

ISeeU 2.0 Backend Design - Shared

- Stack
- Design
- Databases
 - Users
 - Patient Info
- API
 - User Login
 - Update/Create Patient
 - Get Patient
 - Get All Patients
 - Delete Patient

Stack

Frontend Stack:

- HTML
- CSS
- Vanilla JS
- jQuery
- Bootstrap
- JWS

Backend Stack:

- AWS EC2 Server written in NodeJS
- Website hosted on AWS S3
- AWS DyanmoDB Databases
- JWT communication encryption.
- [Dynamoose](#) DB management library
- [Plantt](#) Bot API created in YAML files
- [Plantt](#) functions created in Python language

WhatsApp account by [Tyntec](#), SMS account by [019SMS](#)

Design

We had a single user database and a separate patient database for each hospital. Users database can be split into different tables if required.

We created an API for the Website to communicate with the server with the following calls: user login, create/update patient, get patient, get all patients, delete patient. Each call had a different endpoint in our server.

We didn't create a sign up page yet, so users were added manually via the DB UI.

Login design is straightforward, we used JWT for data encryption.

Patients calls (get patient, get all patient, delete, create/update) triggered updates in the DDB.

The medical info was set into a singular JSON object to allow flexibility per hospital.

The call Update/create patient, in addition to updating the DB, triggered a conversation flow created in [Plantt](#), with the patient data as a parameter. This allows sending the data as part of the conversation.

Databases

Users

We used a standard DDB table. One login table for all Hospitals.

Variable Name	Type	Description
Hospital (partition key)	VARCHAR(255)	The hospital this client has access to.
Email (sort key)	VARCHAR(255)	E-mail address for the account. This will also be used as the username for the login of the user account.
PasswordHash	VARCHAR(255)	Password for the account
UserID	INT	Auto-incrementing primary key
IsActive	BOOLEAN	
DateCreated	DATE	
LastLogin	TIME	

Patient Info

A **different table** for each hospital.

Variable Name	Type	Description
Hospital (partition key)	VARCHAR(255)	The hospital this client has access to.
PatientId (sort key)	VARCHAR(255)	Patient Hospital ID numbe
Identifier	INT	Auto-incrementing
Name	VARCHAR(255)	Patient name
PhoneList	ARRAY	Array of strings representing the patient's family phone numbers that will receive the updates
LastUpdate	DATE	Format: DD/MM/YYYY HH:MM
MedicalInfo	JSON	key:value pairs of medical info

medical_info key values:

internal_comments

doctor

nurse

condition

tendency

is_aware

is_on_respirator

is_in_pain

eat_drink

next_step

general

API

These are the calls and endpoints presented to the Frontend team to get and update the users and patients info.

User Login

Called by Login UI to verify user exists.

(POST) https://.../icu/users/user_login

Request Headers:

Variable Name	Value
Content-Type	application/json

Request Payload:

Variable Name	Type	Description
email	VARCHAR(255)	user email
password	VARCHAR(255)	user password
hospital	VARCHAR(255)	Two options: shaare zedek OR hadassah

example:

```
{
  "email": "harel@spetz.io"
  "password": "sisma1234"
  "hospital": "zedek"
}
```

Response Payload:

Variable Name	Type	Description
success	BOOLEAN	TRUE if credentials are ok, FALSE otherwise

example:

```
{
  "success": FALSE
}
```

Update/Create Patient

Triggered by submitting Create Patient or Update Patient forms in the UI.

In the current design, each update/create triggers a conversation with the numbers in the phone_list variable.

(POST) https://.../icu/patients/update_patient

Request Headers:

Variable Name	Value
Content-Type	application/json

Request Payload:

Variable Name	Type	Description
patient_id	VARCHAR(255)	Patient Hospital ID number
name	VARCHAR(255)	Patient name
phone_list	ARRAY	An array of strings representing the patient's family phone numbers that will receive the updates
hospital	VARCHAR(255)	Hospital name
medical_info	JSON	key:value pairs of medical info, this determines the text that appears in the messages.
medical_info.internal_comments	VARCHAR(255)	
medical_info.doctor	VARCHAR(255)	
medical_info.nurse	VARCHAR(255)	
medical_info.condition	VARCHAR(255)	
medical_info.tendancy	VARCHAR(255)	
medical_info.is_awake	VARCHAR(255)	
medical_info.is_on_respirator	VARCHAR(255)	
medical_info.is_in_pain	VARCHAR(255)	
medical_info.eat_drink	VARCHAR(255)	
medical_info.next_step	VARCHAR(255)	
medical_info.general	VARCHAR(255)	

payload example:

```
{
  "patient_id": "302166522",
  "name": " ",
  "phone_list": [ "0506955444", "0525016444" ],
  "hospital": "s_zedek",
  "medical_info": {
    "internal_comments": "",
    "doctor": "\ ",
    "nurse": " ",
    "condition": "",
    ...
  }
}
```

Response:

200

Get Patient

Used to fill the form of an existing patient with existing information.

(POST) https://.../icu/patients/get_patient

Request Headers:

Variable Name	Value
Content-Type	application/json

Request Payload:

Variable Name	Type	Description
patient_id	VARCHAR(255)	Patient Hospital ID number
hospital	VARCHAR(255)	Hospital name

```
{
  "patient_id": "302166522",
  "hospital": "s_zedek"
}
```

Response Payload:

If patient exists - code 200 with the info bellow.

If patient doesn't exist - code 401

Variable Name	Type	Description
patient_id	VARCHAR(255)	Patient Hospital ID number
name	VARCHAR(255)	Patient name
phone_list	ARRAY	An array of strings representing the patient's family phone numbers that will receive the updates
Last_update	DATE	Format: DD/MM/YYYY HH:MM
medical_info	JSON	key:value pairs of medical info
medical_info.internal_comments	VARCHAR(255)	
medical_info.doctor	VARCHAR(255)	
medical_info.nurse	VARCHAR(255)	
medical_info.condition	VARCHAR(255)	
medical_info.tendancy	VARCHAR(255)	
medical_info.tendancy	VARCHAR(255)	
medical_info.is_awake	VARCHAR(255)	
medical_info.is_on_respirator	VARCHAR(255)	
medical_info.is_in_pain	VARCHAR(255)	
medical_info.eat_drink	VARCHAR(255)	
medical_info.next_step	VARCHAR(255)	
medical_info.general	VARCHAR(255)	

Response payload example:

```

{
  "patient_id": "302166444",
  "name": " ",
  "phone_list": [ "0506955444", "0525016444" ],
  "last_update": "14/04/2020 17:23",
  "medical_info": {
    "internal_comments": "",
    "doctor": "\  ",
    "nurse": " ",
    "condition": "",
    ...
  }
}

```

Get All Patients

Used to get the list of patients on the main page, in order to select one to update.

(POST) `https://.../icu/patients/get_all`

Request Headers:

Variable Name	Value
Content-Type	application/json

Request Payload:

Variable Name	Type	Description
hospital	VARCHAR(255)	Hospital name

```

{
  "hospital": "s_zedek"
}

```

Response Payload:

If patient exists - code 200 with the info bellow.

If patient doesn't exist - code 401

Variable Name	Type	Description
patient_list	ARRAY	List of Patient Hospital ID numbers

Response payload example:

```

[ "302166444", "013880111", ... ]

```

Delete Patient

Triggered by the Delete Patient button on the Edit Patient page.

(POST) `https://.../icu/patients/delete_patient`

Request Headers:

Variable Name	Value
Content-Type	application/json

Request Payload:

Variable Name	Type	Description
patient_id	VARCHAR(255)	Patient Hospital ID number
hospital	VARCHAR(255)	Hospital name

payload example:

```
{
  "patient_id": "302166444",
  "hospital": "s_zedek"
}
```

Response:

200 - if patient exists

401 - if patient doesn't exists