SDD document for final project:

AI-Scheduler

Members:

Amit Bibi - 203262647

Moran Oshia - 313292633

Alex Chagan - 206262123

Moderator:

Dr. Elad Horev

1.1 Purpose

In this document we are going to describe how our software product is going to work (architecture of the software project), how the server side works (backend), a little bit on the algorithm and how our product is going to look (software designer) the user interface and detailed feature specifications of smaller pieces of the design.

In essence, this software design document (SDD) will explain how our software product will be built, its set of technical requirements “what” and “how” it's going to work.

You will find here an extended explanation describing data design, architecture design, interface design and procedural design.

1.2 Scope

Our project is to create a new program an intelligent system that will compute a work schedule for workers based on their roles, preferences, restrictions, etc.

Our goals are to program a user-friendly machine learning system to make schedule planning easier on employees and their managers which can manage projects, tasks, notes and take into consideration their deadlines and requests.

1.3 Overview

This SDD contains:

1.1 Purpose
1.2 Scope
1.3 Overview
1.4 Reference Material
1.5 Definitions and Acronyms
2.1 System Overview
3.1 Architectural Design
3.2 Decomposition Description
3.3 Design Rationale
4.1 Data Description and Dictionary
5.1 Component Design (example)
6.1 Overview of User Interface
6.2 Screen Images (Just functional)
6.3 Screen Objects and Actions
7.1 Requirements Matrix

1.4 Reference Material

http://ariel.zone/Content/sdd_template.pdf
https://senior.ceng.metu.edu.tr/2015/musinspro/Musins-Pro_SRS.pdf

1.5 Definitions and Acronyms

SRS - System Requirements Specification
SDD - System Design Document
DB - Database
AI - Artificial Intelligence
ML - Machine learning

2.1 System Overview

Our AI Scheduler will provide a platform to insert shifts, offer shifts based on previous ones.

It will also provide machine learning algorithms to build a weekly/monthly schedule for the workers.

In addition, AI Scheduler will provide a platform for managing projects, tasks and notes between the employers and the employees.

3.1 Architectural Design

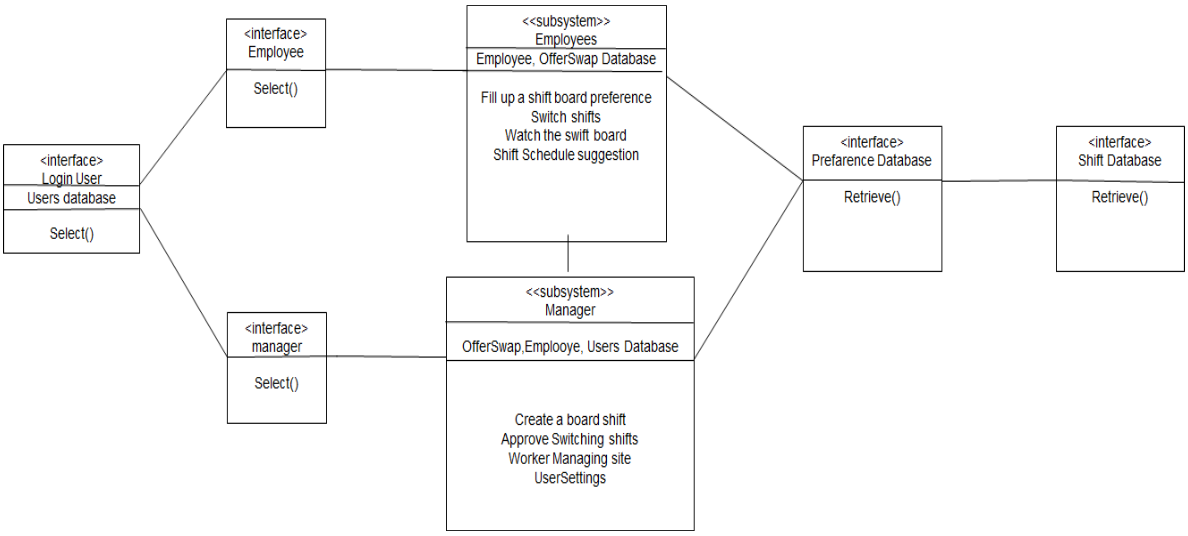


Figure 1 - Top-Level Architecture Diagram

In this section, the information about system architecture will be given. The architectural design consists of 2 interfaces, the first one is for the employees and the second is for the managers as you can see in the figure above of top-level Architecture Design, they all have sub-systems. The manager interface is pulling the data from the employee's interface. In both there are forms, in the employees there are: Fill up a shift board preference, Switch shifts, Watch the swift board and more. The manager side connected to the employees' interface in terms of data flow and use for his form: Create a board shifts, Approve switch shifts between workers and more (detailed in the next section).

3.2 Decomposition Description

This section describes the main components of the project. As stated previously, two components are present namely: Employees and manager.

There will be detailed information about these components in the following sections.

3.2.1 Manager

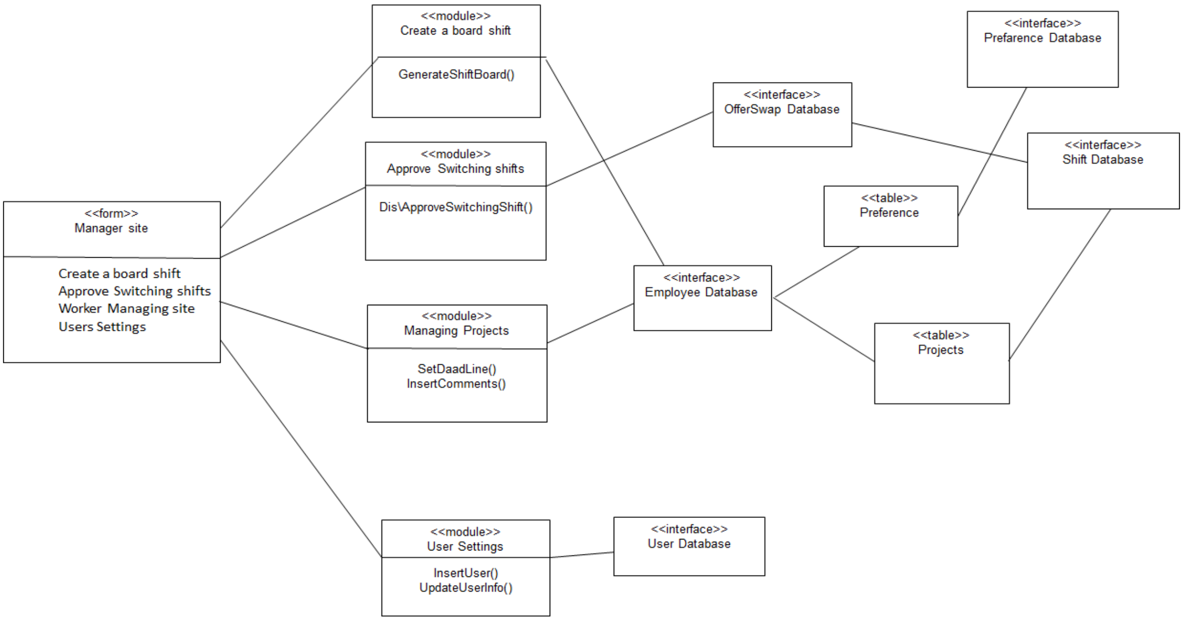


Figure 2 - Manager Architectural Design

Name: Manager
Type: Subsystem
Description: This is the primary entrance to the system for a Manager. Only users with managing permissions can enter here. Manager able to: Create shift board, Create personal projects to employee and approve\disapprove shifts switching.
Attributes: Manager database (see Section 4.1)
Operations (detailed below):

Create a board shift
Approve Switching shifts
Worker Managing site
Insert\Update Users

3.2.1.1 Name: Create a board shift Form

Name: CreateBoard.
Type: Access Form.
Node: Manager.
Description: This is used to create a weekly shift board, get the users preference, and generate shift boards based on them.
Attributes: A weekly board, Users Preference.
Resources: Access to Preference and shift Database

Operations:

Name: GenerateShiftBoard()
Arguments: User’s Preferences.
Returns: shift board.
Pre-condition: Form is active window.
Exceptions: May be abandoned at any time.
Flow of Events:

1. Manager will get the employees preference and generate a board shift based on an algorithm user: min\max shifts per week, employment laws, etc.
2. The algorithm takes the preference of all the employees and creates a new shift board and enter it to Shift DB.

3.2.1.2 Name: Approve Switching shifts

Name: ApproveSwaps.
Type: Access Form.
Node: Manager.
Description: This is used to dis\approve swap or offer shifts by the manager.
Attributes: A swap or offer details.
Resources: Access to OfferSwap DB and shift Database.

Operations:

Name: Dis\ApproveSwitchingShift()
Arguments: Swap\offer details.
Returns: Boolean
Pre-condition: Form is active window.
Exceptions: May be abandoned at any time.
Flow of Events:

1. The manager will get a swap\offer details and dis\app it
2. If the manager approves, it will automatically change in the current shift board DB.

3.2.1.3 Name: Worker Managing site

Name: MngEmp.
Type: Access Form.
Node: Manager.
Description: This is used to update details in personal projects.
Attributes: A weekly board, Users Preference.
Resources: Access to Employees Database.

Operations:

Name: SetDeadLine()
Arguments: None.
Returns: None.
Pre-condition: Form is active window.
Exceptions: May be abandoned at any time.
Flow of Events:

1. The system calls the operation on the project side in the manager interface.
2. Manager will be able to set or update deadlines, etc.
3. The details will automatically change in the project table in Employees DB.

Name: InsertComments ()
Arguments: None.
Returns: None.
Pre-condition: Form is active window.
Exceptions: May be abandoned at any time.
Flow of Events:

1. The system calls the operation on the project side in the manager interface.
2. Managers will be able to insert comments to specific projects of employees.
3. The details will automatically change in the project table in Employees DB.

3.2.1.4 Name: User Settings
Name: UsersSettings.
Type: Access Form.
Node: Manager.
Description: This is used to update and insert users.
Attributes: Users details.
Resources: Access to Users Database.

Operations:
Name: InsertUser ()
Arguments: None.
Returns: None.
Description: This is used to insert new users.
Pre-condition: Form is active window
Exceptions: May be abandoned at any time.
Flow of Events:

1. Manager will insert new users.
2. User information will insert to Users DB.

Name: InsertComments ()
Arguments: None.
Returns: None.
Description: This is used to update user information.
Pre-condition: Form is active window.
Exceptions: May be abandoned at any time.
Flow of Events:

1. Manager will update user information.
2. User information will update in Users DB.

3.2.2 Employees

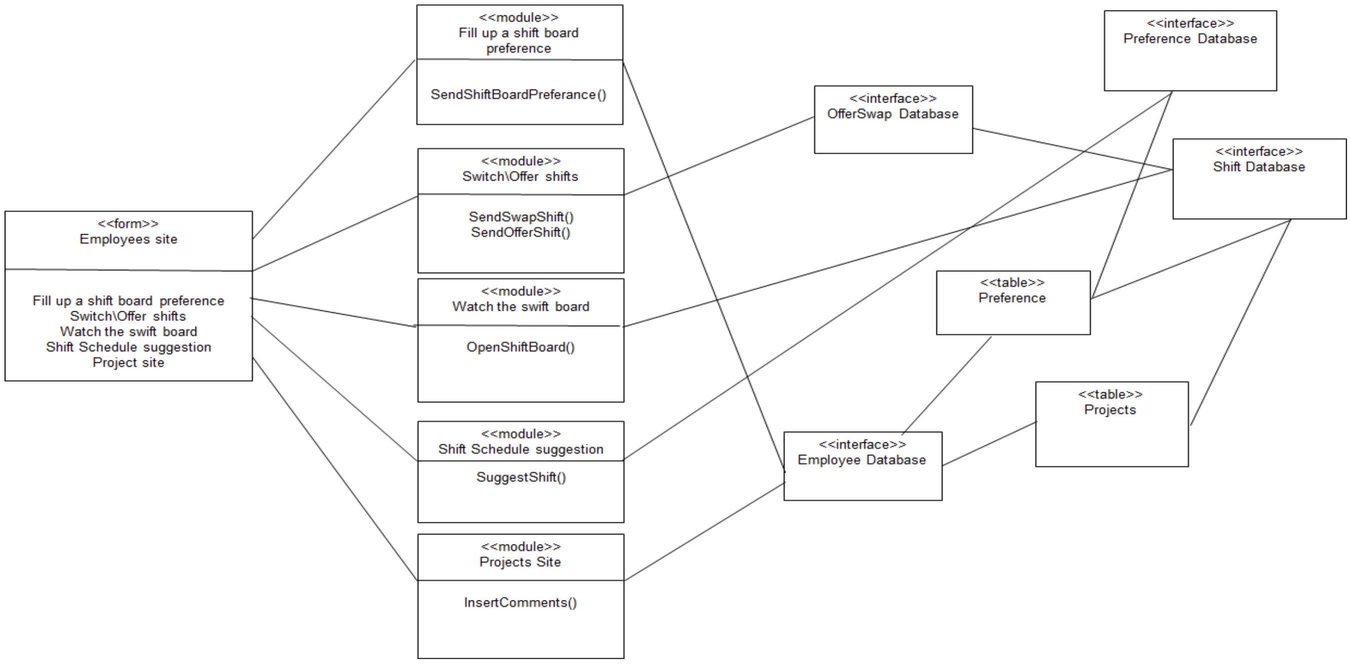


Figure 3 - Employees Architectural Design

Name: Employees
Type: Subsystem
Description: This is the primary entrance to the system for an Employee. Employee able to: Fill up a shift board preference, Switch\Offer shifts, Watch the swift board, Shift Schedule suggestion, Project site.
Attributes: Manager database (see Section 4.1)
Operations (detailed below):

Fill up a shift board preference
Switch\Offer shifts
Watch the swift board
Shift Schedule suggestion
Project site

3.2.2.1 Name: Fill up a shift board preference

Name: FillBoard.
Type: Access Form.
Node: Employee.
Description: This is used to fill up a weekly shift board preference.
Attributes: A weekly board shift, Users Preference.
Resources: Access to Employees and Preference Database

Operations:

Name: SendShiftBoardPreferance ()
Arguments: User's Preferences.
Returns: shift board.
Pre-condition: Form is active window.
Exceptions: May be abandoned at any time.
Flow of Events:

1. The employee will enter his preference on his board shift.
2. The preference will enter the employee and Preference DB.

3.2.1.2 Name: Switch\Offer shifts

Name: Sendswapsoffer.
Type: Access Form.
Node: Employees.
Description: This is used to send swap or offer shifts to other employees.
Attributes: A swap or offer details.
Resources: Access to OfferSwap DB and shift Database.

Operations:

Name: SendSwapShift ()
Arguments: Swap details.
Returns: Boolean.
Pre-condition: Form is active window.
Exceptions: May be abandoned at any time.
Flow of Events:

1. The employee will enter a swap offer to another employee with details of: Date, Type of the shift, Employee.
2. The swap offer was sent to the employee to approve.
3. After the employee approves the swap offer is sent to the manager to finally approve.

Name: SendOfferShift ()
Arguments: offer details.
Returns: Boolean.
Pre-condition: Form is active window.
Exceptions: May be abandoned at any time.
Flow of Events:

1. The employee will enter an offer to another employee with details of: Date, Type of the shift, Employee.
2. The offer was sent to the employee to approve.
3. After the employee approves the offer is sent to the manager to final approve.

3.2.1.3 Name: Watch the swift board

Name: BoardShift
Type: Access Form.
Node: Employees.
Description: This is used to watch the weekly shift board.
Attributes: A swap or offer details.
Resources: Access to shift Database.

Operations:

Name: OpenShiftBoard ()
Arguments: shiftboard details.
Returns: Boolean.
Pre-condition: Form is active window.
Exceptions: May be abandoned at any time.
Flow of Events:

1. The operation will pull from the Shift DB.
2. The shift board will be present in the GUI.

3.2.1.4 Name: Switch\Offer shifts

Name: Sendswapsoffer.
Type: Access Form.
Node: Employees.
Description: This is used to suggest employee board shift based on his last preference of shift board he sent.
Attributes: A swap or offer details.
Resources: Access to Preference and shift Database.

Operations:

Name: SuggestShift ()
Arguments: preference of shift board.
Returns: preference of board shift.
Pre-condition: Form is active window.
Exceptions: May be abandoned at any time.
Flow of Events:

1. The operation will use the preference of board shift the employee entered in the past.
2. With ML Algorithm the operation will learn the cyclical's shift the employee chose, and the systems will automatically suggest a shift board.
3. The employee can dis\approve the suggestion, if approved it will automatically enter on his weekly board shift.

3.2.1.5 Name: Project site

Name: ProjectEmp.
Type: Access Form.
Node: Employees.
Description: This is used to update details in personal projects on the employee side.
Attributes: Project details.
Resources: Access to Employees Database.
Operations:
Name: InsertComments ()
Arguments: None.
Returns: None.
Pre-condition: Form is active window.
Exceptions: May be abandoned at any time.
Flow of Events:

1. The system calls the operation on the project side in the manager interface.
2. Employees will be able to insert comments to specific projects of employees.
3. The details will automatically change in the project table in Employees DB.

3.3 Design Rationale

We decided to split the application to manager and workers and to give other permissions for each.

Manager side is going to have all worker pages and more pages that are related to managing/controlling users accounts, projects, changing shifts and making the schedule for the week.

We decided to make 2 tables for workers preference, one for the ML that will suggest shifts for the workers and one that will hold the shifts just for the week, so the build of the schedule is going to be fast as possible.

4.1 Data Description

We chose to use firebase (Google) as our DB for this project.

We had a lot of options for organizing our data and we decided that this is the best alternative.

we'll make one database with a couple of tables inside.

The first table is “Users” which is mainly for saving all the information for the authentication of the user and general info on the user as: user_id, first_name, last_name, job_title, permission and on.

The second table is “Employees” which is going to save in a new session for each worker his previous preferences of shifts so our algorithm will know what to offer for next week/month shifts. In addition, it is going to hold a session for projects that each worker has.

The third table is “Swap/Offer” which is going to save the information of the swap/offer shifts between the employees.

The fourth table is “Preferences” which is going to save the information of the workers previous shifts by weeks for the algorithm to read the information fast as possible.

The fifth table is “ShiftBoard” which is the most important one, for the reason our algorithm is taking all the information to build the schedules using the ML.

This table is going to be the biggest and with two sessions one that keeps all the published schedules in order to display them on the shift schedule on the home page and the other for the ML tool.

4.1.1 User

user_id	**first_name**	**last_name**	**email**	**password**	**title_job**	**permission**
111111	Elad	Horev	amit@..com	******	worker	E
111112	Alex	Oshia BB	alex@..com	******	manager	M

4.1.2 Employees

user_id	**prefernces_id**	**projects_id**
111111	000	100
111112	001	101

4.1.2.1 Sub-Table preferences - weeks

prefernces_id	**week_number**
000	48/2020
001	49/2020

4.1.2.2 Sub-Table projects - project

projects_id	project_name	due_date	note	status
100	Web-Application	19/8/20	“Good Job!”	Done
101	Automation	6/6/20	null	Waiting for approval

4.1.2.3 Sub-Table weeks - days

week_number	week_day	shift_time
48/2020	Sunday	morning, night
48/2020	Monday	none
48/2020	Tuesday	evening

4.1.3 Swap/Offer

request_id	sender_id	receive_id	shift_id	offer_swap	receive_status	manager_status
300	111111	111112	84941651	O	V	V
301	111111	111112	68849816	S	X	none

4.1.4 Preferences table is the same as the “Preferences” in Employees’ table.

4.1.5 ShiftBoard

week_number	published
48/2020	V
49/2020	X

4.1.5.1 Sub-Table ShiftBoard - shifts

week_number	week_day	morning_shift	evening_shift	night_shift
48/2020	Sunday	Amit, Alex	Michael	Moran
48/2020	Monday	Amit, Alex	Michael	Moran
48/2020	Tuesday	Alex, Moran	Michael	Amit

5.1 Component Design

```
function create_board_shift () {
        list <- load preferences from db (for specific week)
schedule = null
while (schedule!= null)
        r_employees <- list.request_shift()
        schedule.shift <-  check_law_min(schedule, r_employees)
        schedule.next()
list.next()
update_db (schedule)
return schedule;
}
function check_law_min(sch,r_employees){
        for i in r_s
                temp <- min_algo(r_employees)
                if (temp != sch.previous().shift)
                        return temp;
        return null;
}
function approve switching shift{
        list <- get_request_from_db()
        display_request()
        ans <- // waiting for input
        if (ans == “approve”)
                return True;
        else
                return False;
}

function insert_user {

name <- text_box.text()
user <- generate_username_and_password(name)
update_db(users)
}
```

6.1 Overview of User Interface

Product will have a graphical user interface which facilitates user's utilization of the program. GUI will mainly have five screens as follows.

6.1.1 Authentication (Login) Screen

Trying to use the system, users will be demanded to get authorized from the system. In order to display the interface corresponding to his/her role (admin or observer), users must be authenticated from the system. In this screen user will write his/her username and password then login the system. Depending on the user type, the next phase will be decided.

6.1.2 Employer Main Screen

If the user has employer authentication, the user will see this screen. In this interface employers will have a number of different options. Also, a table of the shifts for the current week will be displayed in the center. The table will contain columns that represent the days of the week and rows that represent the time of each shift, every square will be filled with names of employees in that shift and their profession. In addition, there will be arrow buttons on top of the table so the employer can switch between weeks. Other options will include stuff like: A button to access the personal info of the employees (6.1.4), a button to add a new employee (6.1.6), a button to access the Employer swap/offer Screen(6.1.5) and a
button to access the "makeshift" table screen (6.1.3).

6.1.3 Employer Makeshift Table Screen

This screen will be accessible from the Employer Main Screen (6.1.2)
In this screen, the employer can automatically build a shift schedule with the help of an algorithm by clicking a button.
The table will be fully customizable so that the employer can delete or add shifts manually and change the times of each shift.

6.1.4 Employee Info Screen (for Employers)

This screen will be accessible from the Employer Main Screen (6.1.2) and will contain a scrollable list with all the names of the employees. When the employer chooses a name, it will display the employee's info screen (refer to 6.1.5). This screen will also allow to add new employees and delete current ones.

6.1.5 Employer Offer/Swap Screen

This screen will be accessible from the Employer Main Screen (6.1.2). This screen will display the offer/swap requests of the employees and the employer will have the option to accept or decline them.

6.1.6 Add Employee Screen

This screen will be accessible from the Employer Main Screen (6.1.2) and it’s very simple. The employer can add a new employee to the database by entering a name and generating a username and a password.

6.1.7 Employee Main Screen

If the user has employee authentication, the user will see this screen. In this interface employees will have a number of different options. Also, a table of the shifts for the current week will be displayed in the center. The table will contain columns that represent the days of the week and rows that represent the time of each shift, every square will be filled with names of employees in that shift and their profession. Other options will include stuff like accessing the Shift Request Screen (6.1.8), the Employee Info Screen (6.1.9)
and the Employee Swap/Offer Request Screen (6.1.10). In addition, there will be a section with all the requests from other employees for swapping a shift.

6.1.8 Employee Shift Request Screen

This screen will be accessible from the Employee Main Screen (6.1.7). A table of the shifts for the next week will be displayed in the center. The table will contain columns that represent the days of the week and rows that represent the time of each shift (Morning/Evening/Night). The employee will be able to change preferences of certain shifts by clicking on the squares. There will be a button for sending the shift request.

6.1.9 Employee Info Screen

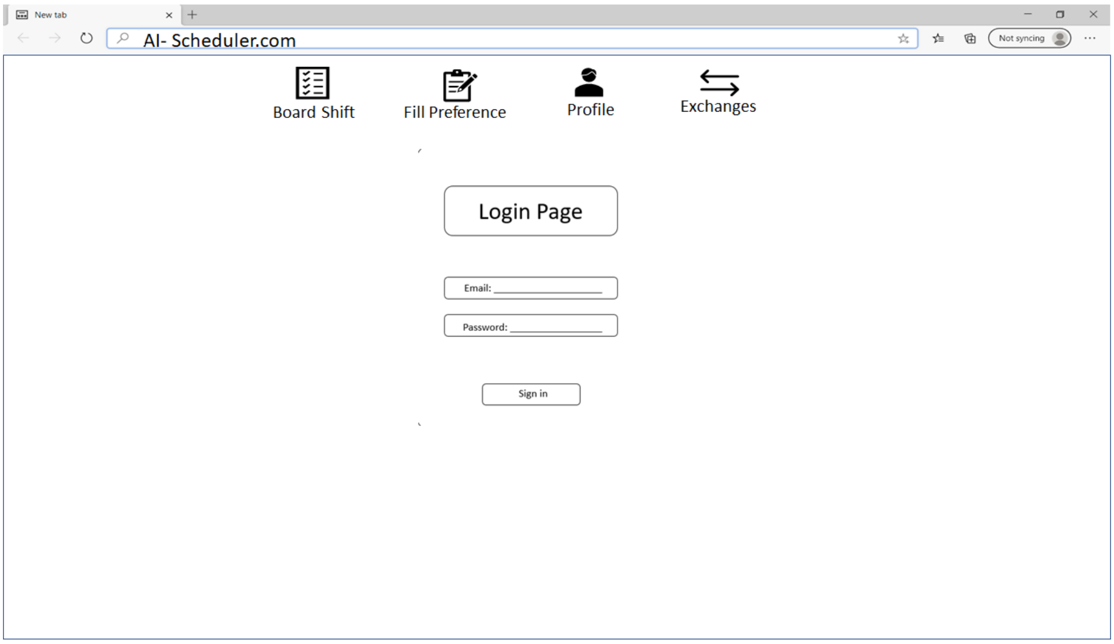
This screen will be accessible from the Employee Main Screen (6.1.7) and will contain info about all the shifts of the specific employee throughout the week/month. In addition, the employee can leave notes for every shift for the employer to read

6.1.10 Employee Swap/Offer Request Screen

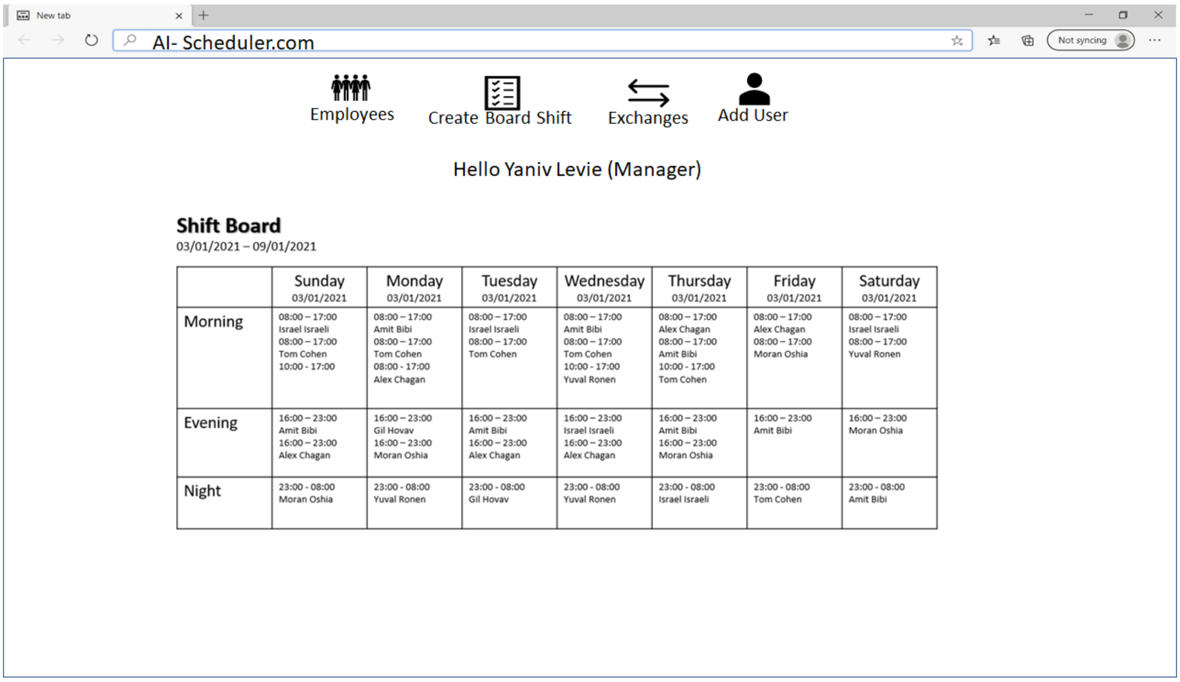
This screen will be accessible from the Employee Main Screen (6.1.7).
The employee will be able to request a shift swap by choosing the shift and the other employee he wants to switch with. In addition, he can offer his shift by requesting a swap with all the other employees at once.

6.2 Screen Images

6.2.1 Authentication (Login) Screen



6.2.2 Employer Main Screen

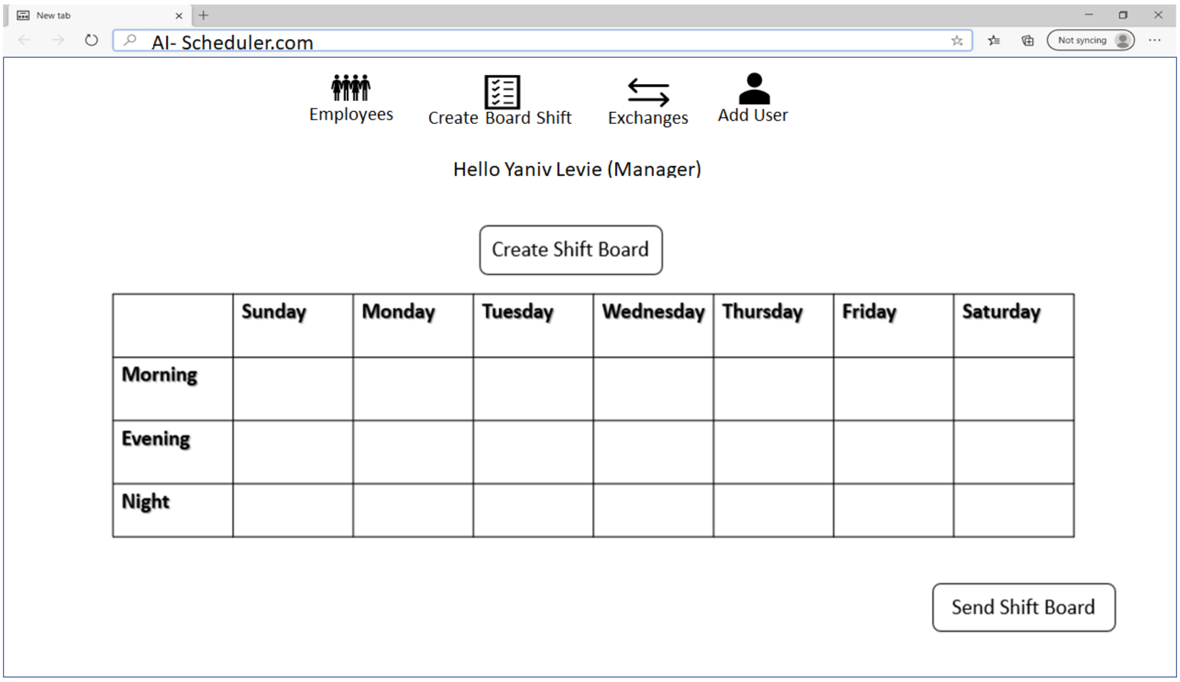


Hello Yaniv Levie (Manager)

Shift Board
03/01/2021 – 09/01/2021

	Sunday 03/01/2021	Monday 03/01/2021	Tuesday 03/01/2021	Wednesday 03/01/2021	Thursday 03/01/2021	Friday 03/01/2021	Saturday 03/01/2021
Morning	08:00 – 17:00 Israel Israeli 08:00 – 17:00 Tom Cohen 10:00 - 17:00	08:00 – 17:00 Amit Bibi 08:00 – 17:00 Tom Cohen 08:00 - 17:00 Alex Chagan	08:00 – 17:00 Israel Israeli 08:00 – 17:00 Tom Cohen	08:00 – 17:00 Amit Bibi 08:00 – 17:00 Tom Cohen 10:00 - 17:00 Yuval Ronen	08:00 – 17:00 Alex Chagan 08:00 – 17:00 Amit Bibi 10:00 - 17:00 Tom Cohen	08:00 – 17:00 Alex Chagan 08:00 – 17:00 Moran Oshia	08:00 – 17:00 Israel Israeli 08:00 – 17:00 Yuval Ronen
Evening	16:00 – 23:00 Amit Bibi 16:00 – 23:00 Alex Chagan	16:00 – 23:00 Gil Hovav 16:00 – 23:00 Moran Oshia	16:00 – 23:00 Amit Bibi 16:00 – 23:00 Alex Chagan	16:00 – 23:00 Israel Israeli 16:00 – 23:00 Alex Chagan	16:00 – 23:00 Amit Bibi 16:00 – 23:00 Moran Oshia	16:00 – 23:00 Amit Bibi	16:00 – 23:00 Moran Oshia
Night	23:00 - 08:00 Moran Oshia	23:00 - 08:00 Yuval Ronen	23:00 - 08:00 Gil Hovav	23:00 - 08:00 Yuval Ronen	23:00 - 08:00 Israel Israeli	23:00 - 08:00 Tom Cohen	23:00 - 08:00 Amit Bibi

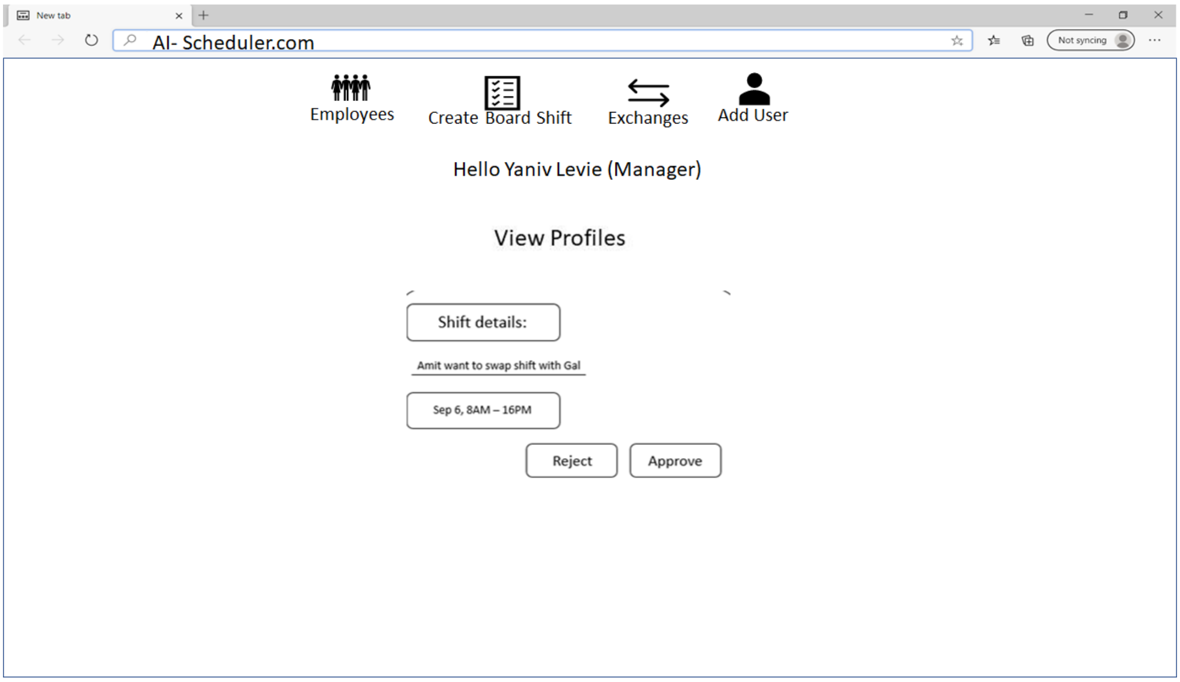
6.2.3 Employer Create Shift Board Screen



Hello Yaniv Levie (Manager)

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Morning							
Evening							
Night							

6.2.4 Employer Offer/Swap Screen

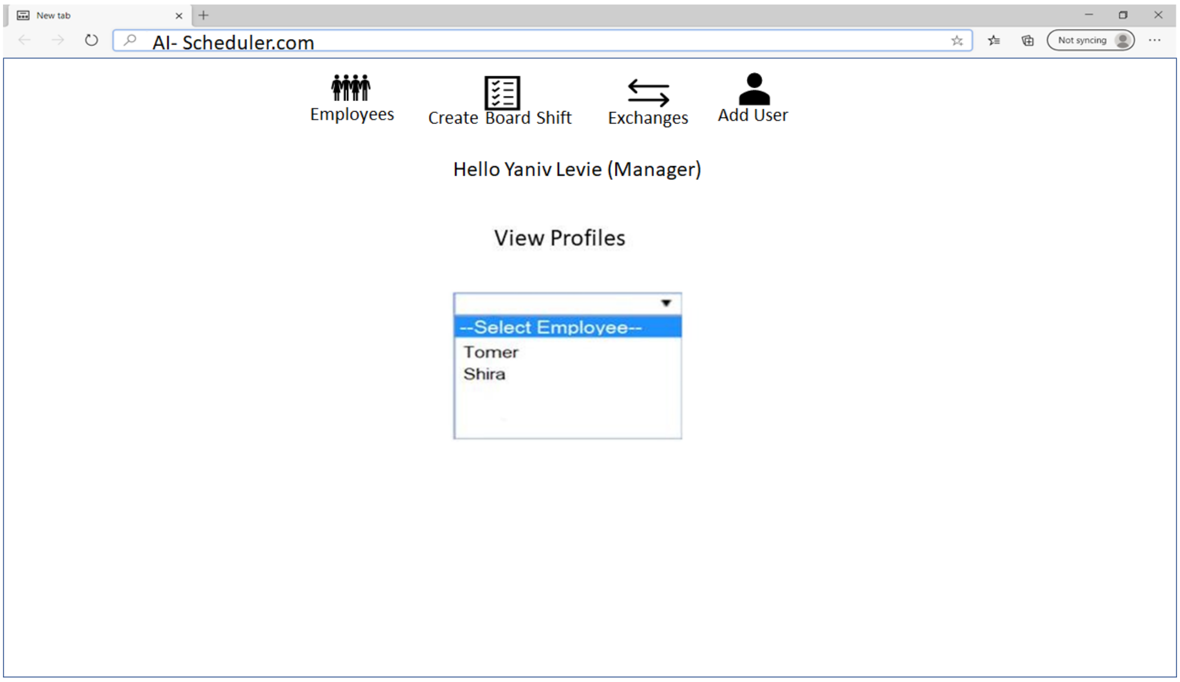


Hello Yaniv Levie (Manager)

Amit want to swap shift with Gal

Sep 6, 8AM – 16PM

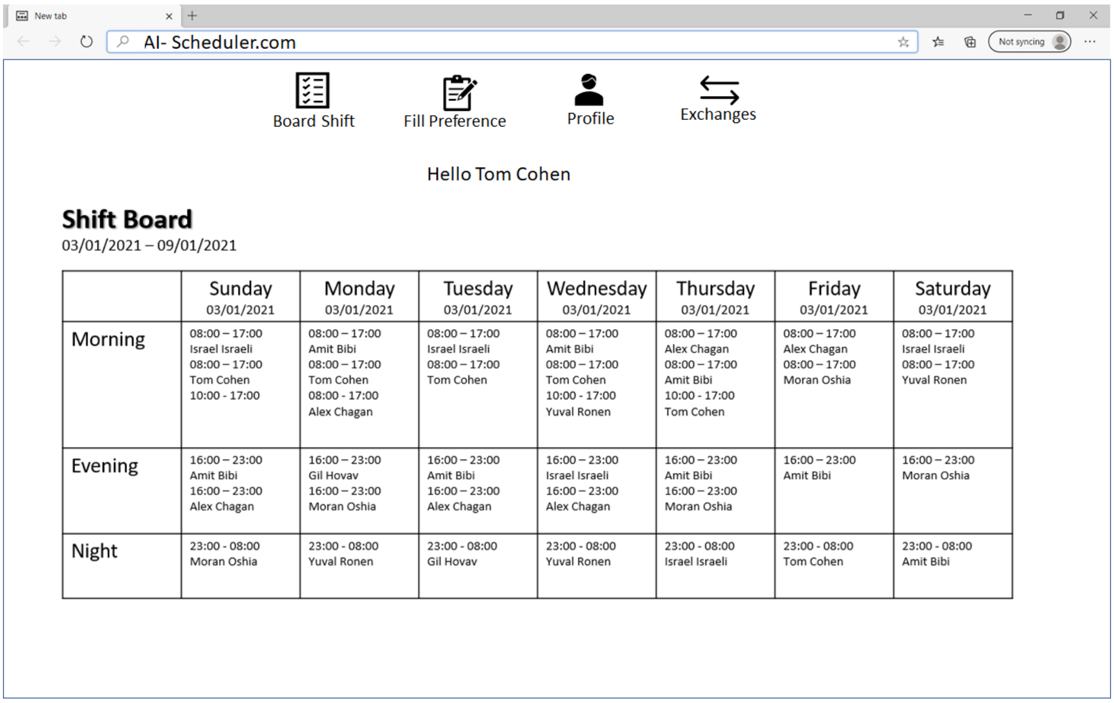
6.2.5 View Employees Profiles Screen



Hello Yaniv Levie (Manager)

View Profiles

6.2.6 Employee Shift Board Screen



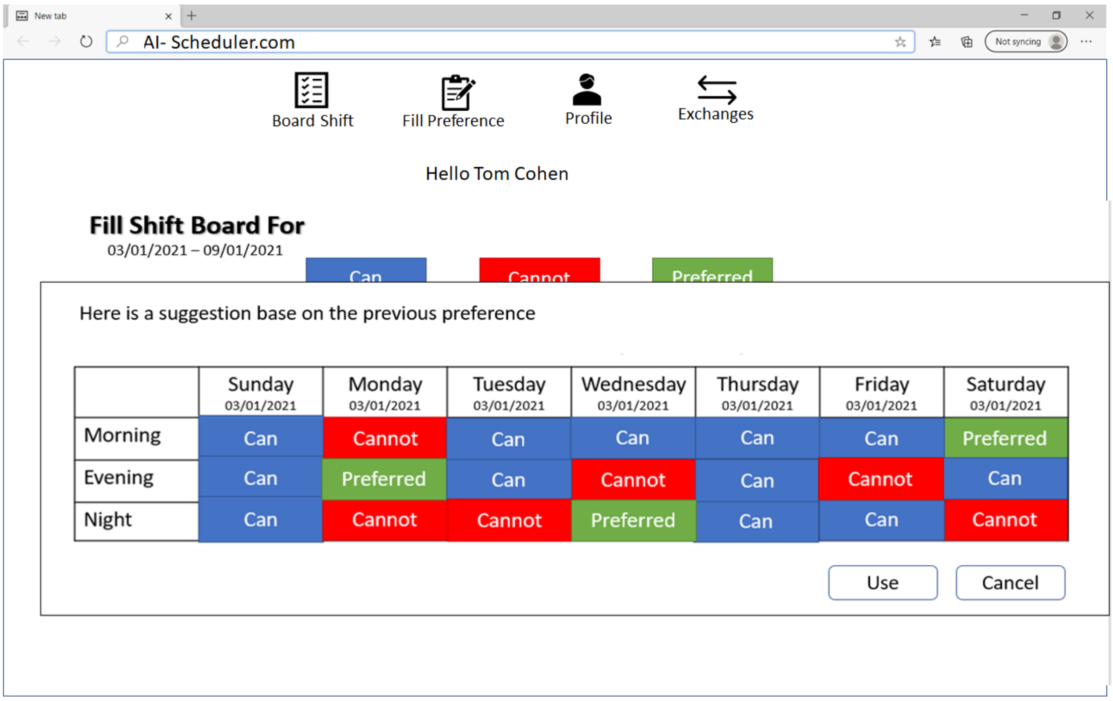
Hello Tom Cohen

Shift Board

03/01/2021 – 09/01/2021

	Sunday 03/01/2021	Monday 03/01/2021	Tuesday 03/01/2021	Wednesday 03/01/2021	Thursday 03/01/2021	Friday 03/01/2021	Saturday 03/01/2021
Morning	08:00 – 17:00 Israel Israeli 08:00 – 17:00 Tom Cohen 10:00 - 17:00	08:00 – 17:00 Amit Bibi 08:00 – 17:00 Tom Cohen 08:00 - 17:00 Alex Chagan	08:00 – 17:00 Israel Israeli 08:00 – 17:00 Tom Cohen	08:00 – 17:00 Amit Bibi 08:00 – 17:00 Tom Cohen 10:00 - 17:00 Yuval Ronen	08:00 – 17:00 Alex Chagan 08:00 – 17:00 Amit Bibi 10:00 - 17:00 Tom Cohen	08:00 – 17:00 Alex Chagan 08:00 – 17:00 Moran Oshia	08:00 – 17:00 Israel Israeli 08:00 – 17:00 Yuval Ronen
Evening	16:00 – 23:00 Amit Bibi 16:00 – 23:00 Alex Chagan	16:00 – 23:00 Gil Hovav 16:00 – 23:00 Moran Oshia	16:00 – 23:00 Amit Bibi 16:00 – 23:00 Alex Chagan	16:00 – 23:00 Israel Israeli 16:00 – 23:00 Alex Chagan	16:00 – 23:00 Amit Bibi 16:00 – 23:00 Moran Oshia	16:00 – 23:00 Amit Bibi	16:00 – 23:00 Moran Oshia
Night	23:00 - 08:00 Moran Oshia	23:00 - 08:00 Yuval Ronen	23:00 - 08:00 Gil Hovav	23:00 - 08:00 Yuval Ronen	23:00 - 08:00 Israel Israeli	23:00 - 08:00 Tom Cohen	23:00 - 08:00 Amit Bibi

6.2.7 Employee Fill Shift board Suggestion Screen



Hello Tom Cohen

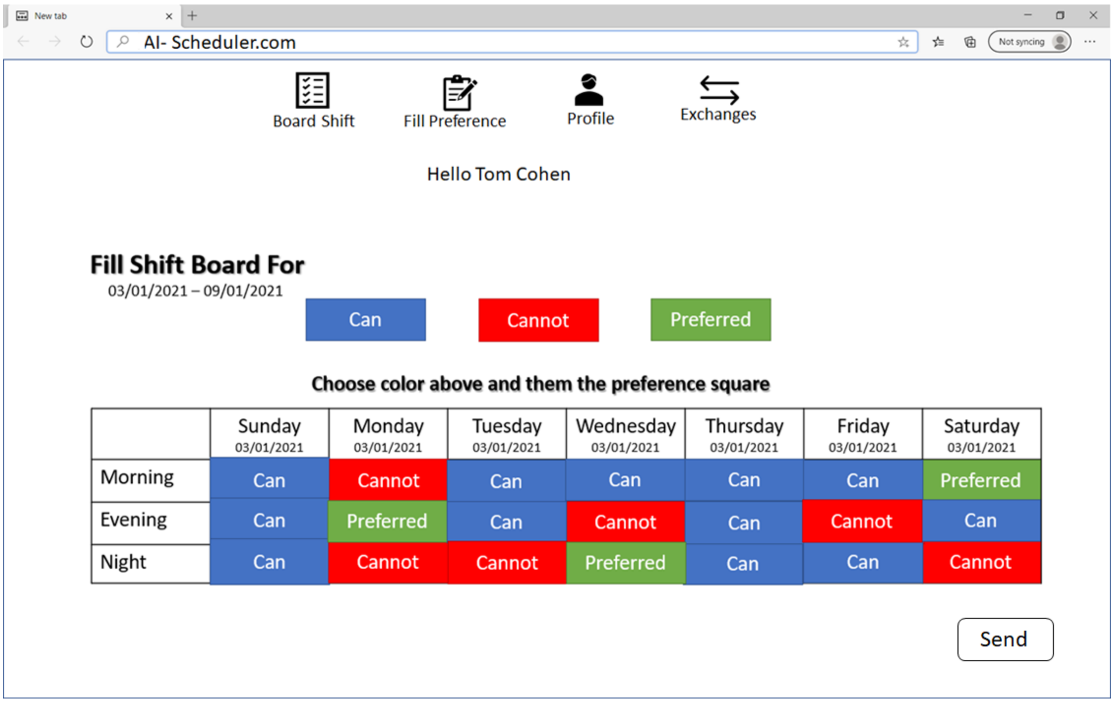
Fill Shift Board For

03/01/2021 – 09/01/2021

Here is a suggestion base on the previous preference

	Sunday 03/01/2021	Monday 03/01/2021	Tuesday 03/01/2021	Wednesday 03/01/2021	Thursday 03/01/2021	Friday 03/01/2021	Saturday 03/01/2021
Morning	Can	Cannot	Can	Can	Can	Can	Preferred
Evening	Can	Preferred	Can	Cannot	Can	Cannot	Can
Night	Can	Cannot	Cannot	Preferred	Can	Can	Cannot

6.2.8 Employee Fill Shift board Screen



Hello Tom Cohen

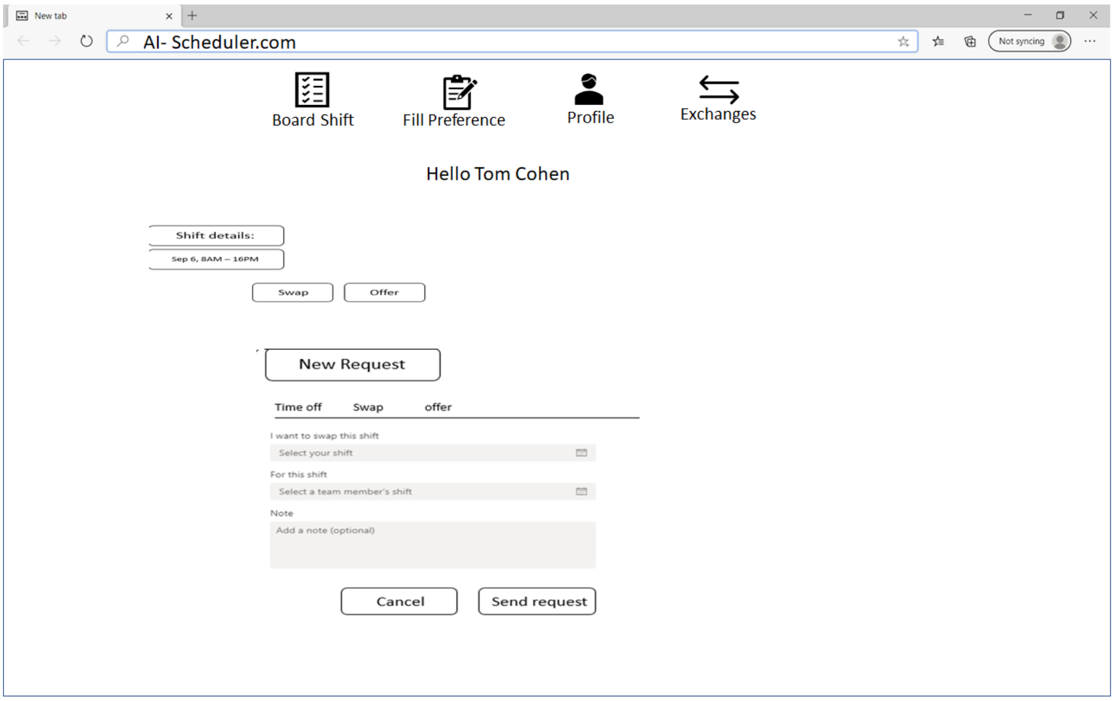
Fill Shift Board For

03/01/2021 – 09/01/2021

Choose color above and them the preference square

	Sunday 03/01/2021	Monday 03/01/2021	Tuesday 03/01/2021	Wednesday 03/01/2021	Thursday 03/01/2021	Friday 03/01/2021	Saturday 03/01/2021
Morning	Can	Cannot	Can	Can	Can	Can	Preferred
Evening	Can	Preferred	Can	Cannot	Can	Cannot	Can
Night	Can	Cannot	Cannot	Preferred	Can	Can	Cannot

6.2.9 Employee Swap/Offer Request Screen



Hello Tom Cohen

Shift details:

Sep 6, 8AM – 16PM

I want to swap this shift

Select your shift

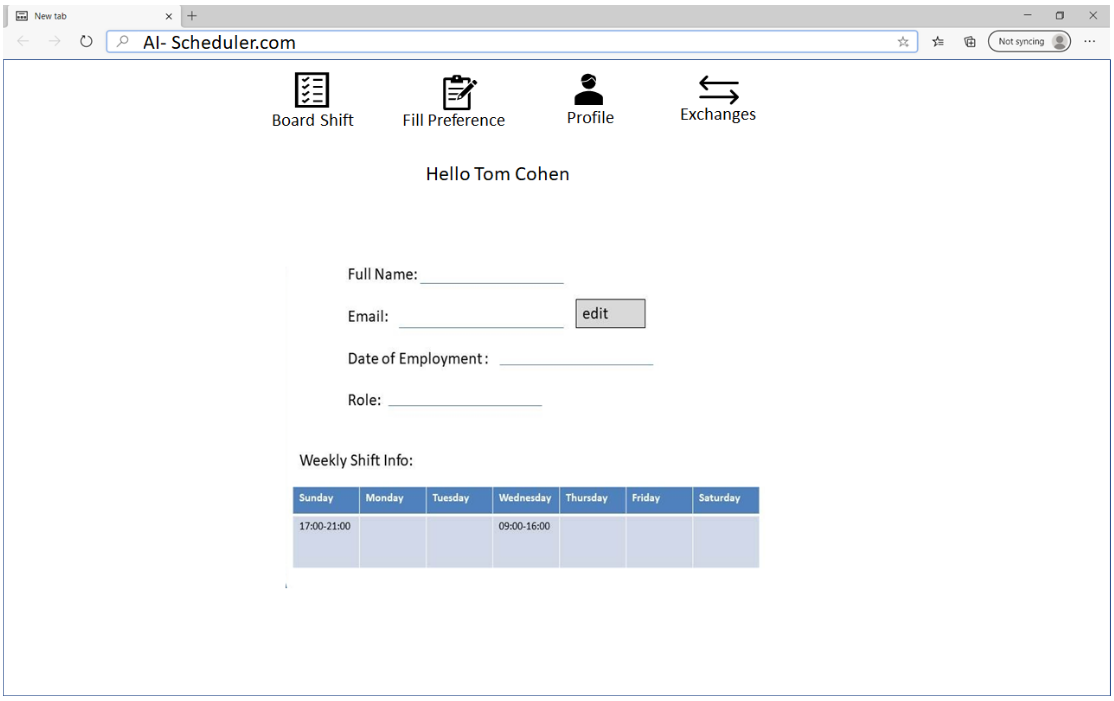
For this shift

Select a team member's shift

Note

Add a note (optional)

6.2.10 Employee Info Screen



Hello Tom Cohen

Full Name: ____________________

Email: ______________________

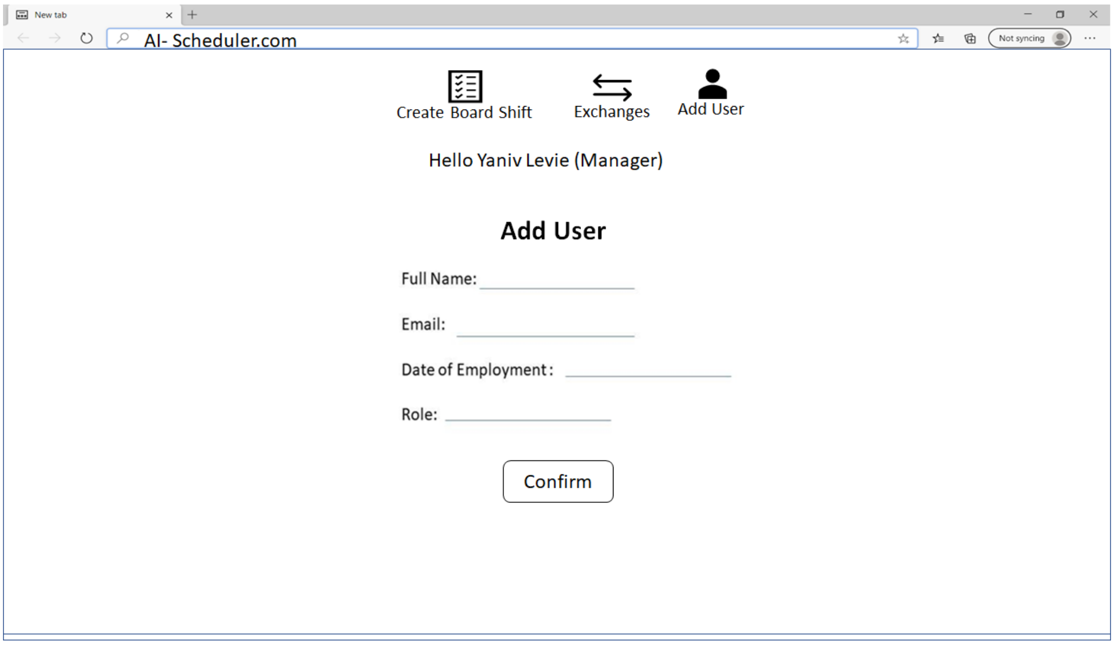
Date of Employment : ____________________

Role: ____________________

Weekly Shift Info:

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
17:00-21:00			09:00-16:00			

6.2.11 Add Employee Screen



Hello Yaniv Levie (Manager)

Add User

Full Name: ____________

Email: ____________

Date of Employment: ____________

Role: ____________

7. Requirements Matrix

No.	Functional Requirements	System components
1	Login: user login page	Database, Web application
2	Insert shifts	Database, Web application
3	Delete shifts	Database, Web application
4	Offer shifts or swap shifts	Database, Web application
5	Manage shifts	Database, Web application
6	Add a new employee	Web application
7	Delete employee	Database, Web application
8	Suggest shift	ML