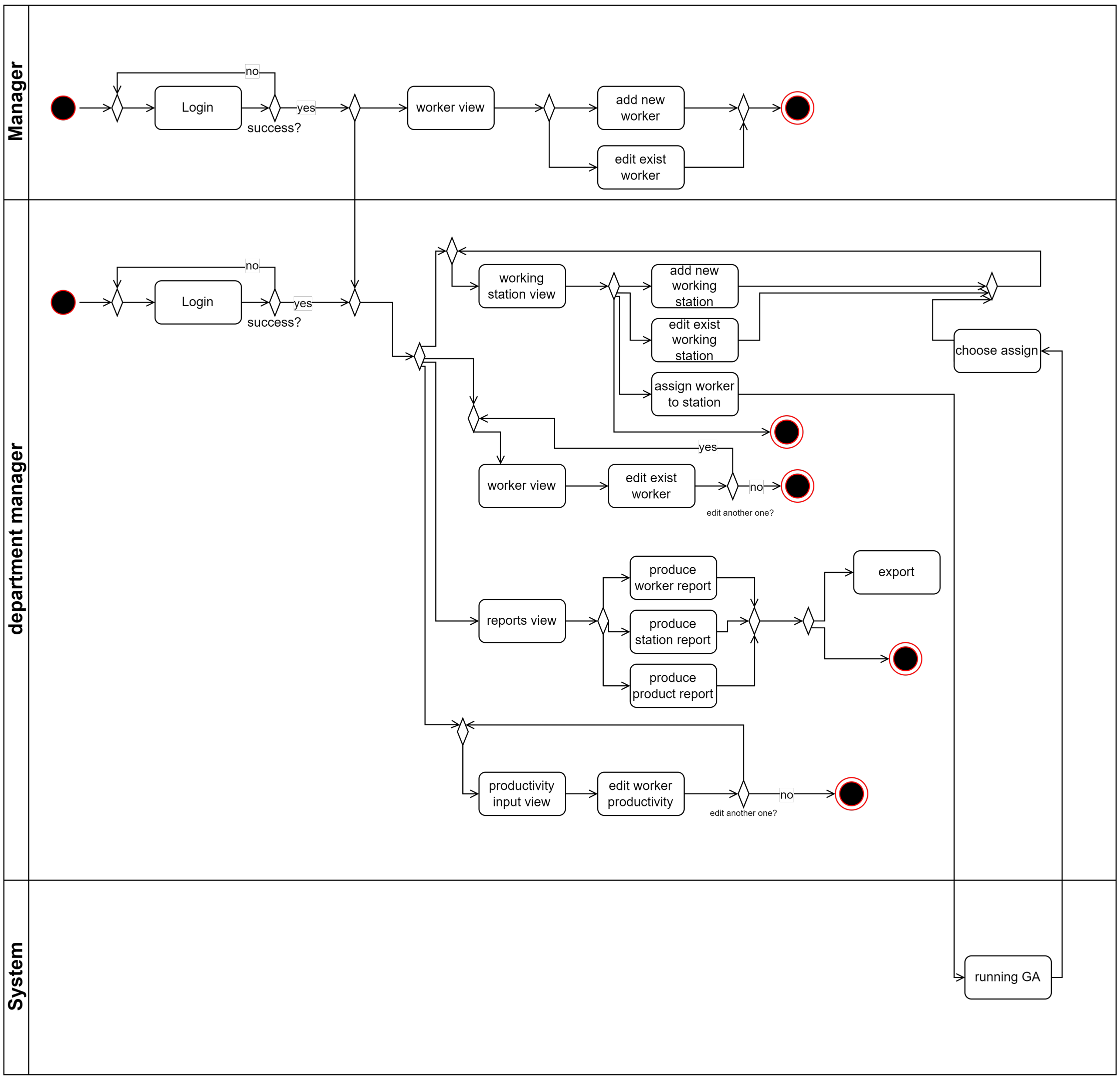


Figure 8: Activity diagram

3.6 Class Diagram

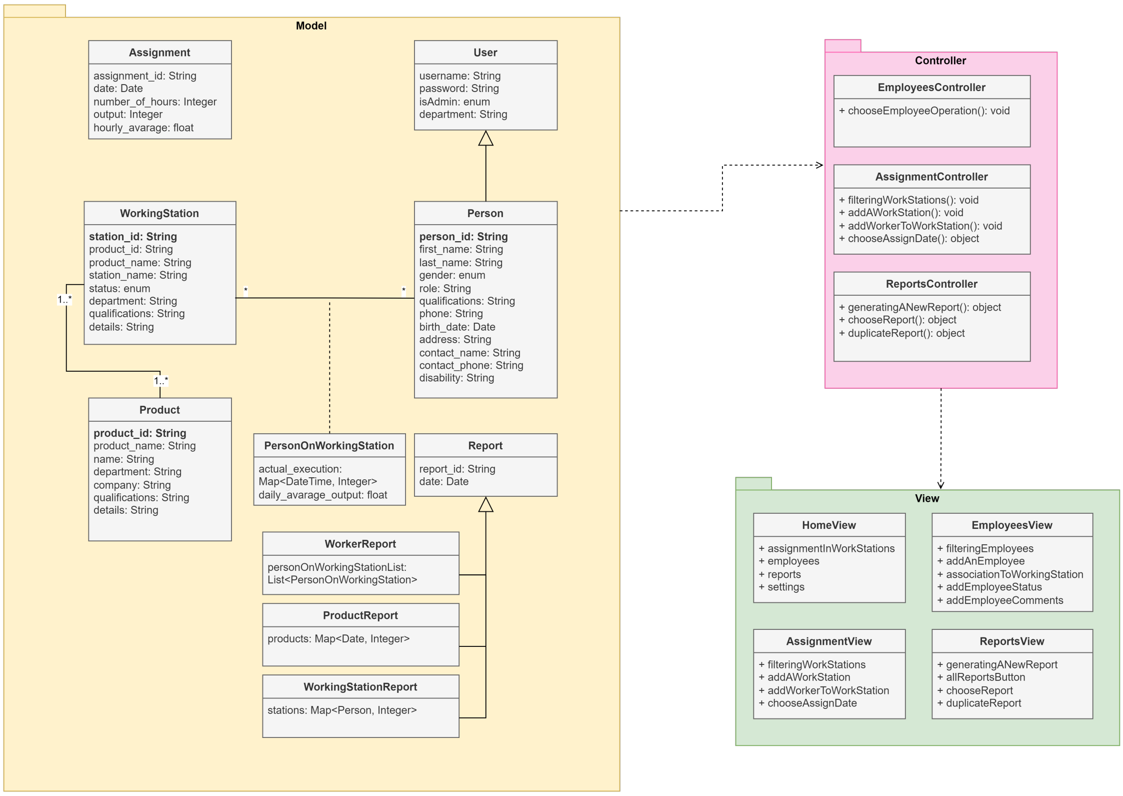


Figure 9: Class Diagram

3.6.1 Model

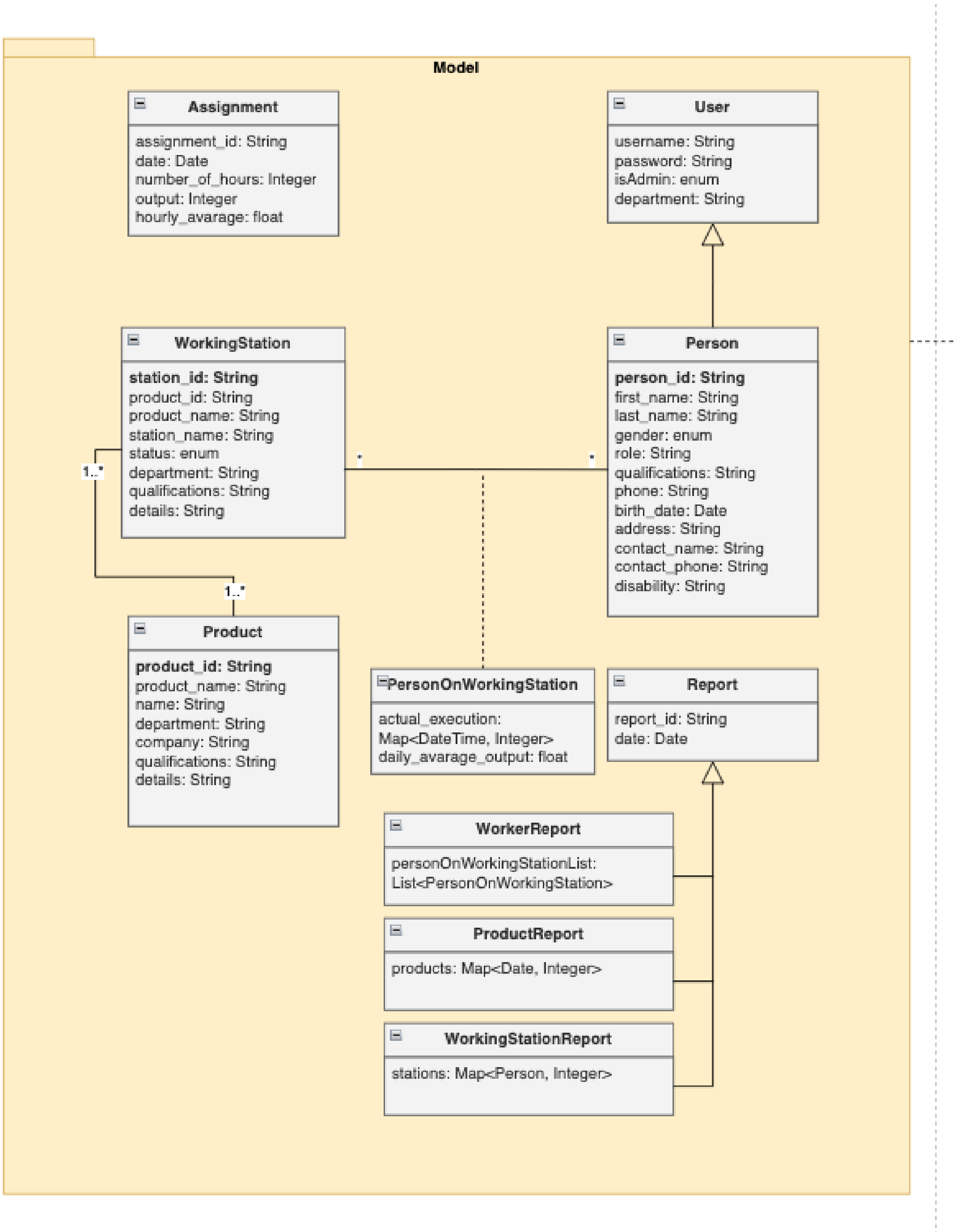


Figure 10: Class Diagram - Model

3.6.2 View

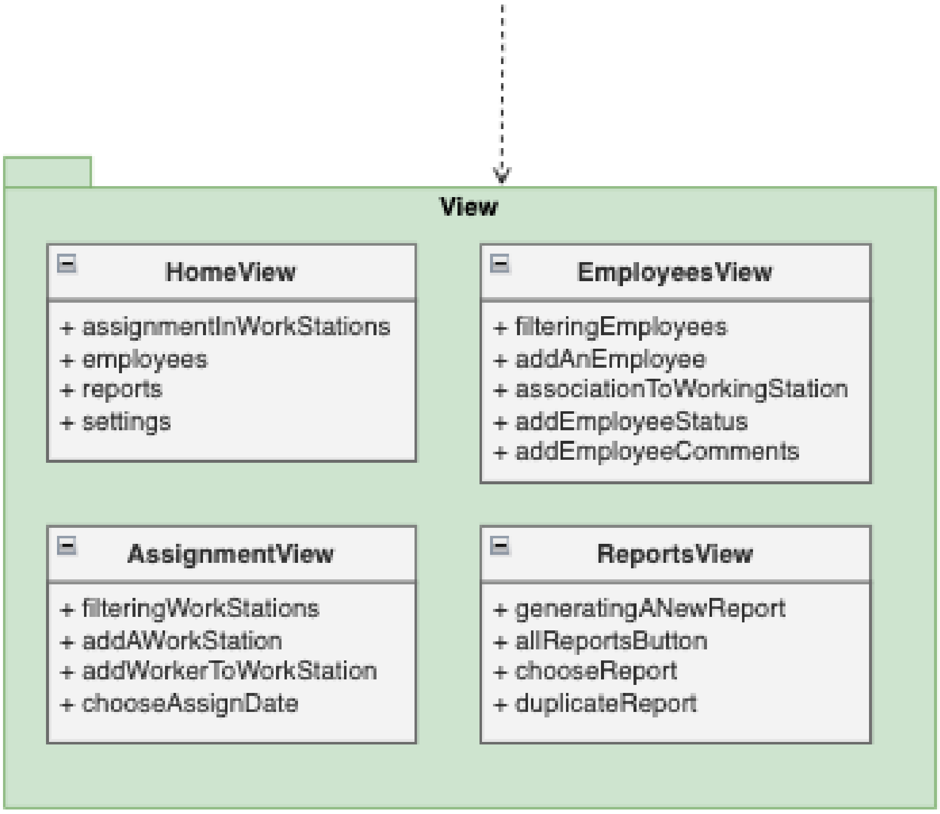


Figure 11: Class Diagram - View

3.6.3 Controller

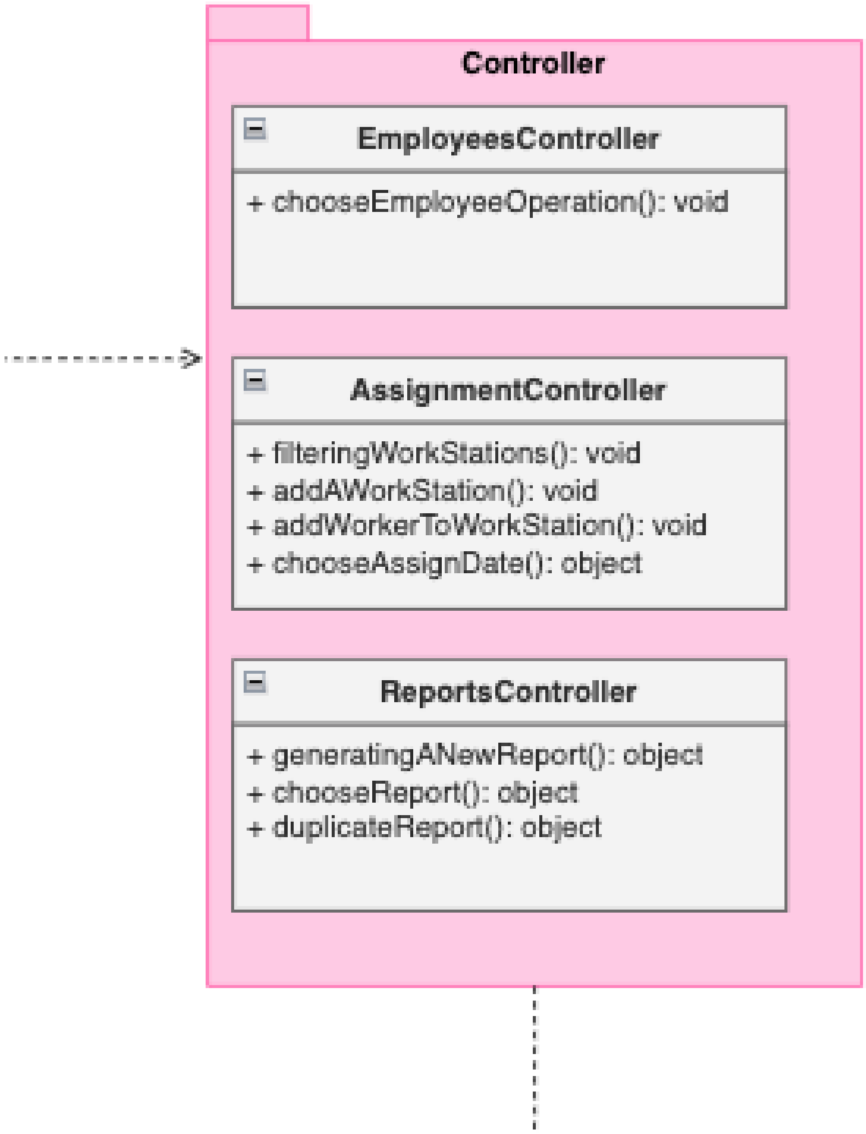


Figure 12: Class Diagram - Controller

4 User Manual

4.0.1 The application

To answer ‘Migdal-Or’ managers need, we designed web application that meet several requirements:

1. Suggestions for placing employees in positions according to their skills.
2. Producing and viewing reports according to several parameters.
3. Monitoring on real-time ‘Shluker’ air pressure test produce.
4. Editing and adding new employees.

Login screen:
The ’Login’ page serves as the initial interface for the application, functioning as a crucial security checkpoint. Users are required to input their unique credentials, consisting of a username and password. This authentication process ensures that only authorized individuals gain access to the system. Upon successful verification of these credentials, the user is seamlessly redirected to the ’Home’ page, where they can interact with the application’s main features and content. This implementation adheres to best practices in user authentication and access control, providing a secure foundation for the application’s functionality.

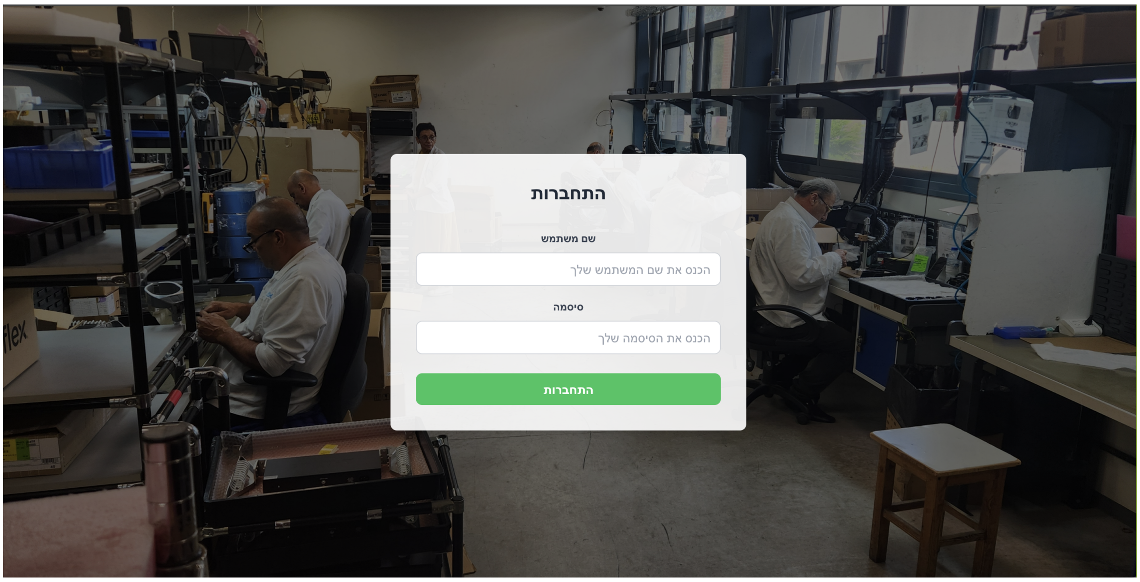


Figure 13: login screen

Home screen:

The 'Home' page functions as a dynamic dashboard, providing real-time insights into critical system metrics. It visualizes key performance indicators (KPIs) including the current count of active and inactive employees, the number of defective Shlukers and a tally of operational stations. This interface demonstrates the application's capability to aggregate and present complex data in an easily digestible format, facilitating quick decision-making for management.

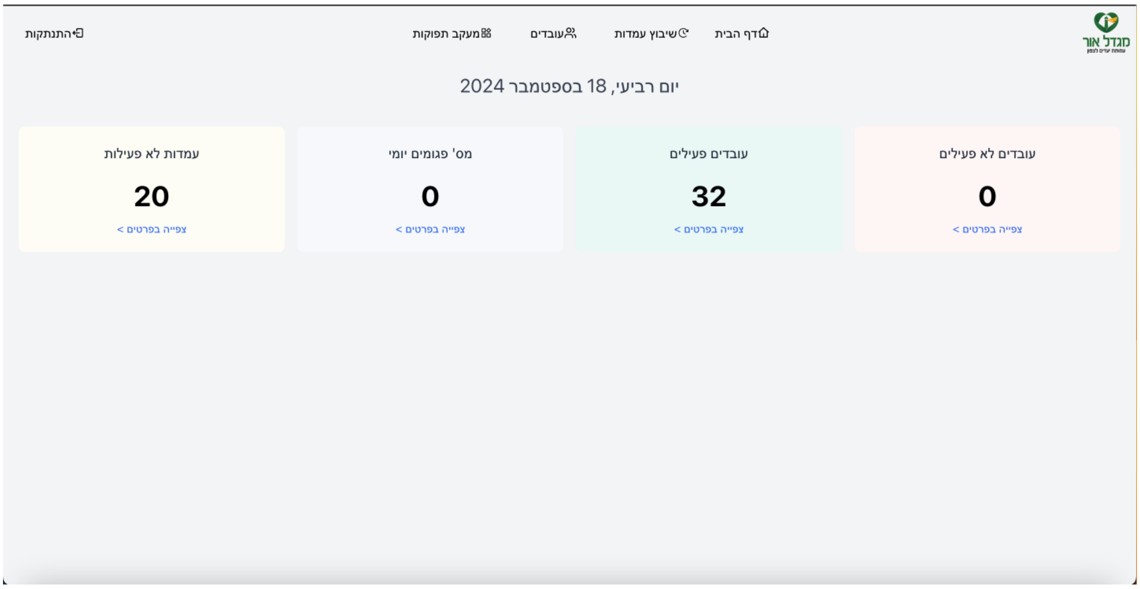


Figure 14: home screen

Workers screens:
This screen shows all the details of the employees in the factory. After choosing an employee from the list a card with his details will be open. There is an option to filter employees by department and export all the employees and their details to Excel file by clicking on export to Excel button.

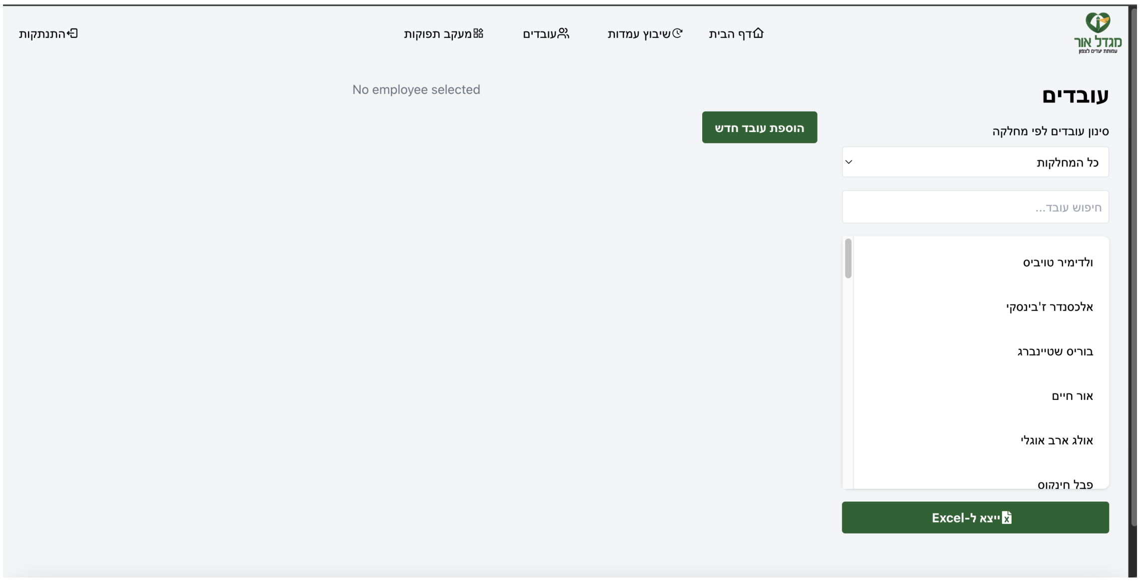


Figure 15: workers screen

After choose employee from the list, a details card displays on the screen. There is an option to edit employee details by clicking on the edit button.

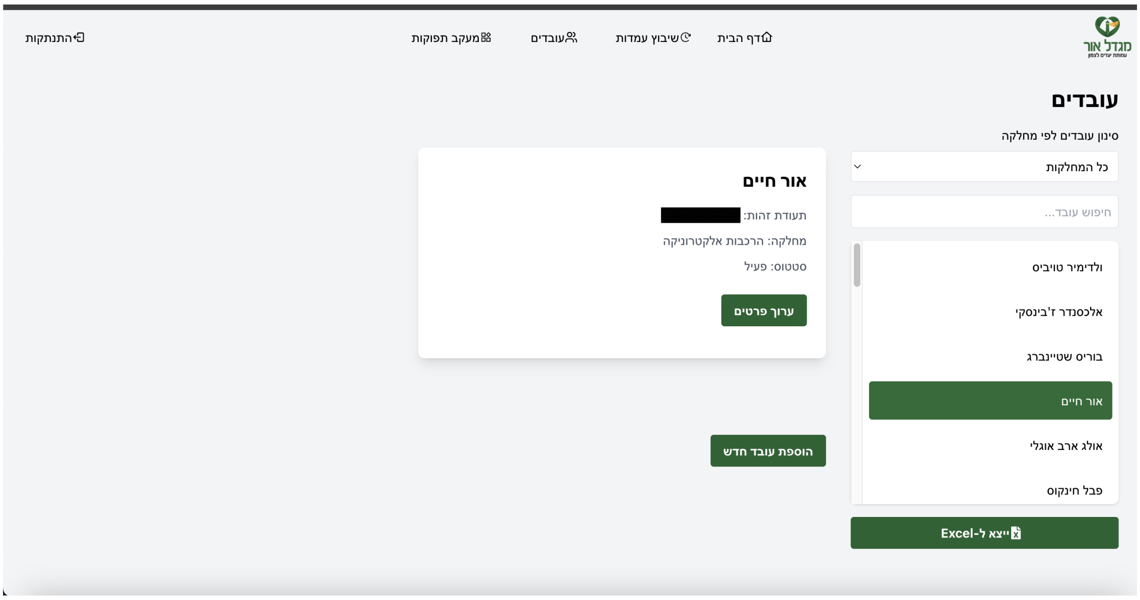


Figure 16: worker details

On edit employee form, the user can change employee department, status and his qualification to work on working station.

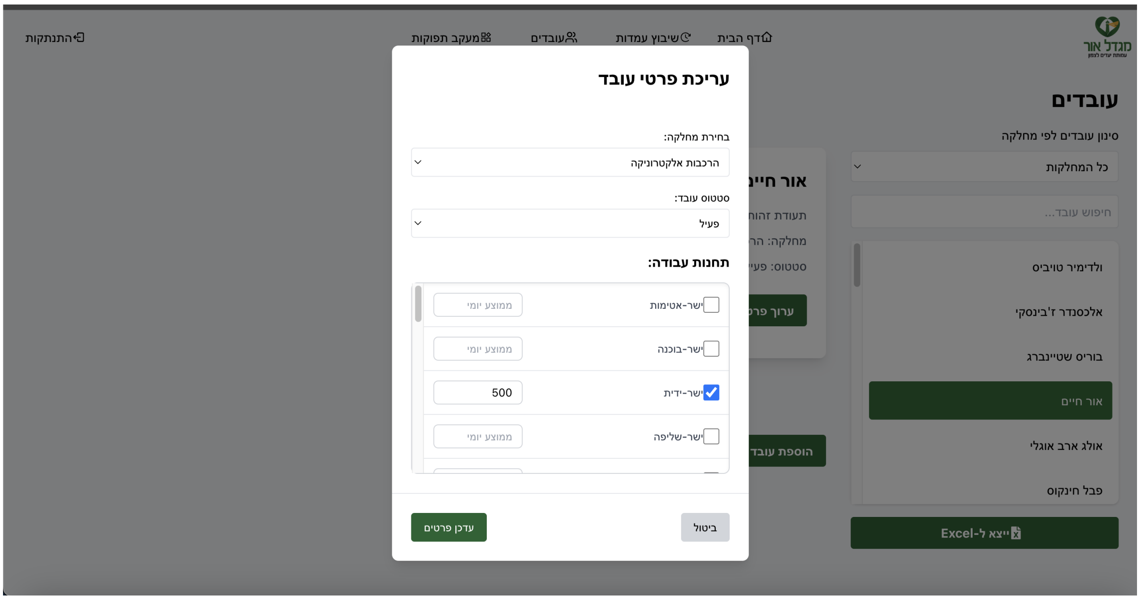


Figure 17: edit employee form

Upon activating the 'Add New Employee' function, an input form is dynamically generated. This feature is exclusively accessible to users with factory manager privileges, implementing a role-based access control system. The form mandates the completion of all fields to ensure comprehensive data collection. Upon submission, the system validates the input and securely stores the new employee's information in the database.

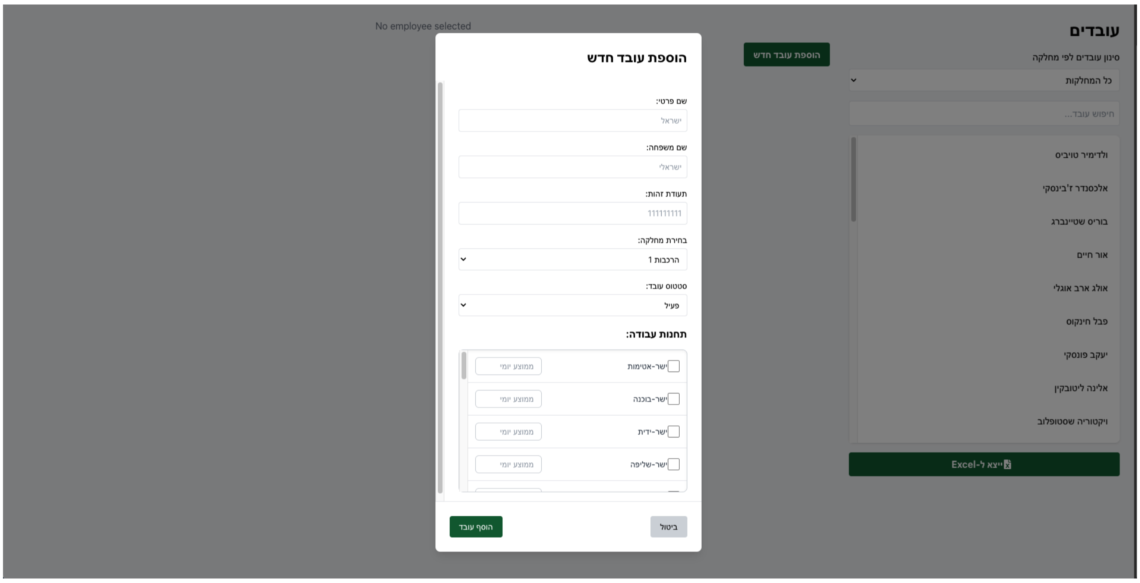


Figure 18: add employee form

Scheduling and work station screen:
The interface facilitates the allocation of employees to specific work positions. The system presents a comprehensive table displaying the current employee roster and their respective assignments as of a specified date. This tabular representation allows for efficient management of workforce distribution. The functionality includes options to remove existing employee assignments and export the assignment data to a spreadsheet format for further analysis or record-keeping purposes. To initiate a new assignment, the user must select a workstation from a predefined list of available positions. This step ensures that employees are allocated to appropriate roles within the organizational structure.

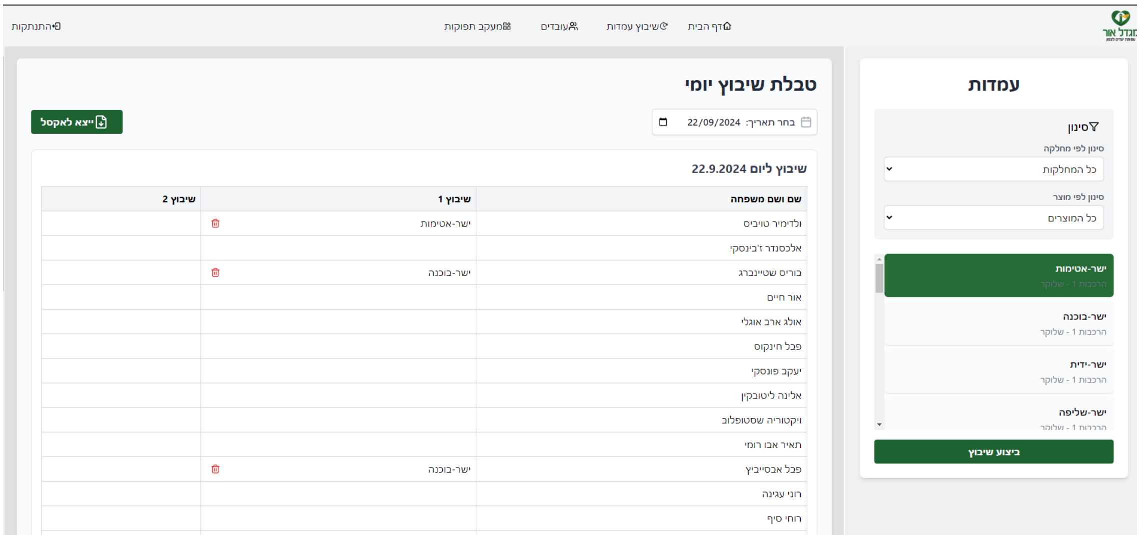


Figure 19: scheduling screen

After selecting a job position and clicking "Make Placement," the user see a screen with the system's suggested employee placements. The list is ordered by recommendation of the Genetic algorithm, with the most suitable employee at the top. The user can choose how many positions to fill. The system assign that number of employees from the list. Note: Each employee can only be assigned to a maximum of two work positions per day. If the user tries to assign a third, they will receive an error message.

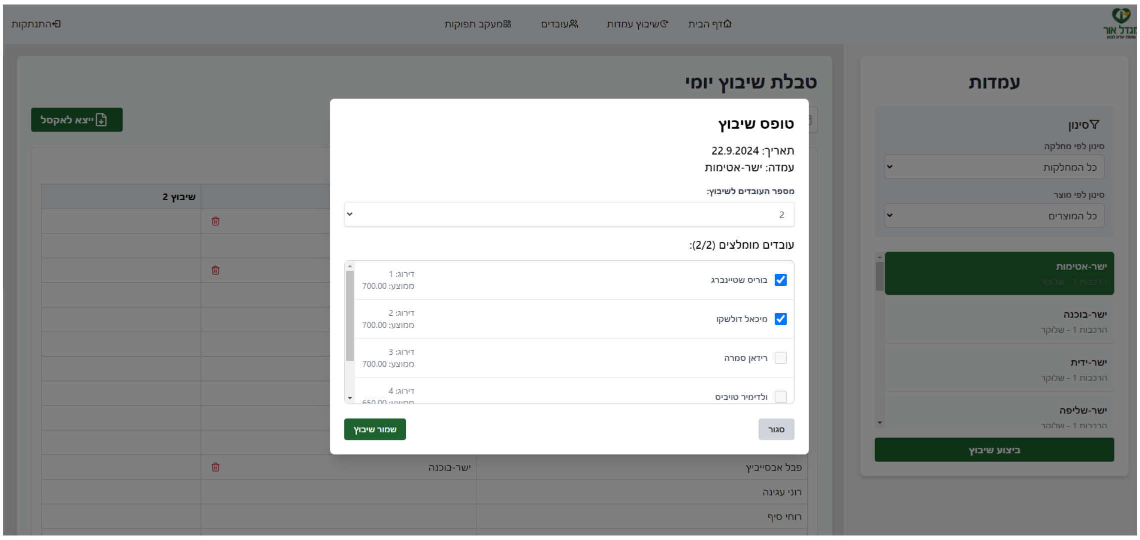


Figure 20: assignment form

Reports screen

This screen allows the user to generate reports and view real-time progress of work at the Shluker's air pressure test position for the current day. The interface provides report generation functionality and real-time status updates of the air pressure test position. The user can utilize these features to monitor productivity and make informed decisions based on up-to-date information.

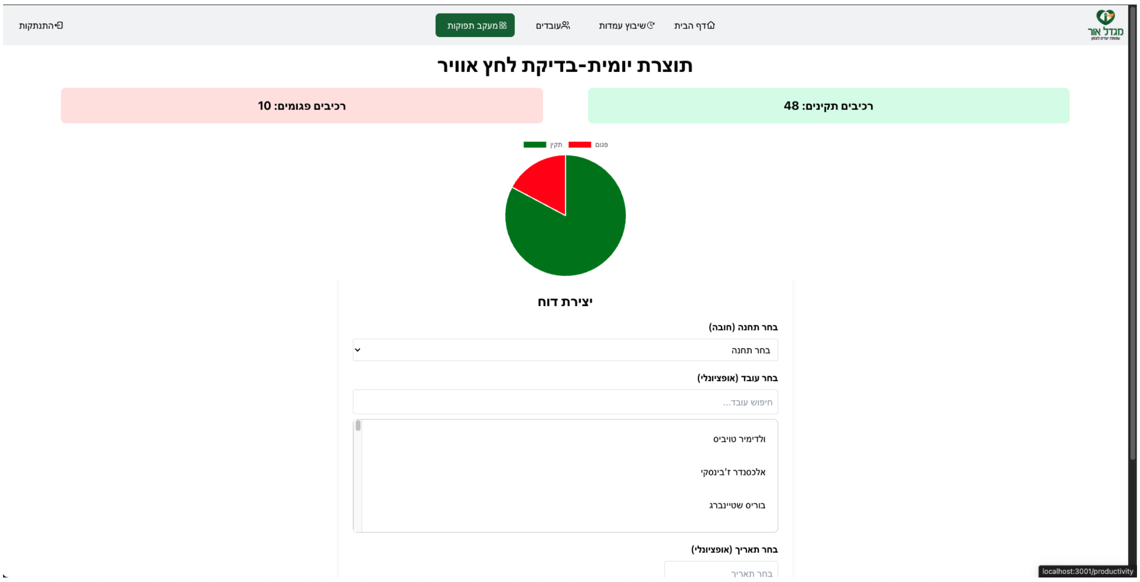


Figure 21: reports screen

First, the user enter the details of the report he wishes to generate. Selecting a workstation is mandatory and selecting a specific employee on whom the report generated or a specific date is optional for the user. Note: The system's current phase concentrates on the air pressure test station, transitioning from manual to automated data collection as per customer specifications.

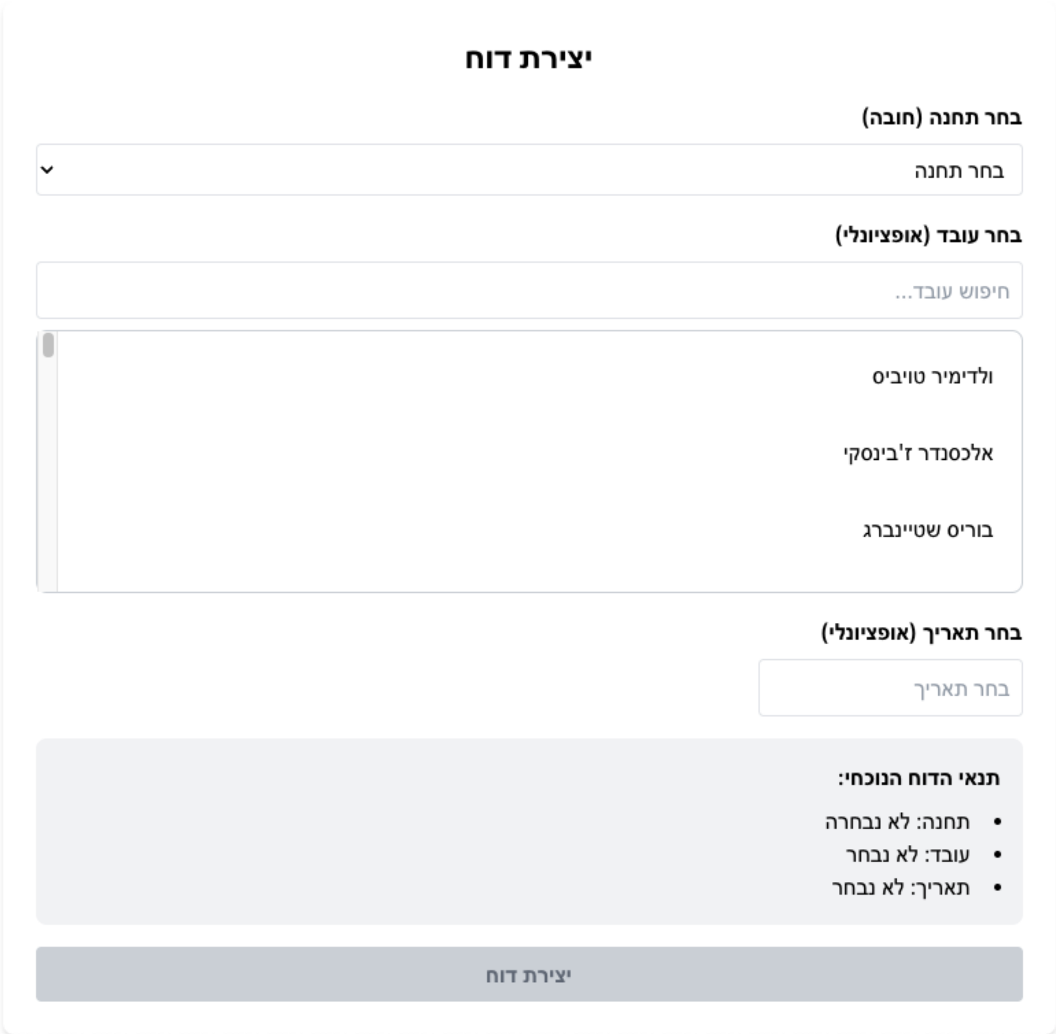


Figure 22: Reports generate form

There is three options for report:

1. If only working station is selected - the report shows monthly report of the amount of produce each day during the last month.

2. If working station and employee selected - the report shows monthly report of the amount of produce each day during the last month of the specific employee.

3. If working station and date selected - the report shows daily report of the amount of produce on the chosen date.

4. If working station, date and employee selected - the report shows daily report of the amount of produce of the selected date of the chosen employee.

The user can download the report and save it on his coputer - the file contains table with the relevant data.

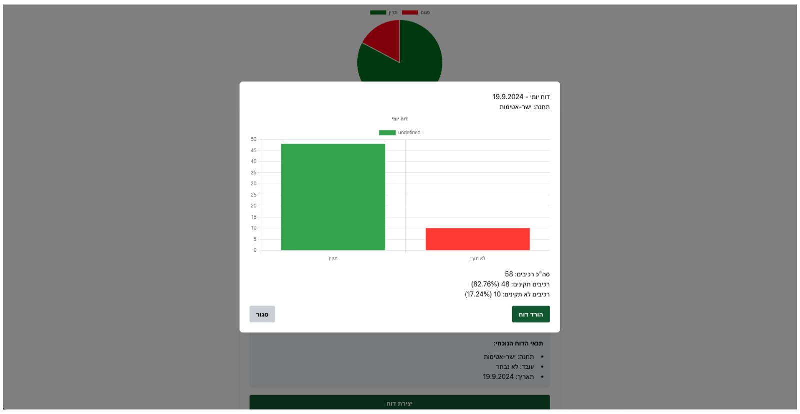


Figure 23: Daily report

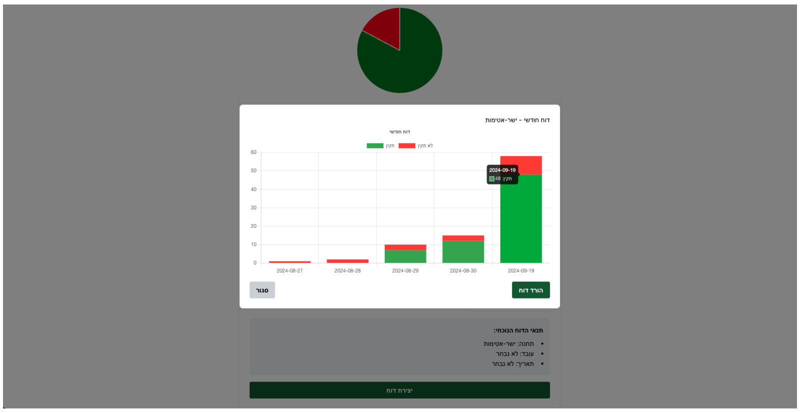


Figure 24: Monthly report

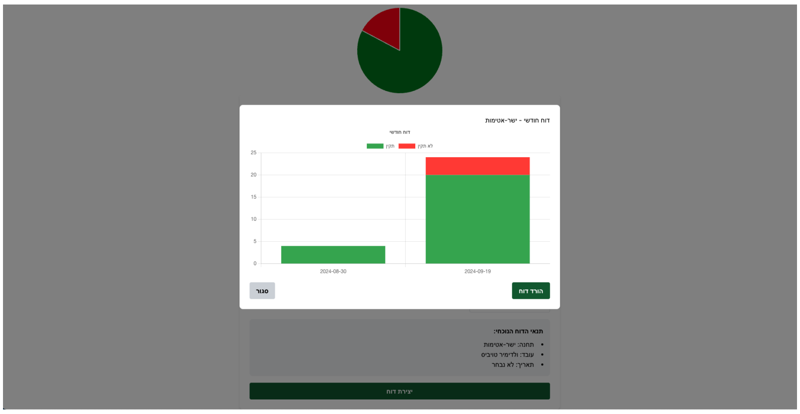


Figure 25: Employee report

5 Developer Manual

This manual provides technical documentation for the 'Migdal-Or' workforce management system. Key Technical Specifications:

- Frontend: React
- Backend: Node.js
- Database: MongoDB
- Sensor Data Communication: MQTTX
- Current Deployment: Vercel (for testing and development)
- Planned Production Deployment: Amazon Web Services (AWS)

Our tech stack is designed for flexibility, strong community support, and efficient real-time data processing. MQTTX is utilized for receiving data from sensors, facilitating IoT integration crucial for our workforce management system. The following sections detail the system architecture, setup procedures, key components, and best practices for development and deployment. We encourage all developers to familiarize themselves with this manual and contribute to its improvement as the project evolves.

5.1 Product Design

5.1.1 System Architecture

Our project epitomizes an amalgamation of cutting-edge technologies that converge to orchestrate a sophisticated digital ecosystem. With React at its forefront, alongside the dynamic capabilities of JavaScript and the structural clarity provided by TypeScript (*TypeScript - MDN Web Docs Glossary: Definitions of Web-related terms*, n.d.), the project exhibits a commitment to crafting interfaces that are not only visually compelling but also robust and responsive. Complementing these front-end technologies, Cascading Style Sheets (CSS) (MDN Contributors, n.d.-a) delineate the aesthetic blueprint, while HyperText Markup Language (HTML) (MDN Contributors, n.d.-b) serves as the foundational bedrock for content structuring. Enhancing the server-side architecture, Node.js, with its event-driven, non-blocking I/O model, offers scalable and efficient backend solutions, seamlessly integrating with the JavaScript-based front-end. MongoDB, embracing the tenets of NoSQL, furnishes a flexible and scalable data management solution. The system is deployed on Amazon Web Services (AWS), leveraging its robust cloud infrastructure to ensure high availability, scalability, and performance. Together, these disparate yet synergistic technologies engender a multifaceted ecosystem poised at the nexus of innovation and functionality, epitomizing the quintessence of contemporary software engineering paradigms in a cloud-native environment.

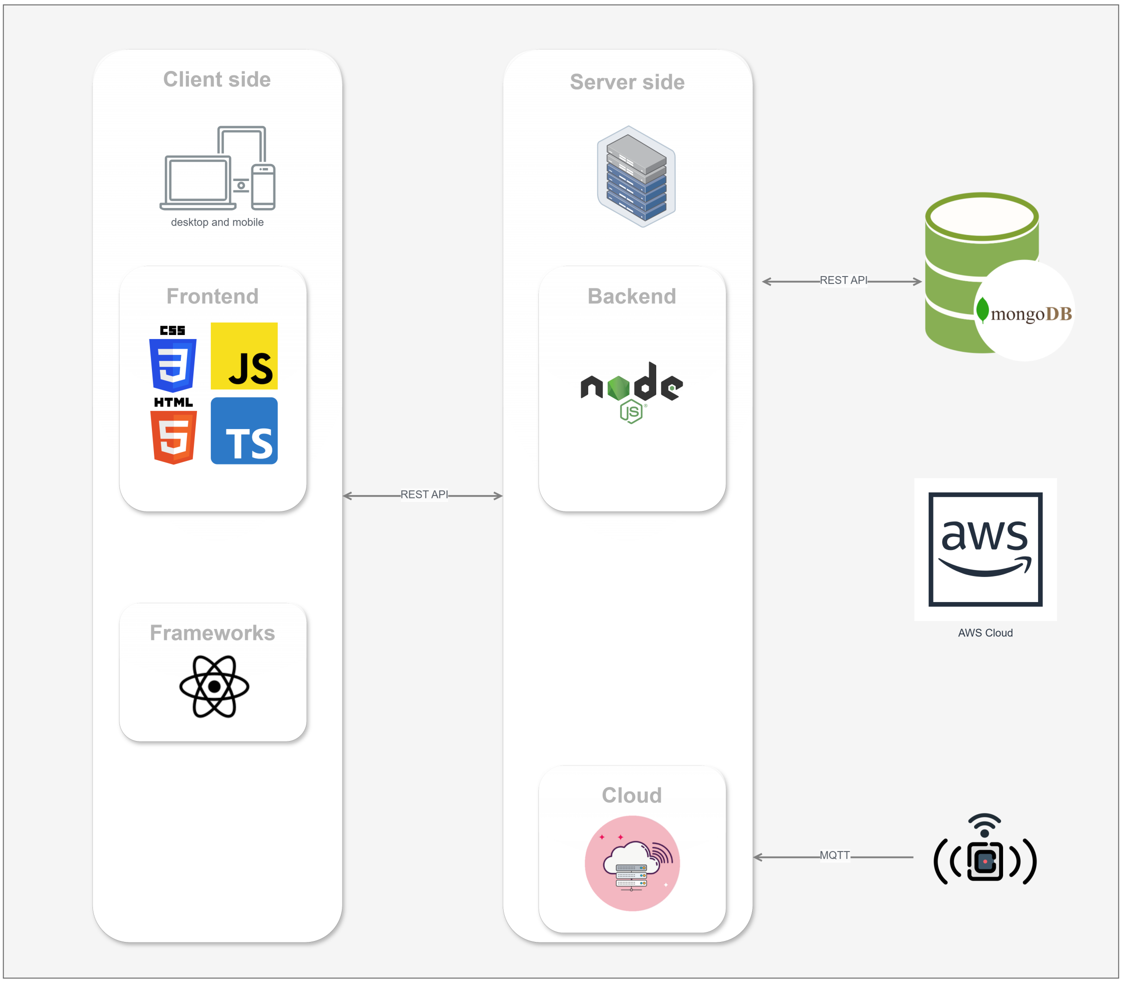


Figure 26: Software Architecture

5.1.2 Database Schema

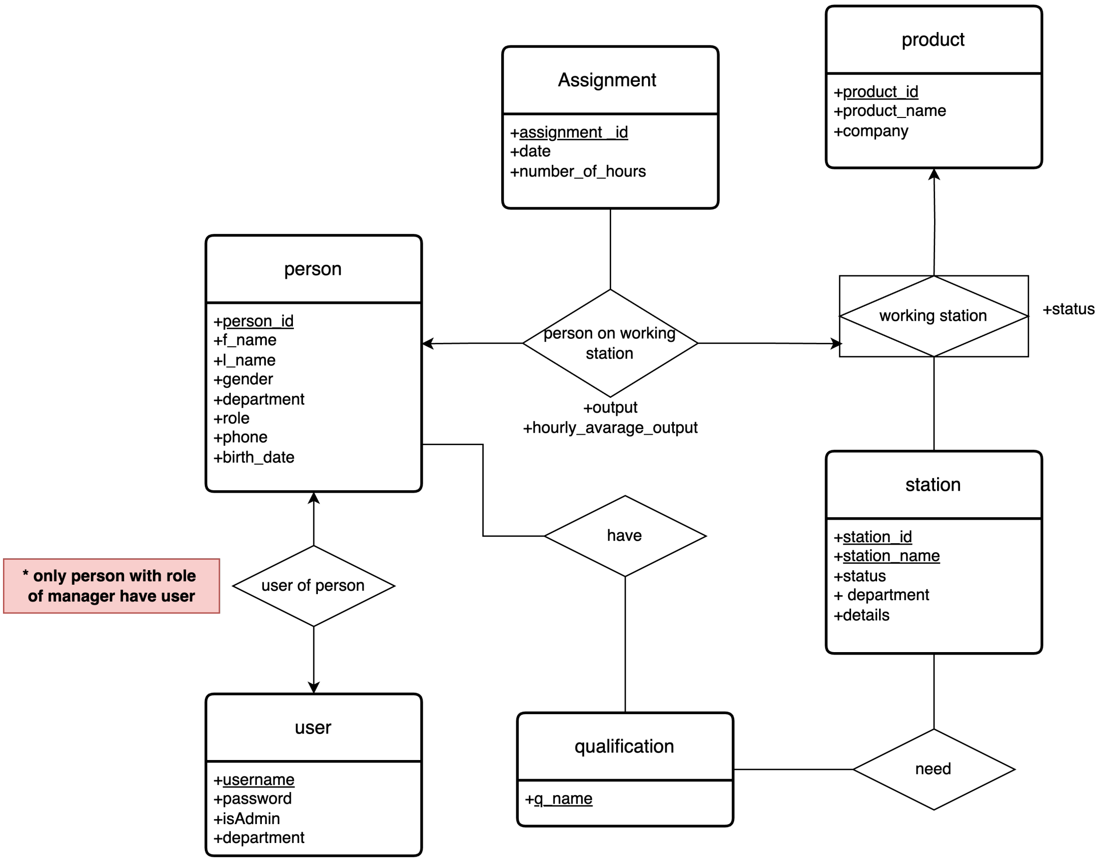
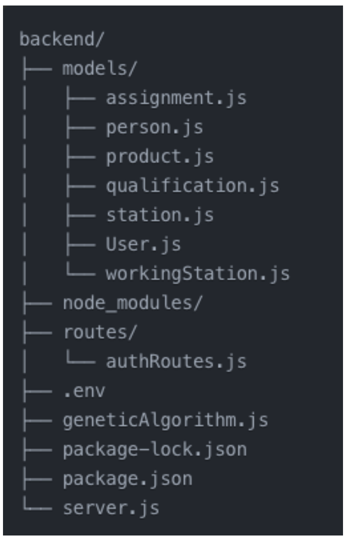


Figure 27: Database Template

5.2 Backend

The backend of our system is built with Node.js and follows a modular structure for better organization and maintainability.

5.2.1 Directory Structure



```
backend/
├── models/
│   ├── assignment.js
│   ├── person.js
│   ├── product.js
│   ├── qualification.js
│   ├── station.js
│   ├── User.js
│   └── workingStation.js
├── node_modules/
├── routes/
│   └── authRoutes.js
├── .env
├── geneticAlgorithm.js
├── package-lock.json
├── package.json
└── server.js
```

Figure 28: Backend directory

5.2.2 Key Components

1. Models: Define the data structures and schemas for our MongoDB database
2. Routs: Includes authRoutes.js that handles user authentication processes such as login and logout. Crucial for securing the application and managing user access.
3. Genetic Algorithm: Contains the logic for optimizing worker assignments to stations
4. Server Configuration: Includes Express.js setup, middleware configuration, and API route integration

5.3 Frontend

The frontend of our application is built using React, with a clear separation of concerns and modular component structure.

5.3.1 Frontend directory

```
src/
├── components/
│   ├── employees/
│   │   └── [Employee-related components]
│   ├── reports/
│   │   └── [Reporting components]
│   └── stations/
│       └── [Station-related components]
├── pages/
│   └── [Page components]
├── App.jsx
├── index.css
├── index.js
└── index.html
public/
└── [Static assets]
```

Figure 29: Backend directory

5.3.2 Key Components

1. Modular structure with components grouped by functionality (employees, reports, stations).
2. Page Components: Separate directory for main page components (HomePage, LoginPage, etc.) that facilitates easy navigation and route management
3. Static Assets: Includes images, icons, and other static files

5.4 Deployment Process

Our system is currently deployed on Vercel, which provides a streamlined process for our React frontend and Node.js backend. This section outlines the current deployment process and briefly mentions future plans.

5.4.1 Deployment

For now, the system deployed on Vercel for testing the system and submit the project. In the near future, after we receive a user from the factory, we will perform a deploy to the system in AWS according to factory requirement.

5.5 Testing Plan

No.	Test	Description	Status
1	User Authentication	Verify login, password reset, and access control	Passed
2	Employee Management	Test adding, editing, and validating employee data	Passed
3	Station Assignment	Verify genetic algorithm for assigning employees to stations	Passed
4	Report Generation	Test creation and accuracy of various system reports	Passed
5	Real-time Data Update	Verify system updates with simulated sensor data via MQTTX	Passed
6	UI Responsiveness	Check interface across devices and screen sizes	Passed
7	Database Integration	Test CRUD operations and data persistence	Passed
8	API Endpoint Testing	Verify all API endpoints for correct responses and error handling	Passed
9	Security Testing	Perform penetration testing and check for vulnerabilities	Passed
10	Performance Testing	Conduct load testing and optimize response times	Passed
11	Integration Testing	Verify seamless operation between all system components	Passed
12	Backup and Recovery	Test data backup procedures and system recovery scenarios	Passed

table Comprehensive System Test Results

Table 1: Test plan

6 Achievements

In the second phase of our project, we have successfully developed and implemented a comprehensive system that significantly enhances work process management and data centralization for 'Migdal-Or'. Our key achievements include:

1. System Implementation: We have fully developed a user-friendly system that streamlines workflow management and centralizes data storage on a unified platform. The system is operational and ready for integration into 'Migdal-Or's daily operations.

2. Genetic Algorithm Optimization: We have refined our genetic algorithm for worker allocation. While it no explicitly identifies bottlenecks, it now provides users with a ranked list of employees for any selected work position. This enhancement allows for more flexible and informed decision-making in workforce allocation.

3. Sensor Integration Framework: We have established a foundational framework for real-time data collection from work stations. While not yet fully implemented due to dependencies on an ongoing project in the Mechanical Engineering department, this framework is prepared for seamless integration with IoT sensors. Once fully operational, it will provide real-time production indicators at various work stations.

4. Data-Driven Decision Support: The system now offers data-driven insights to support management decisions, enhancing current workflows and addressing emerging needs effectively.

5. Scalability and Future-Readiness: Our achievements have positioned the system to be scalable and adaptable to future technological integrations, particularly in anticipation of full IoT sensor implementation.

These accomplishments represent significant progress in optimizing 'Migdal-Or's operations, laying a strong foundation for future enhancements and full IoT integration. The system is poised to significantly improve workforce management and operational efficiency once all components are fully implemented.

6.1 Success Criteria

We note that since the system has not yet been used for a long time, there is no real assessment of whether we have achieved the success criteria.
Defectives Amount: Achieve a 20% or greater decrease in the amount of defectives compared to the baseline. A decrease in the amount of defectives will indicate that the algorithm performs a good scheduling of workers in work positions (in addition to the quantities they produce).
Measurement Method: Measuring the average number of defects today according to data stored in the factory and after the implementation of the system, performing another measurement.

Functioning of Genetic Algorithm: 80% of the suggestions of the algorithm for the placement of employees will be accepted without changing by the managers.
Measurement Method: The system will collect the data about shift scheduling changes and will learn from the changes.

7 Users Evaluation

The system is currently deployed on Vercel for testing and project submission purposes. However, to meet the factory's requirements and obtain an accurate assessment of the project's performance, the final deployment will be conducted on Amazon Web Services (AWS). This transition to AWS will allow for a more realistic evaluation of the system's capabilities in the intended production environment. After presenting the system to key users, we received overwhelmingly positive feedback. They expressed great satisfaction in seeing the progress made and were particularly impressed with the system's overall appearance and user interface. The users' enthusiasm was evident as they explored various features, with special attention given to the worker allocation screen. Upon reviewing the allocation interface, users immediately recognized its potential and shared suggestions for enhancements. They proposed several refinements and additional features they would like to see implemented, demonstrating their engagement with the system and its relevance to their daily operations. These suggestions included requests for more detailed reporting options, customizable views for different user roles, and the ability to perform "what-if" scenarios for workforce planning. The users' proactive approach in providing feedback not only validates the system's core functionality but also highlights areas for potential improvement. Their input has been invaluable in identifying features that could further streamline their workflow and enhance decision-making processes. This collaborative evaluation process has reinforced the importance of user-centered design and iterative development in creating a system that truly meets the needs of its intended users.

8 Future Work

As with any system, there are numerous opportunities for enhancement and expansion of our project. The completion of this initial phase has laid a strong foundation, but also illuminated several promising avenues for future development. These potential improvements and extensions would not only refine the existing functionality but also broaden the system's capabilities to meet evolving needs of factory managers.

1. Enhancement of IIoT Integration: We have an established collaboration with the Mechanical Engineering faculty on their ongoing Industrial Internet of Things (IIoT) project. However, their system is not yet fully implemented in the field. As a result, our current project utilizes simulated data entered manually through Mqttx into the database. A critical next step is to prepare our system for integration with real field controllers once the IIoT infrastructure is deployed in the factory. This will involve:
    - Conducting comprehensive testing and making necessary adjustments to ensure seamless integration with the live IIoT environment.
    - Collaborating closely with the Mechanical Engineering team to align our system with their IIoT implementation timeline and specifications.
2. Scalability for Future IIoT Projects: As the factory plans to develop additional IIoT projects, our system will require adaptations to accommodate these new initiatives. This may involve expanding data processing capabilities, enhancing the user interface, or developing new modules to interface with upcoming IIoT implementations.
3. Genetic Algorithm Improvements: Based on feedback from the factory, the genetic algorithm for worker placement needs refinement. Key enhancements include:
    - Incorporating work order prioritization into the placement algorithm, allowing for more dynamic and responsive workforce allocation.
    - Implementing a feature to adjust worker output percentages. This will enable more flexible staffing solutions, potentially allowing multiple workers to be assigned to a single job, thus mitigating potential unemployment issues during periods of reduced orders.

These improvements will significantly enhance the system's utility, align it more closely with the factory's evolving needs, and leverage the ongoing IIoT research within the university.

References

About aws. (n.d.). https://aws.amazon.com/about-aws/.

Abras, C., Maloney-Krichmar, D., & Preece, J. (2004). User-centered design. In W. Bainbridge (Ed.), *Encyclopedia of human-computer interaction* (Vol. 37, pp. 445–456). Thousand Oaks: Sage Publications.

Amindoust, A., Asadpour, M., & Shirmohammadi, S. (2021). A hybrid genetic algorithm for nurse scheduling problem considering the fatigue factor. *Journal of Healthcare Engineering*, *2021*.

Aws iot core for mqtt. (n.d.). https://aws.amazon.com/iot-core/mqtt/.

Aws pricing. (n.d.). https://aws.amazon.com/pricing/.

Aws products. (n.d.). https://aws.amazon.com/products/.

Aws security. (n.d.). https://aws.amazon.com/security/.

Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)*, *35*(3), 268–308.

Boyes, H., Hallaq, B., Cunningham, J., & Watson, T. (2018). The industrial internet of things (iiot): An analysis framework. *Computers in industry*, *101*, 1–12.

Brucker, P., Burke, E. K., Curtois, T., Qu, R., & Vanden Berghe, G. (2010). A shift sequence based approach for nurse scheduling and a new benchmark dataset. *Journal of Heuristics*, *16*, 559–573.

Burke, E. K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., & Qu, R. (2013). Hyper-heuristics: A survey of the state of the art. *Journal of the Operational Research Society*, *64*, 1695–1724.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley Longman Publishing Co., Inc.

Hivemq - mqtt essentials - all core concepts explained. (n.d.). https://www.hivemq.com/.

Javascript - mdn web docs. (n.d.). https://developer.mozilla.org/en-US/docs/Web/JavaScript.

Johnson, M. W., & Abumeeiz, S. (2023). The limits of inclusion in open access: Accessible access, universal design, and open educational resources. *Journal of Librarianship and Scholarly Communication*, *11*(1).

Lee, I., & Lee, K. (2015). The internet of things (iot): Applications, investments, and challenges for enterprises. *Business horizons*, *58*(4), 431–440.

Li, G., Li, D., & Tang, T. (2023). Bibliometric review of design for digital inclusion. *Sustainability*, *15*(14), 10962.

Liu, T., & Zhang, L. (2021). Apply artificial neural network to solving manpower scheduling problem. In *2021 ieee 4th international conference on big data and artificial intelligence (bdai)* (pp. 58–64).

MDN Contributors. (n.d.-a). *Css: Cascading style sheets.* https://developer.mozilla.org/en-US/docs/Web/CSS.

MDN Contributors. (n.d.-b). *Html: Hypertext markup language.* https://developer.mozilla.org/en-US/docs/Web/HTML.

Migdal or services brochure. (n.d.).

Mineraud, J., Mazhelis, O., Su, X., & Tarkoma, S. (2016). A gap analysis of internet-of-things platforms. *Computer Communications*, *89*, 5–16.

Mongodb. (n.d.). https://www.mongodb.com/basics/get-started.

Nearchou, A. C., & Lagodimos, A. G. (2013). Heuristic solutions for the economic manpower shift planning problem. *European Journal of Industrial Engineering*, *7*(6), 657–686.

Node.js - about node js. (n.d.). https://nodejs.org/en/about/.

Norman, D. (2013). *The design of everyday things: Revised and expanded edition.* Basic Books.

Quincozes, S., Emilio, T., & Kazienko, J. (2019). Mqtt protocol: fundamentals, tools and future directions. *IEEE Latin America Transactions*, *17*(09), 1439–1448.

Ravindran, A. R. (Ed.). (2007). *Operations research and management science handbook* (1st Edition ed.). CRC Press. doi: 10.1201/9781420009712

React – a javascript library for building user interfaces. (n.d.). https://legacy.reactjs.org/.

Rodríguez-Pérez, G., Nadri, R., & Nagappan, M. (2021). Perceived diversity in software engineering: a systematic literature review. *Empirical Software Engineering*, *26*, 1–38.

Typescript - mdn web docs glossary: Definitions of web-related terms. (n.d.). https://developer.mozilla.org/en-US/docs/Glossary/TypeScript.

Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., & De Boeck, L. (2013). Personnel scheduling: A literature review. *European journal of operational research*, *226*(3), 367–385.

Voß, S. (2001). Meta-heuristics: The state of the art. In *Local search for planning and scheduling (lsps 2000)* (Vol. 2148, pp. 1–23). Springer. doi: 10.1007/3-540-45612-0_1

Vredenburg, K., Mao, J.-Y., Smith, P. W., & Carey, T. (2002). A survey of user-centered design practice. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 471–478).

Wang, J., Zhang, W., Shi, Y., Duan, S., & Liu, J. (2018). Industrial big data analytics: challenges, methodologies, and applications. *arXiv preprint arXiv:1807.01016*.

9 AI Tools and Prompts

We used: Gemini, Copilot, Claude.ai and chatGPT. prompts that we used:

- Write me a literature review on IIoT. include references.
- give some options of names for my final project in college. The project is about building an information system for a factory that employs visually impaired workers, it will include inclusive design, IIoT and the use of a genetic algorithm to place workers
- Explain to me about a genetic algorithm for the placement of employees and give me a number of articles that talk about the subject

10 Appendices

10.1 First Interview with Gabi ('Migdal-Or' manager)

Date: February 1, 2024.
Participants:

- Gabi Cohen - plant manager and digital accessibility field.
- Naomi Onklos-Spiegel - project moderator.
- Paz Fayer - Information Systems Engineering student.
- Eden Bar - Information Systems Engineering student.

"Migdal Or" is an occupational and social framework for the visually impaired, there are also rehabilitation centers. Sometimes the recipients of the service also go to the free market. The factory tries to export an environment as similar as possible to the open market. The factory has 32 active customers this year and they produce as subcontractors in the field of electronics and plastics. 75The factory works with large companies such as Keter, Strauss, Flextronics, Mitronics as well as startups. Most of the workers cannot see at all and some are visually impaired, there are workers who deal with additional disabilities or illnesses. Developments to make the production floor accessible to employees: Talking and counting weight - a weight that calculates how many screws are on it. Schlocker nozzle for water flask - checking that there are no leaks after assembly. Checking by connecting to the system and checking if there are bubbles. Problems: need a sighted person, it takes a lot of time, need to dry the nozzle. Solution: One of the factory workers developed a dry test by air pressure (faster, suitable for the blind because it makes a sound).
The system we will develop is for managers and occupational instructors (each department has an instructor) who need to receive the information. Today, the data collection process is done manually and only on larger products.

Orders - navigation to which department the order should go, for example if there is an order for kits of 'Keter' - the system will receive from the order the data regarding how much items should be weighed for a kit and deliver it to department managers. Collecting data about the workers - the production, how they work, how many are defective, notification of drastic changes in workers' productivity, how many iterations it takes for the worker to enter the correct amount against the order ,check how many ready kits he finished. The employee's work rate is according to the good products that he finished.

For Shlukers - there is a threshold level of 0.5% defective, it is desirable that the system alerts if there are more defective than the standard.
The software should be accessible to the blind (HTML 5 standard)
Employee data is collected manually under the platform "Employees Net"

10.2 Second Interview with Gabi ('Migdal-Or' manager)

Date: February 29, 2024.
Participants:

- Gabi Cohen - plant manager and digital accessibility field.
- Paz Fayer - Information Systems Engineering student.

- Eden Bar - Information Systems Engineering student.

Explanation of the data table sent to us: In the file there is a list of jobs that employees from several departments can perform. If there is a number in the cell, it means the employee can perform this work, if the cell is empty - he cannot perform it. The number represents the daily average output of the employee on this work. Notes:

- A working day is 5 hours
- The 2 leftmost columns are the employee's assignment for a certain day
- Usually an employee is assigned to one position per day - indicated by 'Other' entry in the 2 columns, some employees move between 2 positions during the day - the names of the positions will be listed.
- The customer pays by product, he is not aware of the internal division into workstations within the factory. The reason for the division is so that all service recipients have employment.
- Employees are paid according to work - there is a rate for each work per standard unit.

Question: Can an employee move between departments or is he only associated with a specific department?
Answer: During the assignment, if a certain position is needed, priority will be given to employees from within the department, but if they have already been assigned to other jobs, employees from other departments who are qualified to work in this position can also be assigned.

There is no one-size-fits-all mapping of positions and departments - a position can be relevant to several departments.

Question: How do you check according to what an item is considered improper (due to assembly or arrived improperly in advance)?
Answer: If there is a defective item (for example, a Shluker), we check whether the problem is in the assembly, if so, we return it to be fixed on the relevant station, if the problem is in the product itself, we return it to the customer (happens a lot more). We don't want the workers to work fast but the products won't be correct, it's better that they work slower and with better quality.
Question: Do the employees go through some kind of training according to which the assignment is made?
Answer: There is management of training, a record of what each employee can do (the department managers know by heart), only those who are qualified to do a certain job do it.
Question: Do the employees go through some kind of training according to which the assignment is made?
Answer: There is management of workers qualifications, a record of what each employee can do (the department managers know by heart), only those who are qualified to do a certain job do it.
Question: Is the training something changing? Are koshers added? Are placement positions added?
Answer: We are constantly trying to introduce new products, if there is someone who cannot work in a certain position, then we do try to train him so that he can take part in the production process. The positions are also dynamic depending on the needs and orders that are available. There is a desire for each employee to have as much "training" as possible so that he can take part in multiple positions in the production process. We do not want to get to a situation where there is a person without a job or a job without a person who can perform it.

The placement algorithm will not be according to who is the best in each position, it should be taken into account that every employee needs to have a placement, there is no problem to be less efficient in order for every employee to have employment.

There are mechanisms to encourage employees to learn new things so they could work in new positions. It is taken into account that an employee who starts working in a new position, the production will be less good and there will be more malfunctions than usual. Each job guide knows in depth the recipients of the service under him so that he knows who can work in which position and whether it is possible to try to integrate new employees in different positions (and also when this can be done depending on their personal/mental state).

Question: Is it possible for a guide to fill in the production for employees who are not in his department?
Answer: Yes, if there is a job guide who is on vacation and must close the month, then he can ask another job guide to do so.
Question: Is what the plant manager will be able to do in the system what the instructors will be able to do in the system? What is the division of roles?
Answer: For now the whole team will be able to do the actions, yes there should be a permissions mechanism that can be accommodated in the future. Only the assignment of an employee to a department, and the introduction of a service recipient into this system is the responsibility of the factory manager.
Question: Is there a preferred way in which you would like to see the data analyzed (graphs, tables...)?
Answer: I prefer to start with tables that can be exported to excel. Filtering will be by - department, people, jobs. Even if initially it will be all the data and he will be able to filter in Excel.

10.3 Third Interview with Gabi and Shmuel(department manager)

Date: February 1, 2024.
Participants:

- Gabi Cohen - plant manager and digital accessibility field.
- Shmuel Kalbo- Occupational guide at the factory
- Paz Fayer - Information Systems Engineering student.
- Eden Bar - Information Systems Engineering student.

The purpose of the meeting is to view the screens and coordinate expectations between the project developers and the clients.
First, the interviewer (Eden) shows the customers (Gabi and Shmuel) the prototype of the screens.

Gabi: What do the numbers at the top of the home page mean?
Eden: The number of positions we have in total, each department will show the corresponding numbers.
Gabi: There are some things that we do in several departments, so the assignment should be according to which employees are relevant and what are the suppliers of each employee at each station.
Eden: Do you want Shmuel to be able to assign employees from all departments? Even for positions that are not in his department?
Gabi: I see it this way: there is a work order, and the system checks.
Shmuel: I want to see in one work order what I have in total, regardless of which department.
Eden: Do you do daily or weekly assignments?
Gabi: We make a weekly plan, but it can be updated every day, if a certain person didn't come for any other reason, there should be a possibility of change on a daily basis. But the plan is to plan every Thursday for the following week.
Eden: So this is how we see it, choose a position and the position details will be opened for you and what will appear there are the positions that, in terms of the algorithm, will give an optimal placement. According to the principles that all positions will be staffed, that no employee will lack work and also that the output will be as high as possible. You choose a position, then a date and then choose a placement.
Shmuel: I would first like to choose a type of work and not a position. Project type and know what is happening with that project. For example, I chose "Shluker" and all working stations related to "Shluker" opening.
Eden: It is important for me to make it clear that whatever is the recommendation for placement, you choose and can change the choice of who will be placed where.
Gabi: Yes, I understand, does this interface allow what you say?
Eden: This is just a prototype of the screens, later it will be developed.
Gabi: How do I enter a work order here?
Eden: This option is not available yet, we will redesign the screens and open according to this conversation.