User Manual
Clinical Decision Support System (CDSS)

Contents

1 General Description

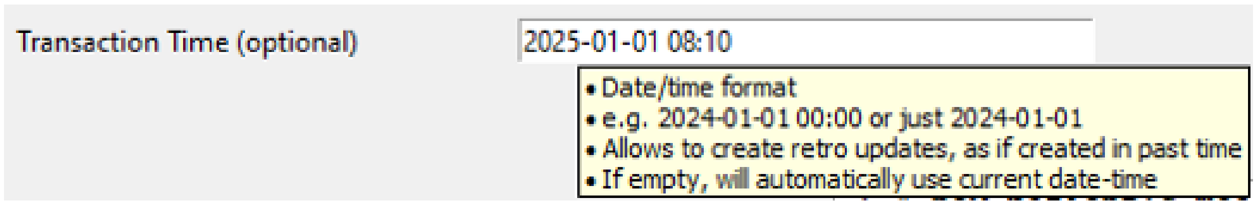
The Clinical Decision Support System (CDSS) provides healthcare professionals with an intuitive interface for managing patient data. The system streamlines clinical workflows through a user-friendly graphical interface that ensures accurate, timely updates and supports informed clinical decision-making.

Through this system, users can:

- **Patient Management:** Register and search for patient records
- **Measurement Tracking:** Record and store standardized medical measurements
- **Historical Data Access:** Search and retrieve past records using precise time filters
- **Record Maintenance:** Update or delete existing measurements as needed
- **LOINC Integration:** Access and track measurements using standardized LOINC codes

2 System Screens Overview

The following sections will guide you through each screen in the CDSS application. As you use the system, you'll notice that every input field provides helpful tooltips when you hover your mouse over them. These tooltips give you clear instructions, format guidelines, and examples offering real-time support exactly when and where you need it.



Transaction Time (optional) 2025-01-01 08:10

- Date/time format
- e.g. 2024-01-01 00:00 or just 2024-01-01
- Allows to create retro updates, as if created in past time
- If empty, will automatically use current date-time

Figure 1: In-context assistance.

2.1 Get Patient by Name

This screen allows you to find a patient's ID by searching with their full name. The purpose of this page is to help you retrieve the intended patient's ID, and to avoid duplication problems in name based search when patients share the same first name + last name.

Required Fields:

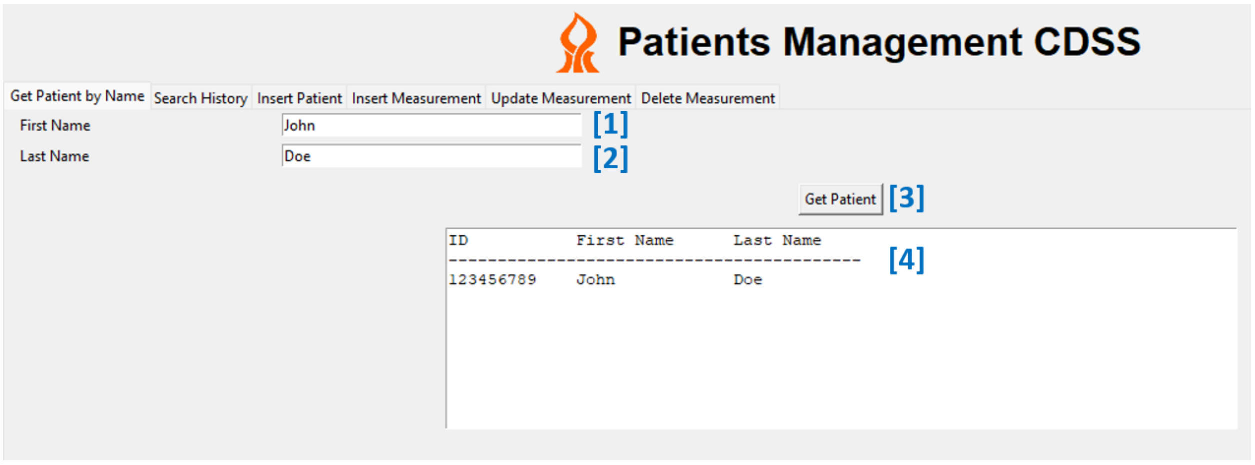
- First Name [1]
- Last Name [2]

Process:

- Enter the patient's first and last name
- Click "Get Patient" [3]
- View results in the display area [4]

Notes:

- All fields are required
- Patient must already be registered in the system
- Search returns all IDs that match the provided name combination



First Name: John
Last Name: Doe

```
ID          First Name      Last Name
------------------------------------------
123456789   John            Doe
```

Figure 2: Finding a patient's ID based on their full name.

2.2 Search History

This screen allows you to retrieve a patient's historical measurement records using various filters.

Required Fields:

- Patient ID [1]

Optional Filters:

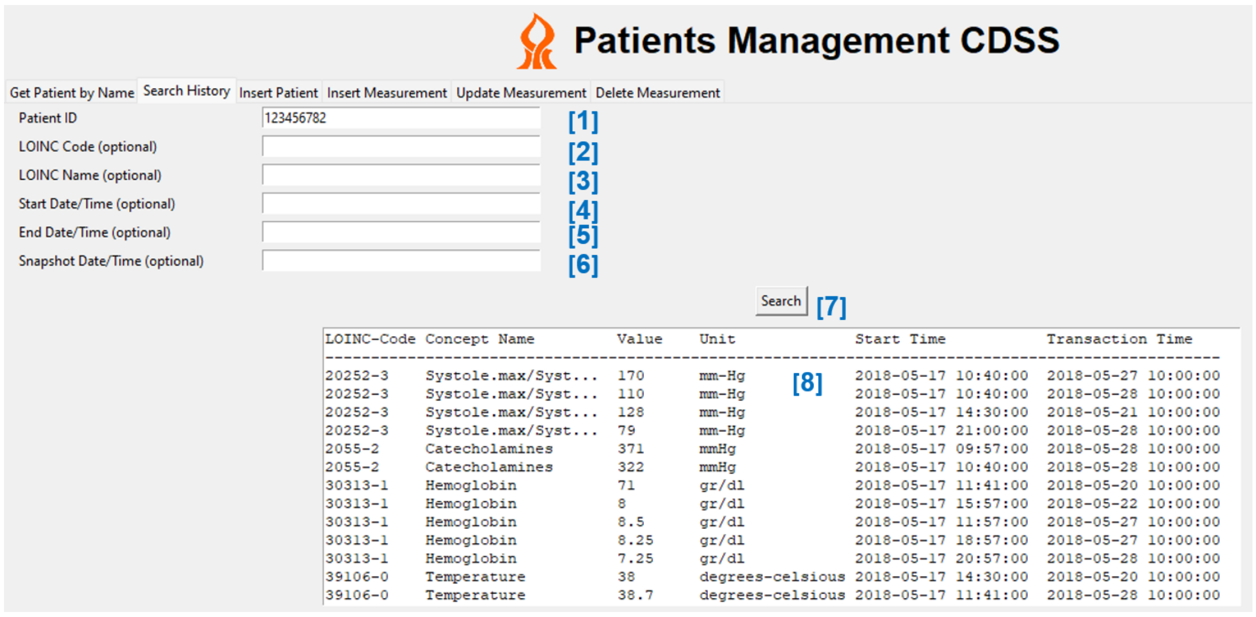
- LOINC Code [2] – Search for a specific standardized code
- LOINC Name [3] – Search for records **containing** this component name (sub strings are acceptable for this screen)
- Start Date/Time [4] – Beginning of search period
- End Date/Time [5] – End of search period
- Snapshot Date/Time [6] – View database as it existed at this point in time

Process:

- Enter the Patient ID (required)
- Apply any desired filters
- Click "Search" [7]
- View matching records in the results area [8]

Notes:

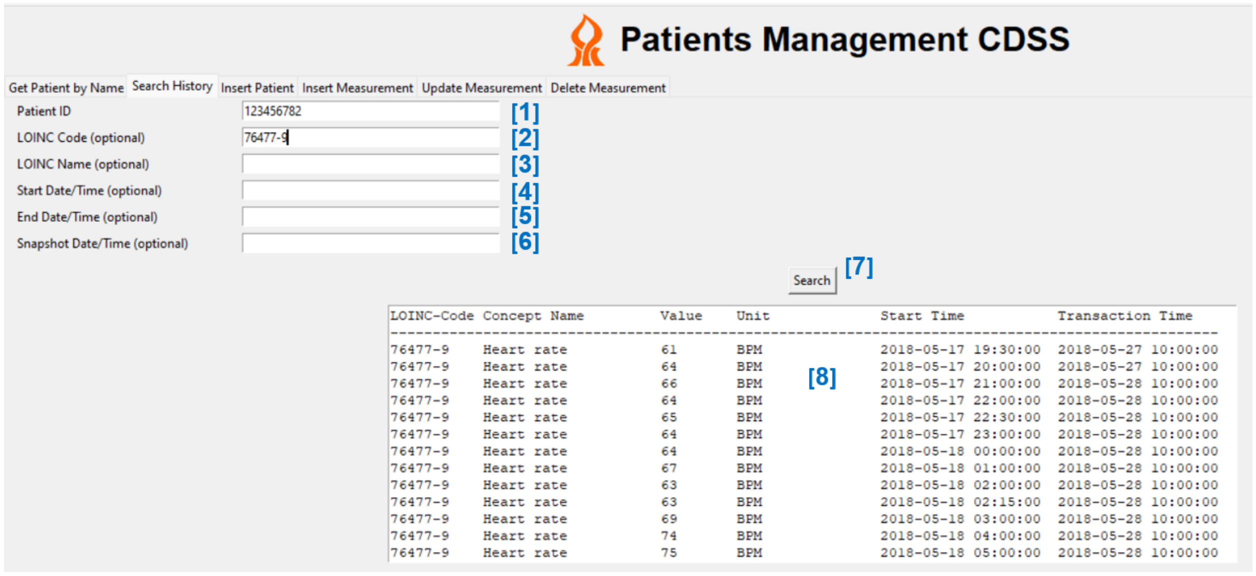
- Date/time format: YYYY-MM-DD HH:MM:SS (24-hour clock)
- For date ranges: if only date is provided, the system will use the full range from the beginning of the first day (00:00:00) to the end of the last day (23:59:59).
- For snapshots: if no time is provided, the system defaults to 00:00:00
- If no records match your criteria, you'll see: "No measurement records found for this patient under these conditions"



Patient ID: 123456782
LOINC Code (optional):
LOINC Name (optional):
Start Date/Time (optional):
End Date/Time (optional):
Snapshot Date/Time (optional):

```
LOINC-Code Concept Name         Value     Unit              Start Time           Transaction Time
----------------------------------------------------------------------------------------------------
20252-3    Systole.max/Syst...  170       mm-Hg             2018-05-17 10:40:00  2018-05-27 10:00:00
20252-3    Systole.max/Syst...  110       mm-Hg             2018-05-17 10:40:00  2018-05-28 10:00:00
20252-3    Systole.max/Syst...  128       mm-Hg             2018-05-17 14:30:00  2018-05-21 10:00:00
20252-3    Systole.max/Syst...  79        mm-Hg             2018-05-17 21:00:00  2018-05-28 10:00:00
2055-2     Catecholamines       371       mmHg              2018-05-17 09:57:00  2018-05-28 10:00:00
2055-2     Catecholamines       322       mmHg              2018-05-17 10:40:00  2018-05-28 10:00:00
30313-1    Hemoglobin           71        gr/dl             2018-05-17 11:41:00  2018-05-20 10:00:00
30313-1    Hemoglobin           8         gr/dl             2018-05-17 15:57:00  2018-05-22 10:00:00
30313-1    Hemoglobin           8.5       gr/dl             2018-05-17 11:57:00  2018-05-27 10:00:00
30313-1    Hemoglobin           8.25      gr/dl             2018-05-17 18:57:00  2018-05-27 10:00:00
30313-1    Hemoglobin           7.25      gr/dl             2018-05-17 20:57:00  2018-05-28 10:00:00
39106-0    Temperature          38        degrees-celsious  2018-05-17 14:30:00  2018-05-20 10:00:00
39106-0    Temperature          38.7      degrees-celsious  2018-05-17 11:41:00  2018-05-28 10:00:00
```

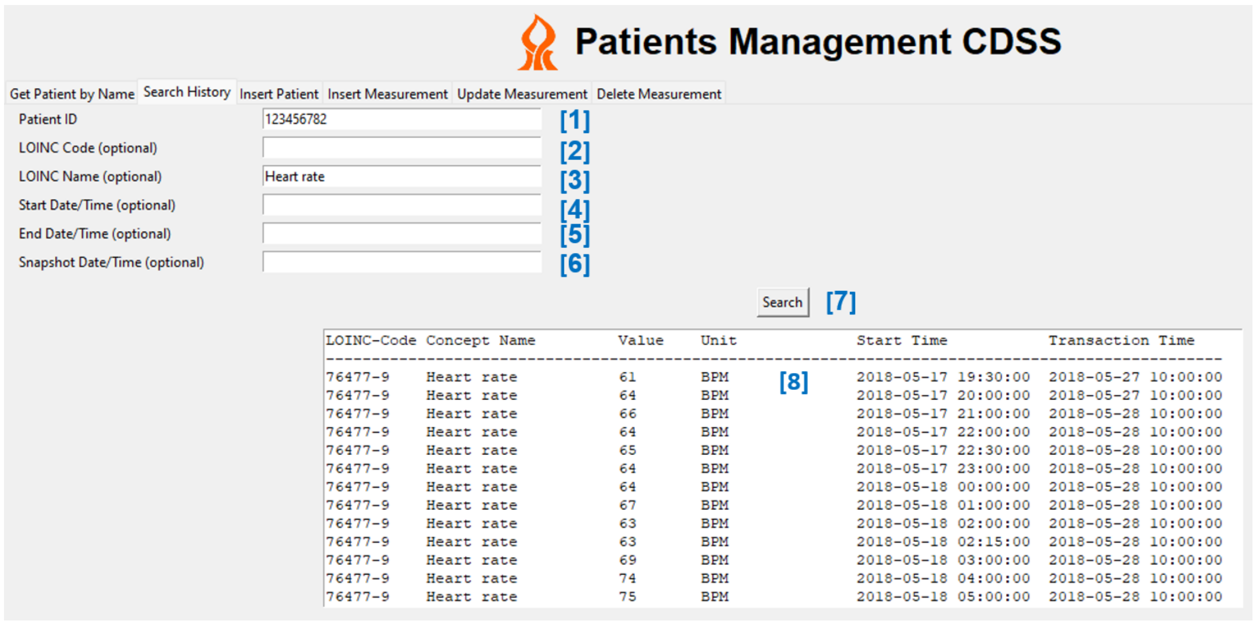
Figure 3: Loading all records matching the patient with the entered ID.



Patient ID: 123456782
LOINC Code (optional): 76477-9
LOINC Name (optional):
Start Date/Time (optional):
End Date/Time (optional):
Snapshot Date/Time (optional):

```
LOINC-Code Concept Name         Value     Unit              Start Time           Transaction Time
----------------------------------------------------------------------------------------------------
76477-9    Heart rate           61        BPM               2018-05-17 19:30:00  2018-05-27 10:00:00
76477-9    Heart rate           64        BPM               2018-05-17 20:00:00  2018-05-27 10:00:00
76477-9    Heart rate           66        BPM               2018-05-17 21:00:00  2018-05-28 10:00:00
76477-9    Heart rate           64        BPM               2018-05-17 22:00:00  2018-05-28 10:00:00
76477-9    Heart rate           65        BPM               2018-05-17 22:30:00  2018-05-28 10:00:00
76477-9    Heart rate           64        BPM               2018-05-17 23:00:00  2018-05-28 10:00:00
76477-9    Heart rate           64        BPM               2018-05-18 00:00:00  2018-05-28 10:00:00
76477-9    Heart rate           67        BPM               2018-05-18 01:00:00  2018-05-28 10:00:00
76477-9    Heart rate           63        BPM               2018-05-18 02:00:00  2018-05-28 10:00:00
76477-9    Heart rate           63        BPM               2018-05-18 02:15:00  2018-05-28 10:00:00
76477-9    Heart rate           69        BPM               2018-05-18 03:00:00  2018-05-28 10:00:00
76477-9    Heart rate           74        BPM               2018-05-18 04:00:00  2018-05-28 10:00:00
76477-9    Heart rate           75        BPM               2018-05-18 05:00:00  2018-05-28 10:00:00
```

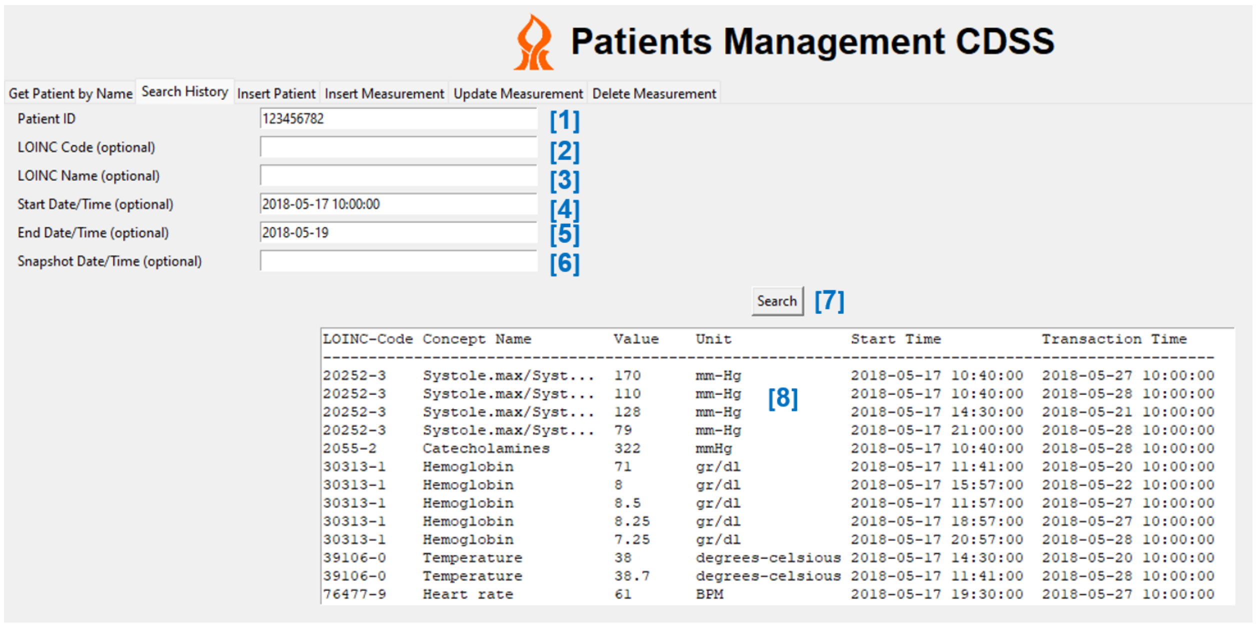
Figure 4: Loading all records for the patient with the entered ID, filtered by the specified LOINC number.



Patient ID	123456782	[1]
LOINC Code (optional)		[2]
LOINC Name (optional)	Heart rate	[3]
Start Date/Time (optional)		[4]
End Date/Time (optional)		[5]
Snapshot Date/Time (optional)		[6]

```
LOINC-Code Concept Name          Value    Unit                Start Time           Transaction Time
---------------------------------------------------------------------------------------------------------
76477-9    Heart rate            61       BPM                 2018-05-17 19:30:00  2018-05-27 10:00:00
76477-9    Heart rate            64       BPM                 2018-05-17 20:00:00  2018-05-27 10:00:00
76477-9    Heart rate            66       BPM                 2018-05-17 21:00:00  2018-05-28 10:00:00
76477-9    Heart rate            64       BPM                 2018-05-17 22:00:00  2018-05-28 10:00:00
76477-9    Heart rate            65       BPM                 2018-05-17 22:30:00  2018-05-28 10:00:00
76477-9    Heart rate            64       BPM                 2018-05-17 23:00:00  2018-05-28 10:00:00
76477-9    Heart rate            64       BPM                 2018-05-18 00:00:00  2018-05-28 10:00:00
76477-9    Heart rate            67       BPM                 2018-05-18 01:00:00  2018-05-28 10:00:00
76477-9    Heart rate            63       BPM                 2018-05-18 02:00:00  2018-05-28 10:00:00
76477-9    Heart rate            63       BPM                 2018-05-18 02:15:00  2018-05-28 10:00:00
76477-9    Heart rate            69       BPM                 2018-05-18 03:00:00  2018-05-28 10:00:00
76477-9    Heart rate            74       BPM                 2018-05-18 04:00:00  2018-05-28 10:00:00
76477-9    Heart rate            75       BPM                 2018-05-18 05:00:00  2018-05-28 10:00:00
```

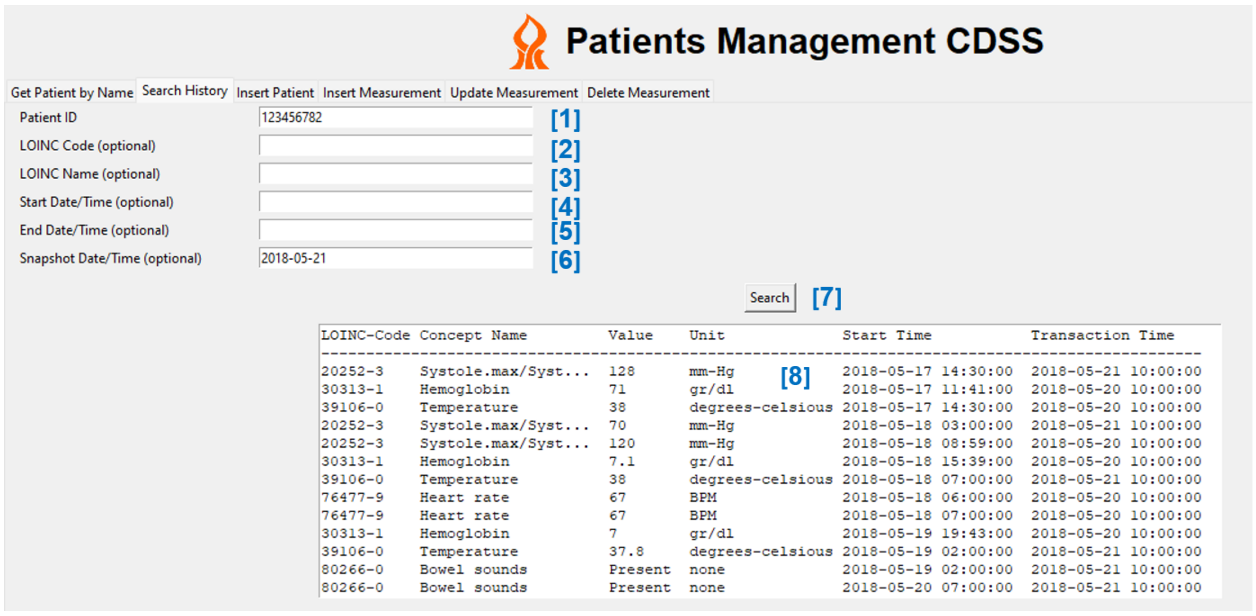
Figure 5: Loading all records for the patient with the entered ID, filtered by the specified LOINC name.



Patient ID	123456782	[1]
LOINC Code (optional)		[2]
LOINC Name (optional)		[3]
Start Date/Time (optional)	2018-05-17 10:00:00	[4]
End Date/Time (optional)	2018-05-19	[5]
Snapshot Date/Time (optional)		[6]

```
LOINC-Code Concept Name          Value    Unit                Start Time           Transaction Time
---------------------------------------------------------------------------------------------------------
20252-3    Systole.max/Syst...   170      mm-Hg               2018-05-17 10:40:00  2018-05-27 10:00:00
20252-3    Systole.max/Syst...   110      mm-Hg               2018-05-17 10:40:00  2018-05-28 10:00:00
20252-3    Systole.max/Syst...   128      mm-Hg               2018-05-17 14:30:00  2018-05-21 10:00:00
20252-3    Systole.max/Syst...   79       mm-Hg               2018-05-17 21:00:00  2018-05-28 10:00:00
2055-2     Catecholamines        322      mmHg                2018-05-17 10:40:00  2018-05-28 10:00:00
30313-1    Hemoglobin            71       gr/dl               2018-05-17 11:41:00  2018-05-20 10:00:00
30313-1    Hemoglobin            8        gr/dl               2018-05-17 15:57:00  2018-05-22 10:00:00
30313-1    Hemoglobin            8.5      gr/dl               2018-05-17 11:57:00  2018-05-27 10:00:00
30313-1    Hemoglobin            8.25     gr/dl               2018-05-17 18:57:00  2018-05-27 10:00:00
30313-1    Hemoglobin            7.25     gr/dl               2018-05-17 20:57:00  2018-05-28 10:00:00
39106-0    Temperature           38       degrees-celsious    2018-05-17 14:30:00  2018-05-20 10:00:00
39106-0    Temperature           38.7     degrees-celsious    2018-05-17 11:41:00  2018-05-28 10:00:00
76477-9    Heart rate            61       BPM                 2018-05-17 19:30:00  2018-05-27 10:00:00
```

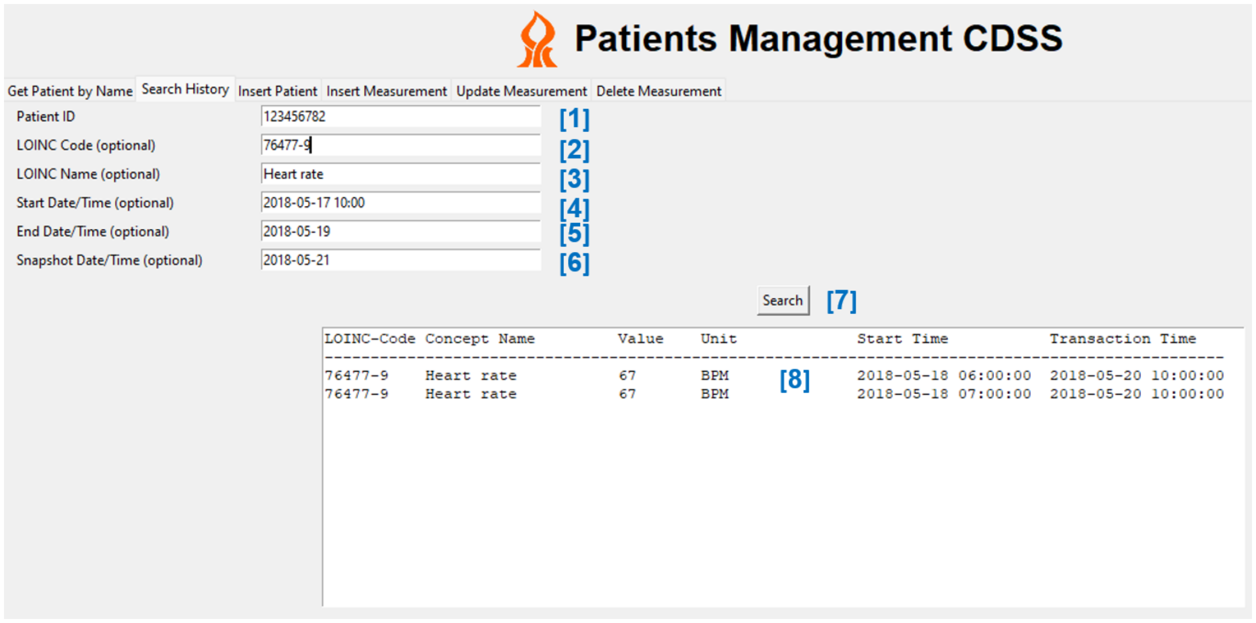
Figure 6: Loading all records for the patient with the entered ID, showing results within the specified date range.



Patient ID	123456782
LOINC Code (optional)	
LOINC Name (optional)	
Start Date/Time (optional)	
End Date/Time (optional)	
Snapshot Date/Time (optional)	2018-05-21

```
LOINC-Code Concept Name        Value    Unit             Start Time           Transaction Time
-----------------------------------------------------------------------------------------------
20252-3    Systole.max/Syst... 128      mm-Hg            2018-05-17 14:30:00  2018-05-21 10:00:00
30313-1    Hemoglobin          71       gr/dl            2018-05-17 11:41:00  2018-05-20 10:00:00
39106-0    Temperature         38       degrees-celsious 2018-05-17 14:30:00  2018-05-20 10:00:00
20252-3    Systole.max/Syst... 70       mm-Hg            2018-05-18 03:00:00  2018-05-21 10:00:00
20252-3    Systole.max/Syst... 120      mm-Hg            2018-05-18 08:59:00  2018-05-20 10:00:00
30313-1    Hemoglobin          7.1      gr/dl            2018-05-18 15:39:00  2018-05-20 10:00:00
39106-0    Temperature         38       degrees-celsious 2018-05-18 07:00:00  2018-05-21 10:00:00
76477-9    Heart rate          67       BPM              2018-05-18 06:00:00  2018-05-20 10:00:00
76477-9    Heart rate          67       BPM              2018-05-18 07:00:00  2018-05-20 10:00:00
30313-1    Hemoglobin          7        gr/dl            2018-05-19 19:43:00  2018-05-20 10:00:00
39106-0    Temperature         37.8     degrees-celsious 2018-05-19 02:00:00  2018-05-21 10:00:00
80266-0    Bowel sounds        Present  none             2018-05-19 02:00:00  2018-05-21 10:00:00
80266-0    Bowel sounds        Present  none             2018-05-20 07:00:00  2018-05-21 10:00:00
```

Figure 7: Loading all records for the patient with the entered ID, showing results according to the specified snapshot.



Patient ID	123456782
LOINC Code (optional)	76477-9
LOINC Name (optional)	Heart rate
Start Date/Time (optional)	2018-05-17 10:00
End Date/Time (optional)	2018-05-19
Snapshot Date/Time (optional)	2018-05-21

```
LOINC-Code Concept Name        Value    Unit             Start Time           Transaction Time
-----------------------------------------------------------------------------------------------
76477-9    Heart rate          67       BPM              2018-05-18 06:00:00  2018-05-20 10:00:00
76477-9    Heart rate          67       BPM              2018-05-18 07:00:00  2018-05-20 10:00:00
```

Figure 8: Loading all records for the patient with the entered ID, with all filters applied.

2.3 Insert Patient

This screen allows you to add a new patient to the system database.

Required Fields:

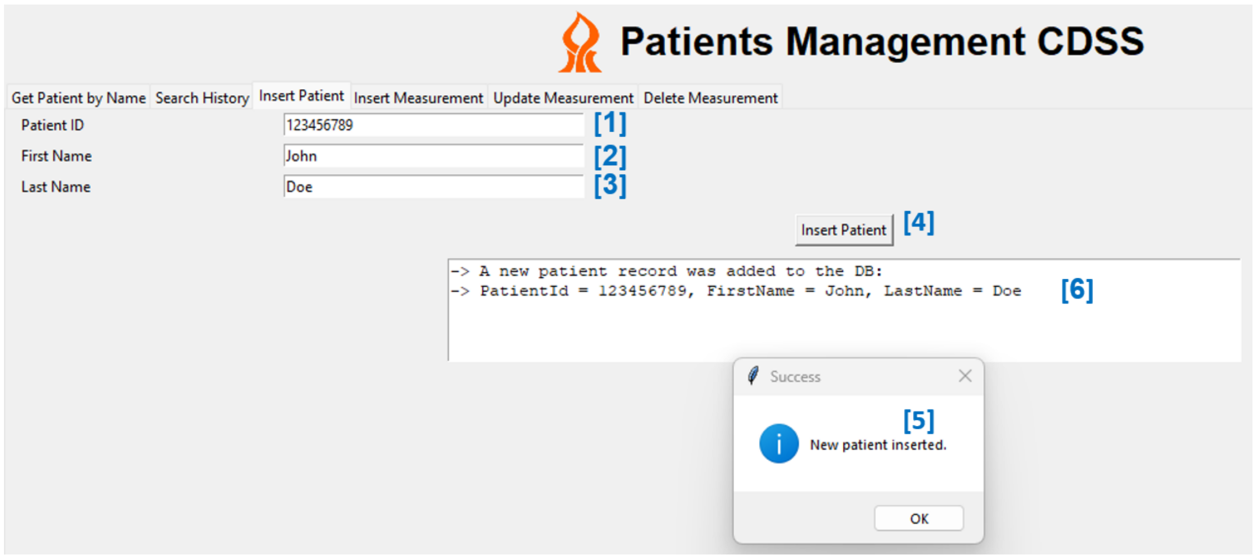
- Patient ID [1]
- First Name [2]
- Last Name [3]

Process:

- Enter all required information
- Click "Insert Patient" [4]
- A confirmation message [5] will be displayed, and a summary of the new patient's registration will be shown on the screen [6]

Notes:

- All fields are mandatory
- The system will prevent duplicate patient IDs with an error message



Patient ID: 123456789
First Name: John
Last Name: Doe

```
-> A new patient record was added to the DB:
-> PatientId = 123456789, FirstName = John, LastName = Doe
```

New patient inserted.

Figure 9: Registering a New Patient.

2.4 Insert Measurement

This screen allows you to record a new measurement for an existing patient.

Required Fields:

- Patient ID [1] – Must match an existing patient
- Valid Start Time [4] – Exact time the measurement was taken
- Value [5] – The measurement result
- Unit [6] – Unit of measurement
- Either LOINC Code [2] OR Component [3] – At least one must be provided. A lookup will be performed in the patient's data to find the relevant LOINC code.

Optional Fields:

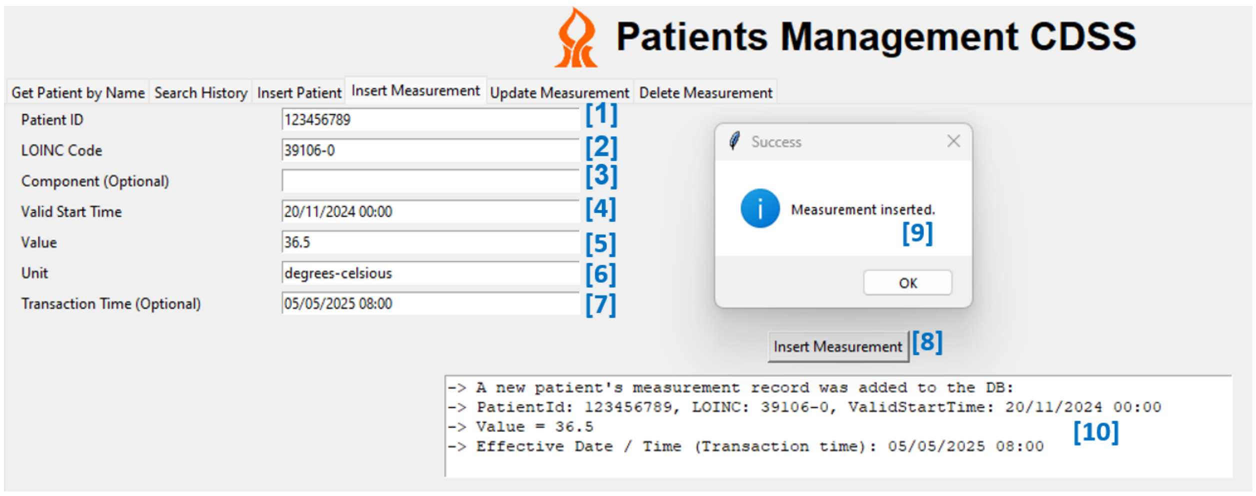
- Transaction Time [7] – When the record was created (defaults to current time)

Process:

- Enter all required information
- Click "Insert Measurement" [8]
- View confirmation message [9] and record summary [10]

Notes:

- Date/time format: YYYY-MM-DD HH:MM:SS (24-hour clock). You are encouraged to use the "Search History" page to locate the exact record you wish to modify.
- If no transaction time is entered, the current time will be used automatically
- When using component name only, the system will attempt to match it to the appropriate LOINC code. If several codes are found to match the input, the user will be notified and the process will be stopped.
- This page will only allow you to modify records that are active. A deleted record cannot be modified further.



Patient ID	123456789	[1]
LOINC Code	39106-0	[2]
Component (Optional)		[3]
Valid Start Time	20/11/2024 00:00	[4]
Value	36.5	[5]
Unit	degrees-celsious	[6]
Transaction Time (Optional)	05/05/2025 08:00	[7]

Measurement inserted. [9]

```
-> A new patient's measurement record was added to the DB:
-> PatientId: 123456789, LOINC: 39106-0, ValidStartTime: 20/11/2024 00:00
-> Value = 36.5
-> Effective Date / Time (Transaction time): 05/05/2025 08:00
```
[10]

Figure 10: Inserting a New Measurement for a Patient.

2.5 Delete Measurement

This screen allows you to remove a measurement from active records.

Required Fields:

- Patient ID [1] – ID of the patient
- Valid Start Time [2] – Timestamp of the measurement to delete
- EITHER LOINC Code [3] OR LOINC Component Name [4] – At least one must be provided. A lookup will be performed in the patient's data to find the relevant LOINC code.

Optional Fields:

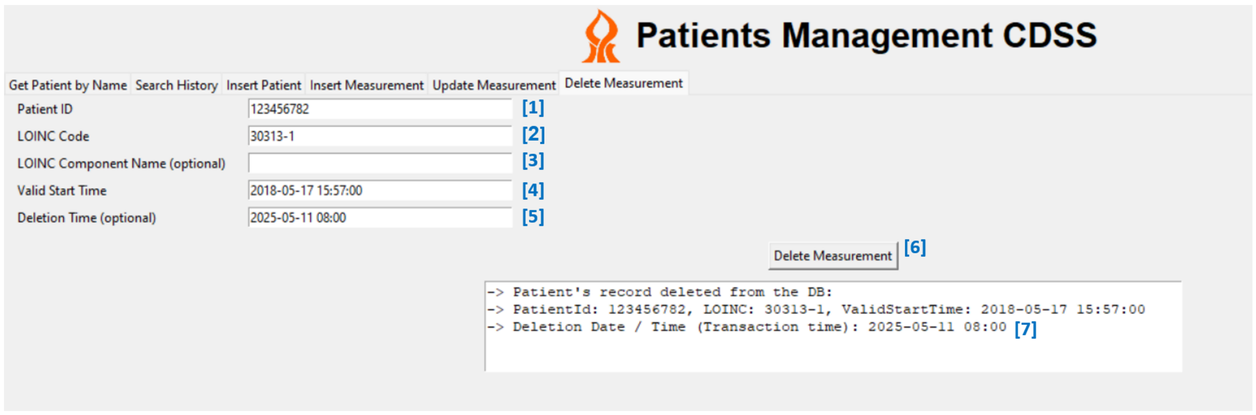
- Deletion Time [5] – When the deletion is recorded (defaults to current time)

Process:

- Enter all required information to identify the record
- Click "Delete Measurement" [6]
- View deleted record details [7]

Notes:

- Date/time format: YYYY-MM-DD HH:MM:SS (24-hour clock) or just the YYYY-MM-DD
- Date/time format: YYYY-MM-DD HH:MM (24-hour clock)
- For Valid Start Time, you can enter just the date to delete the latest record from that day
- Deletions are "logical" rather than physical – the record remains in the database but is marked as deleted
- If no matching record is found, you'll receive an error message



Patient ID	123456782	[1]
LOINC Code	30313-1	[2]
LOINC Component Name (optional)		[3]
Valid Start Time	2018-05-17 15:57:00	[4]
Deletion Time (optional)	2025-05-11 08:00	[5]

```
-> Patient's record deleted from the DB:
-> PatientId: 123456782, LOINC: 30313-1, ValidStartTime: 2018-05-17 15:57:00
-> Deletion Date / Time (Transaction time): 2025-05-11 08:00 [7]
```

Figure 11: Deleting a Measurement for a Patient.

2.6 Patient Snapshot

This screen allows users to view the clinical status of all patients as it was at a specific point in time. The system retrieves the relevant patient records from the database based on the selected snapshot date. The snapshot view includes each patient's **Systemic Toxicity**, **Hematological State**, and **Treatment Recommendations**, and is primarily used for global patient analysis or retrospective clinical review.

Required Field

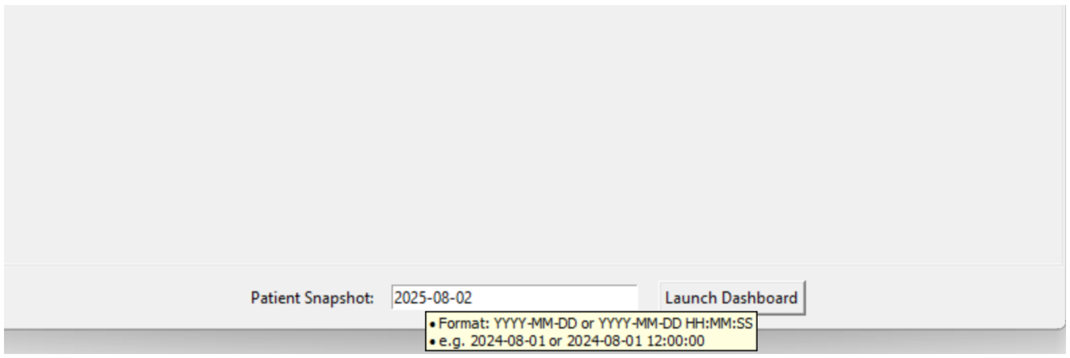
- **Snapshot Date** – A user-specified date and time used to query the database and reconstruct the patients' clinical states as they were at that moment.

Process

- Enter the desired snapshot date and time.
- Click the `Launch Dashboard` button.
- A dashboard will appear, displaying a patient card for each patient who was active at that point in time.

Notes

- If the date is not provided the snapshot date will be the current time and if the format of the Date time is incorrectly formatted - an error message will be shown.
- The snapshot does not modify the data — it simply queries the database in a historical view mode.
- The dashboard includes:
  - Patient filtering (by ID or name)
  - Summary statistics
  - Export options (Excel and JSON)
- The selected snapshot timestamp is displayed at the top of the dashboard screen.



2025-08-02

• Format: YYYY-MM-DD or YYYY-MM-DD HH:MM:SS
• e.g. 2024-08-01 or 2024-08-01 12:00:00

Figure 12: Entering a snapshot date using the format tooltip (e.g., 2025-08-02 or 2024-08-01 12:00:00).

2.7 Snapshot Based Dashboard

After launching a snapshot view, the system displays a visual dashboard summarizing the clinical state of all patients as they were at the specified point in time. This dashboard enables quick triage, review of treatment recommendations, and export of relevant data.

Purpose

The dashboard provides a clear overview of the patient population's clinical status, based on historical data. It is designed to help clinicians and researchers:

- Identify and prioritize critical patients.
- Review systemic toxicity and hematological state.
- Track treatment recommendations and update their status.
- Analyze patient distributions across categories using visual summaries.

Patient Cards Each patient is displayed in an interactive card that contains:

- **Name and ID**
- **Treatment Recommendations** – Presented as checkboxes. Completion status is saved using browser `localStorage`.
- **Systemic Toxicity Indicator** – A colored circle with tooltip:
  - **Red**: GRADE I–IV (severity-based)
  - **Green**: no toxicity detected
- **Hematological Indicator** – A colored circle with tooltip:
  - **Red**: Abnormal (e.g., Anemia, Leukopenia)

  - **Blue**: insufficient data
  - **Green**: Normal

- **Urgency Effect** – GRADE IV cases blink with a red border to indicate critical status.

Patient Filtering A text input field allows filtering of patients by name or ID. Results are automatically sorted by toxicity severity (highest first).

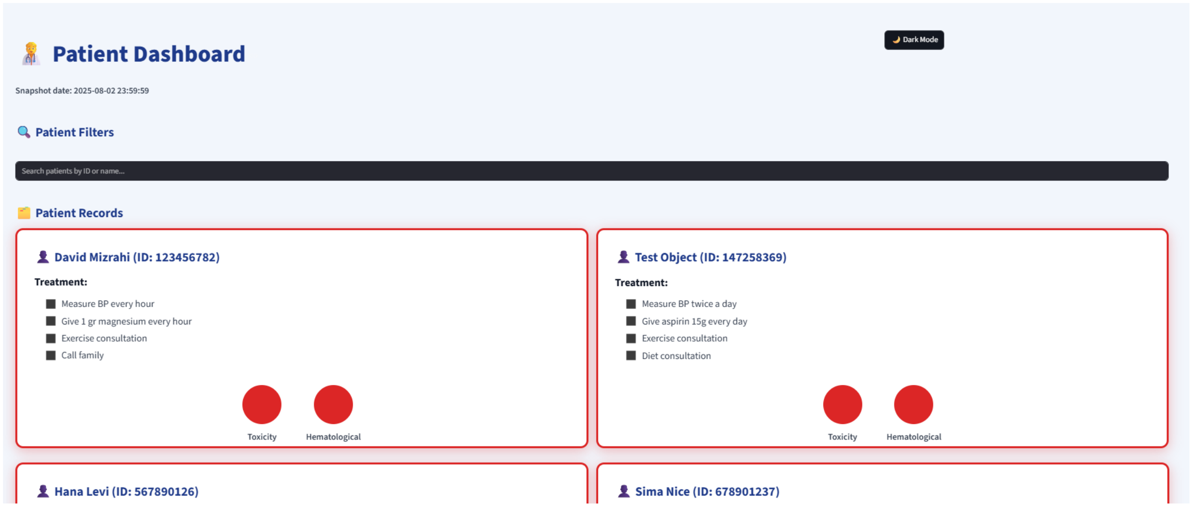
Summary Statistics

- Total number of patients in the snapshot is displayed at the center.
- Two pie charts visualize:
  - Systemic Toxicity distribution
  - Hematological State distribution

5. Export Options

- **Excel Export** – Saves patient data (ID, name, toxicity, hematological state, treatments) in an `.xlsx` file.
- **JSON Export** – Saves the full filtered dataset including snapshot timestamp and patient details.

6. Dark/Light Mode Users can toggle between light and dark themes at runtime.



Patient Dashboard

Snapshot date: 2025-08-02 23:59:59

Patient Filters

Search patients by ID or name...

Patient Records

David Mizrahi (ID: 123456782)

Treatment:

- Measure BP every hour
- Give 1 gr magnesium every hour
- Exercise consultation
- Call family

Toxicity Hematological

Test Object (ID: 147258369)

Treatment:

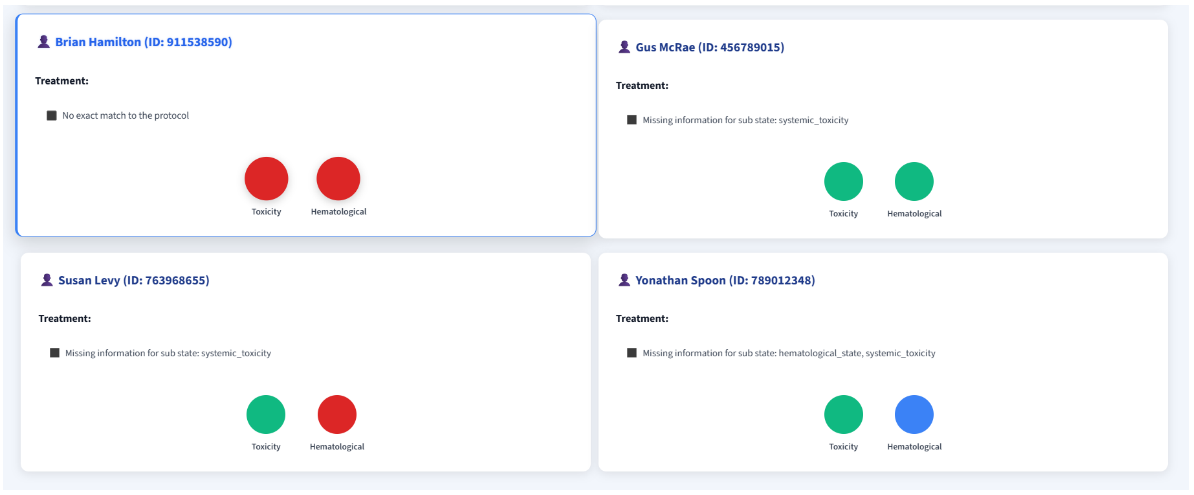
- Measure BP twice a day
- Give aspirin 15g every day
- Exercise consultation
- Diet consultation

Toxicity Hematological

Hana Levi (ID: 567890126)

Sima Nice (ID: 678901237)

Figure 13: Top of the Patient Dashboard: snapshot date, dark mode toggle, search bar, and patient filters


Brian Hamilton (ID: 911538590)

Treatment:

- No exact match to the protocol

Gus McRae (ID: 456789015)

Treatment:

- Missing information for sub state: systemic_toxicity

Susan Levy (ID: 763968655)

Treatment:

- Missing information for sub state: systemic_toxicity

Yonathan Spoon (ID: 789012348)

Treatment:

- Missing information for sub state: hematological_state, systemic_toxicity

Figure 14: Patient Records Section: each card displays ID, name, treatment plan, and two clinical states

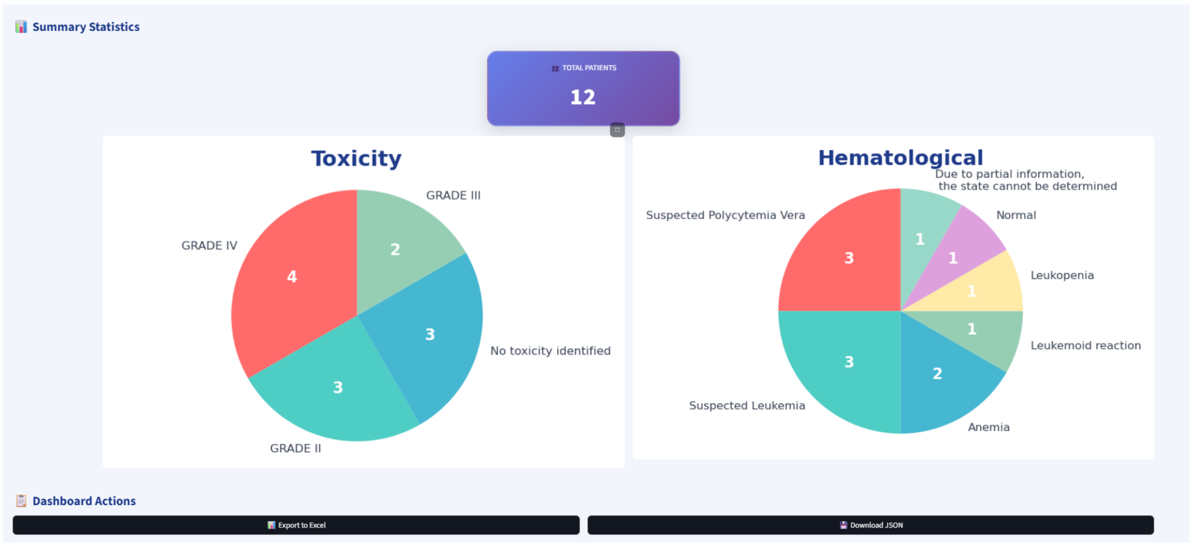


Figure 15: Summary Statistics and Export Options

3 System's Schema

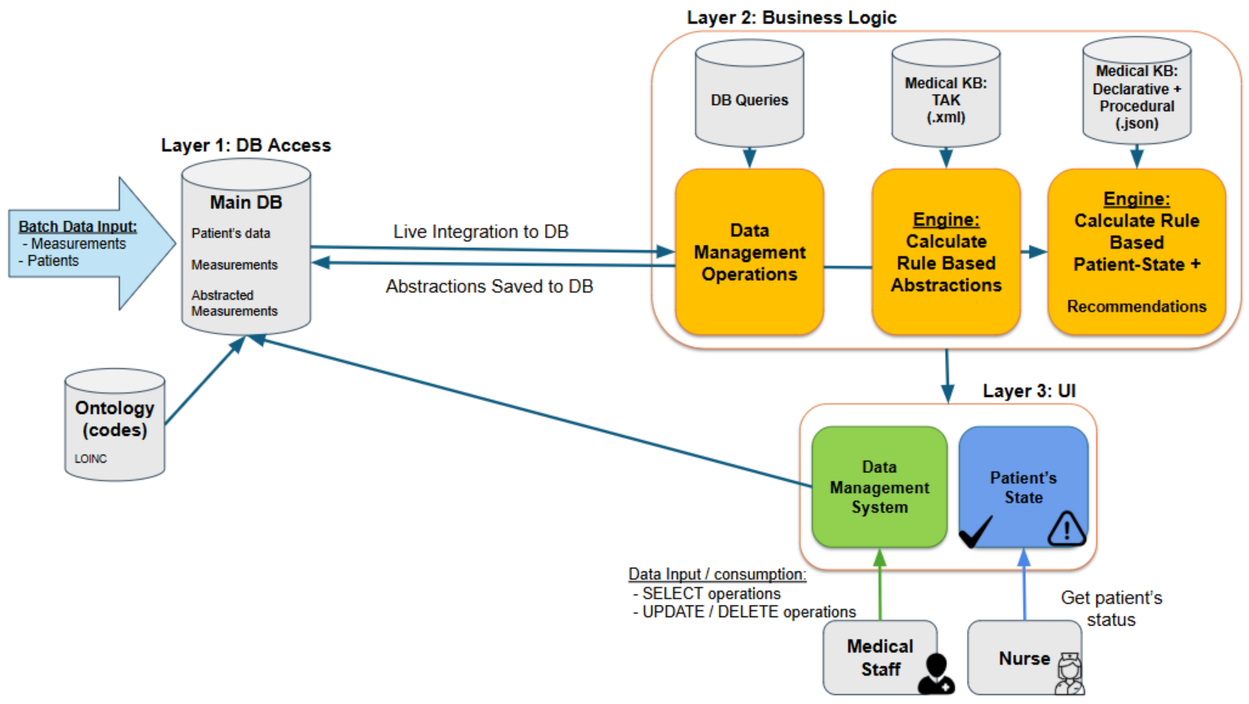
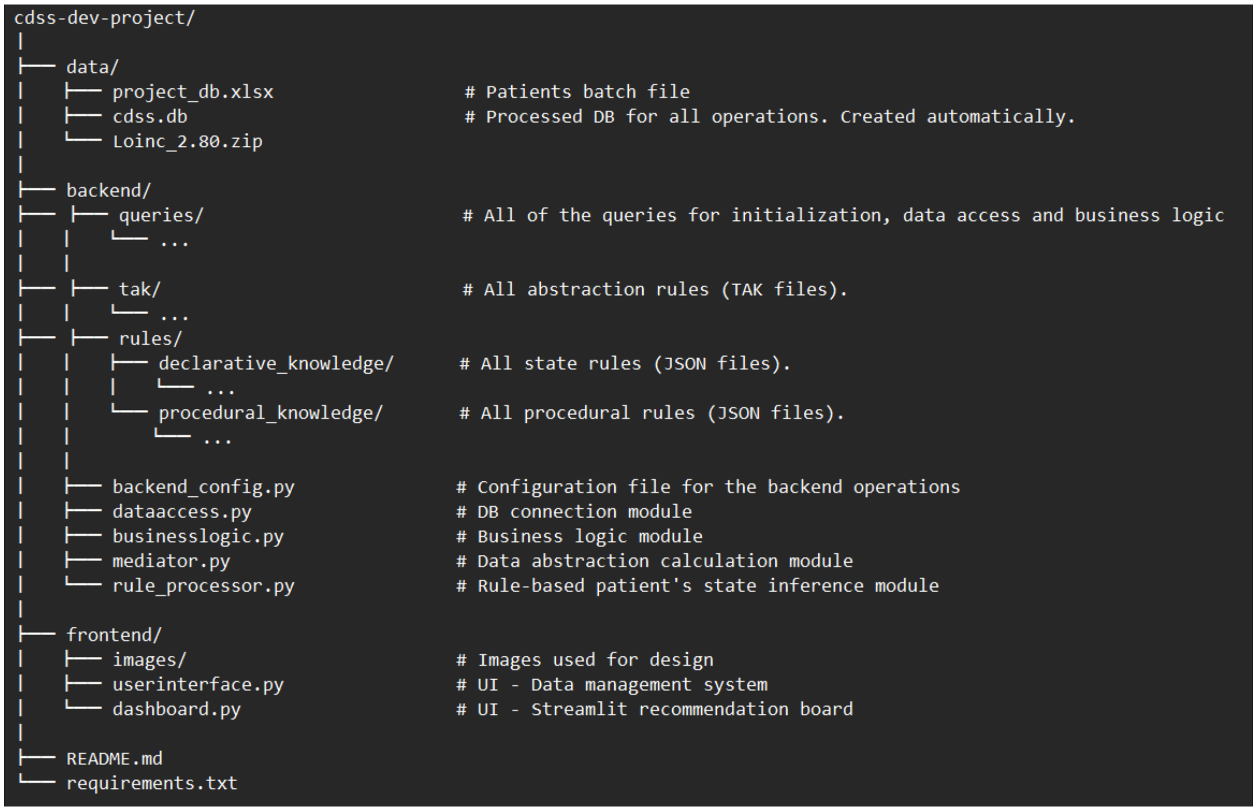


Figure 16: The system's operational schema and different components (high-level).



```
cdss-dev-project/
|
├── data/
|    ├── project_db.xlsx                 # Patients batch file
|    ├── cdss.db                         # Processed DB for all operations. Created automatically.
|    └── Loinc_2.80.zip
|
├── backend/
├── ├── queries/                         # All of the queries for initialization, data access and business logic
|    |    └── ...
|    |
├── ├── tak/                             # All abstraction rules (TAK files).
|    |    └── ...
├── ├── rules/
|    |    ├── declarative_knowledge/     # All state rules (JSON files).
|    |    |    └── ...
|    |    └── procedural_knowledge/      # All procedural rules (JSON files).
|    |         └── ...
|    |
|    ├── backend_config.py               # Configuration file for the backend operations
|    ├── dataaccess.py                   # DB connection module
|    ├── businesslogic.py                # Business logic module
|    ├── mediator.py                     # Data abstraction calculation module
|    └── rule_processor.py               # Rule-based patient's state inference module
|
├── frontend/
|    ├── images/                         # Images used for design
|    ├── userinterface.py                # UI - Data management system
|    └── dashboard.py                    # UI - Streamlit recommendation board
|
├── README.md
└── requirements.txt
```

Figure 17: The system's code structure (low-level).

3.1 Overview

The Clinical Decision Support System (CDSS) is designed around a modular architecture that cleanly separates data handling, business logic, inference, and user interaction. It follows a three-layer design—Data Access, Business Logic, and User Interface—and will be extended with an Abstraction Layer, Inference Engine, and Recommendation Board. This layered structure ensures maintainability, robustness, and future extensibility.

3.2 System Components

1. Data Access Layer (DAL)
This layer is implemented in `backend/dataaccess.py` and is responsible for:

- Connecting to the SQLite database (`cdss.db`).
- Executing raw or file-based SQL queries.
- Loading data from external sources such as:
  - Patient records (Excel file `project_db.xlsx`).
  - LOINC terminology set (from `Loinc_2.80.zip`).
- Initializing and populating tables if they do not exist.

2. Business Logic Layer (BLL)
Defined in `backend/businesslogic.py`, this layer manages domain operations:

- Validates and registers patient information.
- Inserts, updates, deletes, or retrieves measurement records.
- Enforces business rules such as patient identity verification, timestamp integrity, and LOINC code validity.
- Supports temporal data abstraction and enforcement of knowledge-based rules on the data for longitudinal analysis.
- Supports querying over the abstracted temporal snapshots, monitoring a patient's state, and gives treatment recommendation.

3. User Interface (UI)
Built using `Tkinter` in `frontend/userinterface.py` and `Streamlit` in `frontend/dashboard.py`, the UI offers:

- Tab-based interaction for all CRUD operations.
- Data validation and live error handling with user feedback.
- Measurement and patient search based on multiple filters.

- Entry tool-tips and result previews for usability.
- Monitoring and recommendation dashboard is available. The dashboard is launched from within the main UI ("Launch Dashboard" button).

Knowledge Processing Engines:

- **Abstraction Layer (`mediator.py`):** Processes raw measurement data and applies clinical abstractions based on pre-defined TAK files (`.xml`).
- **Inference Engine (`rule processor.py`):** Infers patient health states at specific time points from abstracted data using rule-based logic.This engine also processes the procedural knowledge and proposes clinical actions based on inferred states and guidelines.

3.3 Operational Flow

The system flow follows this logic:

1. On launch, the UI is initialized and connected to the `PatientRecord` handler.
2. All data manipulations are routed through the `businesslogic` module, which validates, transforms, and passes operations to the `dataaccess` module.
3. Query logic is stored under `backend/queries/` as standalone `.sql` files, allowing flexible editing and reuse.
4. Database is built and loaded automatically on first run if `cdss.db` is not found.

3.4 Dependencies and Requirements

Minimum Requirements:

- Python **3.7+**
- SQLite (auto-handled via `sqlite3`)
- Required Python packages: See `requirements.txt`

Key Packages:

- `pandas` — For data manipulation and import.
- `tkinter` — For GUI design - main.
- `streamlit` — For GUI design - state dashboard.
- `openpyxl` — For loading Excel-based patient data.

3.5 File System Summary

- `data/` — Contains the database file, patient data, and LOINC zip.
- `backend/` — Houses business logic, database handler, config, queries, and knowledge files.
- `frontend/` — UI code and image assets.
- `requirements.txt` — Python dependency list.
- `README.md` — Setup and usage instructions.

3.6 Execution Instructions

Initialize System:

```
python -m backend.dataaccess
```

This assumes you already placed your codes directory, your initial dataset as Excel (if exists), edited the data-access module to read them and edited all related paths in the backend configuration file.

Run the Application:

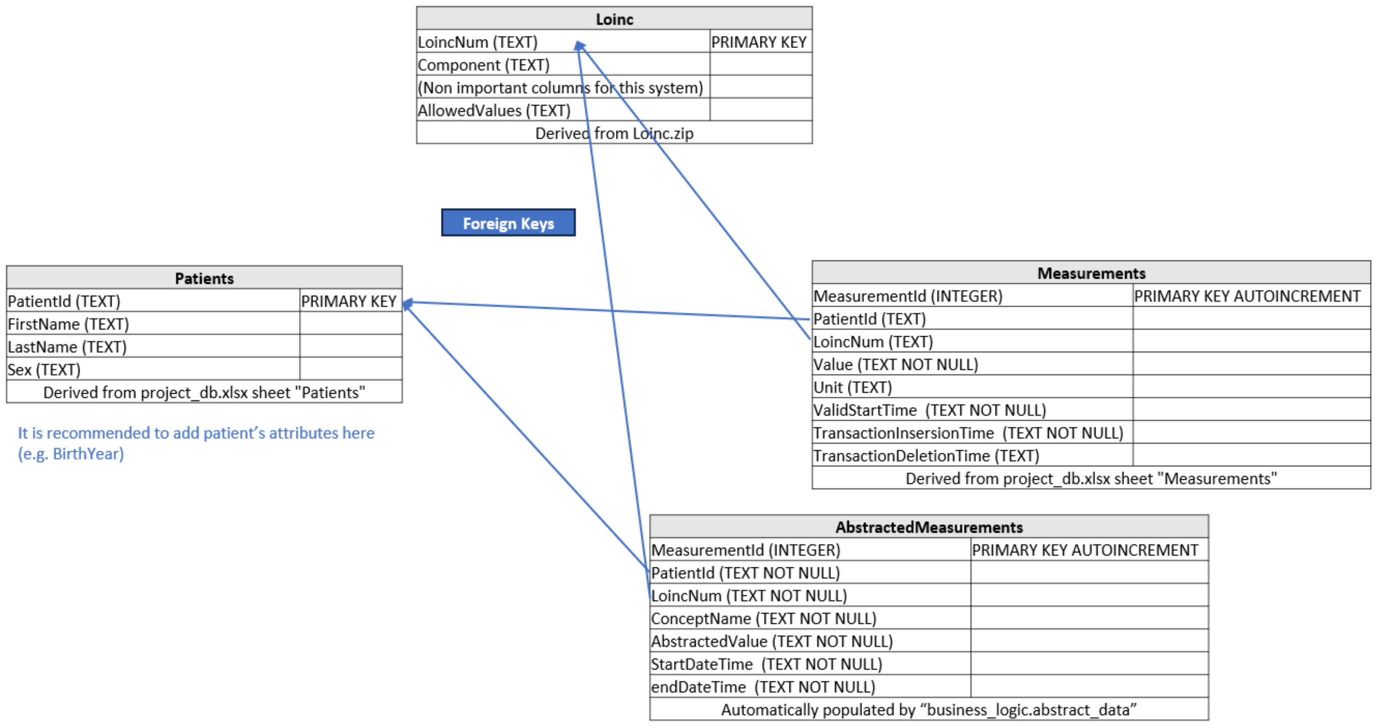
```
python -m frontend.userinterface.py
```

Reset the System (delete DB):

```
rm cdss.db
```

3.7 Database Schema & Initial Data Requirements

The CDSS persists data in a single SQLite file (`cdss.db`) that is automatically created on first run. Figure **??** shows the four core tables and their relationships.


Foreign Keys
It is recommended to add patient's attributes here (e.g. BirthYear)

Loinc	
LoincNum (TEXT)	PRIMARY KEY
Component (TEXT)	
(Non important columns for this system)	
AllowedValues (TEXT)	
Derived from Loinc.zip	

Patients	
PatientId (TEXT)	PRIMARY KEY
FirstName (TEXT)	
LastName (TEXT)	
Sex (TEXT)	
Derived from project_db.xlsx sheet "Patients"	

Measurements	
MeasurementId (INTEGER)	PRIMARY KEY AUTOINCREMENT
PatientId (TEXT)	
LoincNum (TEXT)	
Value (TEXT NOT NULL)	
Unit (TEXT)	
ValidStartTime (TEXT NOT NULL)	
TransactionInsersionTime (TEXT NOT NULL)	
TransactionDeletionTime (TEXT)	
Derived from project_db.xlsx sheet "Measurements"	

AbstractedMeasurements	
MeasurementId (INTEGER)	PRIMARY KEY AUTOINCREMENT
PatientId (TEXT NOT NULL)	
LoincNum (TEXT NOT NULL)	
ConceptName (TEXT NOT NULL)	
AbstractedValue (TEXT NOT NULL)	
StartDateTime (TEXT NOT NULL)	
endDateTime (TEXT NOT NULL)	
Automatically populated by "business_logic.abstract_data"	

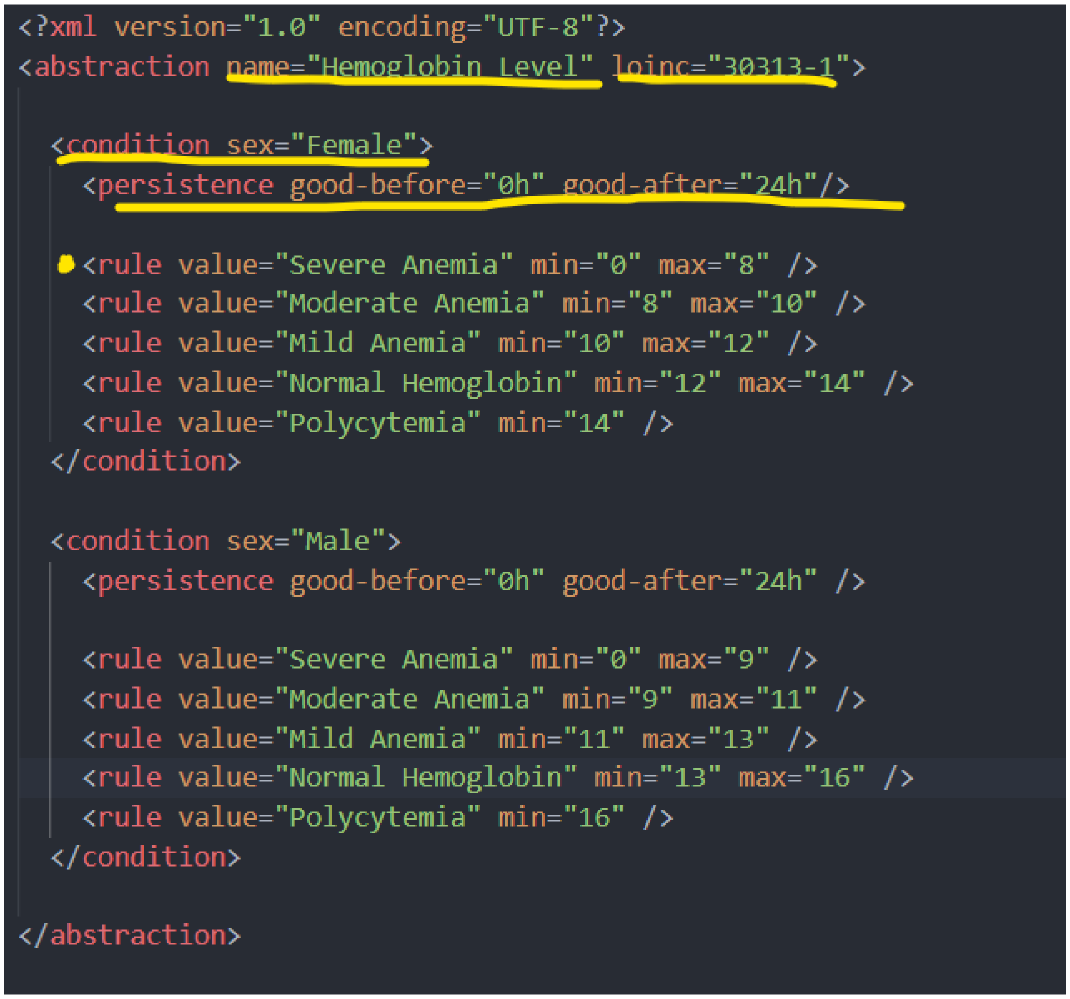
Figure 18: The DB schema. All columns must be available in the initialization `.xlsx` file

3.8 Update Knowledge Base

The Clinical Decision Support System (CDSS) supports local customization through a modular knowledge base that separates concerns into three layers: abstraction, state inference, and procedural recommendation. These layers allow knowledge engineers to introduce clinical logic without modifying the core engine.

Updates are made by editing or adding structured files in three key repositories:

1. **Abstraction Layer** (`backend/taks/`)



```xml
<?xml version="1.0" encoding="UTF-8"?>
<abstraction name="Hemoglobin Level" loinc="30313-1">

  <condition sex="Female">
    <persistence good-before="0h" good-after="24h"/>

    <rule value="Severe Anemia" min="0" max="8" />
    <rule value="Moderate Anemia" min="8" max="10" />
    <rule value="Mild Anemia" min="10" max="12" />
    <rule value="Normal Hemoglobin" min="12" max="14" />
    <rule value="Polycytemia" min="14" />
  </condition>

  <condition sex="Male">
    <persistence good-before="0h" good-after="24h" />

    <rule value="Severe Anemia" min="0" max="9" />
    <rule value="Moderate Anemia" min="9" max="11" />
    <rule value="Mild Anemia" min="11" max="13" />
    <rule value="Normal Hemoglobin" min="13" max="16" />
    <rule value="Polycytemia" min="16" />
  </condition>

</abstraction>
```

Figure 19: Sample abstraction rule file for Hemoglobin Level (TAK layer)

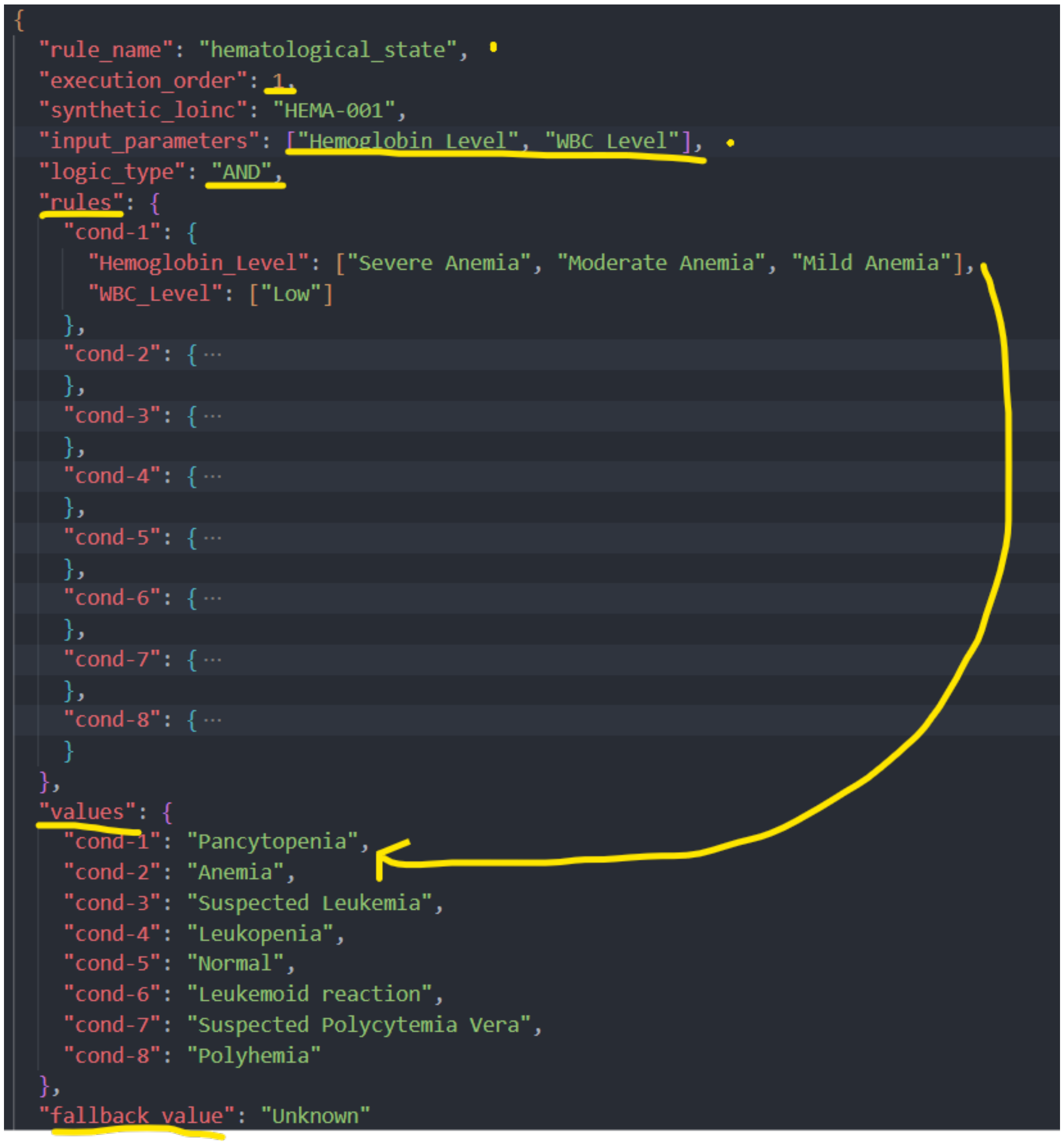
This layer transforms raw numeric or categorical clinical measurements into meaningful temporal intervals. Each abstraction file is an XML document that defines how input values are mapped into discrete labeled states such as *"Mild Anemia"* or *"Polycythemia"*.

- `name` and `loinc` attributes define the concept and LOINC code being abstracted.

- Each `condition` block can be filtered by patient attributes (e.g., sex). You can also leave `<condition>` if you don't need to add condition the abstraction by a parameter. These attributes must be found in your DB `Patients` table, with the same name (there is a lower() forgiveness) and available for extraction by the query `GET_PATIENT_PARAMS_QUERY`.
- `persistence` defines temporal smoothing logic, specifying how long a value remains valid before or after its timestamp.
- Each `rule` maps a numeric range (via `min/max`) to a semantic label like `\Moderate Anemia"`.
- `persistence` block is responsible of the relevance window of an interval (max distance to connect intervals). In this example, the interval is valid for 0 hours before raw record occurred, and 24 hours after. If 2 `\Moderate Anemia"` records occurred with a 6 hours difference, the abstracted new record will have a duration of 30 hours, from the beginning of the first one till the end of the second. There is no limitation on how many records can be concatenated.

Example: Hemoglobin levels are abstracted into sex-specific categories, each with their own numeric ranges and persistence window (see Figure 19).

2. **State Layer** (`backend/rules/declarative_knowledge/`)



```
{
  "rule_name": "hematological_state",
  "execution_order": 1,
  "synthetic_loinc": "HEMA-001",
  "input_parameters": ["Hemoglobin_Level", "WBC_Level"],
  "logic_type": "AND",
  "rules": {
    "cond-1": {
      "Hemoglobin_Level": ["Severe Anemia", "Moderate Anemia", "Mild Anemia"],
      "WBC_Level": ["Low"]
    },
    "cond-2": {···
    },
    "cond-3": {···
    },
    "cond-4": {···
    },
    "cond-5": {···
    },
    "cond-6": {···
    },
    "cond-7": {···
    },
    "cond-8": {···
    }
  },
  "values": {
    "cond-1": "Pancytopenia",
    "cond-2": "Anemia",
    "cond-3": "Suspected Leukemia",
    "cond-4": "Leukopenia",
    "cond-5": "Normal",
    "cond-6": "Leukemoid reaction",
    "cond-7": "Suspected Polycytemia Vera",
    "cond-8": "Polyhemia"
  },
  "fallback_value": "Unknown"
```

Figure 20: Declarative rule file mapping abstracted features to clinical states

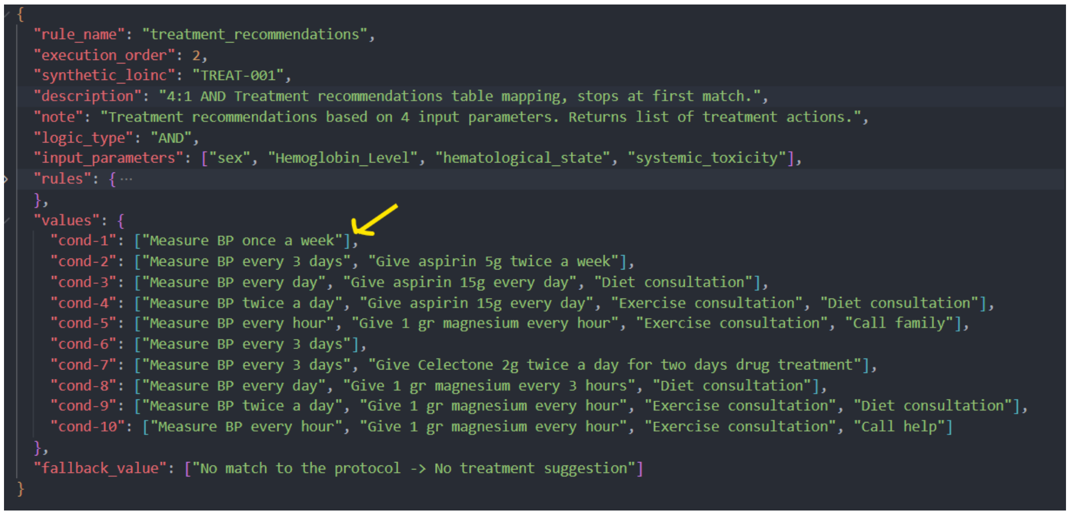
This layer evaluates abstracted intervals and infers higher-level clinical states like *"Toxic Exposure"* or *"Pancytopenia"*. These files are authored in JSON format.

- `rule_name` defines the concept being computed.
- `execution_order` defines the hierarchal order of execution. If a state's condition is dependent on a previous states result, execution order allows you to sort their executions. Assign smaller numbers to states you want executed sooner.
- `synthetic_loinc` decide on a fake ID for the new record so that these abstractions will be possible to log in a DB.

- `input_parameters` are the names of abstracted intervals and context parameters that are required as inputs. These parameters will be sought after by name in the `AbstractedMeasurement.ConceptName` column and in the `Patients` table columns.
- `rules` define logical matching conditions (e.g., a value match for multiple parameters). To match a parameter, the record must match at least 1 of the values associated with this parameter (e.g. either Severe, Moderate or Mild Anemia under `Hemoglobin_Level`.
- `logic_type` can be `AND` or `OR`, determining rule evaluation logic. An `AND` logic means a patient must match all the parameters under a specific condition. First matched condition is assigned. An `OR` condition means a patient must match at least one parameter to be assigned a condition match, and the maximal matched condition is assigned to the patient.
- `values` assigns a label (state) to a patient based on their assigned / matched condition (e.g., "Anemia", "Suspected Leukemia").
- `fallback_value` assigns a label (state) to a patient under this state in case they won't match any of the pre-defined conditions (no nulls system).

Example: A rule may define "Pancytopenia" when both WBC and Hemoglobin are below defined thresholds (see Figure 20).

3. **Recommendation Layer** (`backend/rules/procedural_knowledge/`)



```
{
  "rule_name": "treatment_recommendations",
  "execution_order": 2,
  "synthetic_loinc": "TREAT-001",
  "description": "4:1 AND Treatment recommendations table mapping, stops at first match.",
  "note": "Treatment recommendations based on 4 input parameters. Returns list of treatment actions.",
  "logic_type": "AND",
  "input_parameters": ["sex", "Hemoglobin_Level", "hematological_state", "systemic_toxicity"],
  "rules": {…
  },
  "values": {
    "cond-1": ["Measure BP once a week"],
    "cond-2": ["Measure BP every 3 days", "Give aspirin 5g twice a week"],
    "cond-3": ["Measure BP every day", "Give aspirin 15g every day", "Diet consultation"],
    "cond-4": ["Measure BP twice a day", "Give aspirin 15g every day", "Exercise consultation", "Diet consultation"],
    "cond-5": ["Measure BP every hour", "Give 1 gr magnesium every hour", "Exercise consultation", "Call family"],
    "cond-6": ["Measure BP every 3 days"],
    "cond-7": ["Measure BP every 3 days", "Give Celectone 2g twice a day for two days drug treatment"],
    "cond-8": ["Measure BP every day", "Give 1 gr magnesium every 3 hours", "Diet consultation"],
    "cond-9": ["Measure BP twice a day", "Give 1 gr magnesium every hour", "Exercise consultation", "Diet consultation"],
    "cond-10": ["Measure BP every hour", "Give 1 gr magnesium every hour", "Exercise consultation", "Call help"]
  },
  "fallback_value": ["No match to the protocol -> No treatment suggestion"]
}
```

Figure 21: Procedural rule file defining treatment actions based on inferred states

This layer encodes procedural knowledge by mapping patient states to suggested actions or interventions. Each file is written in JSON format and follows an almost identical structure to the state layer but outputs a list of actions instead of a single state.

This difference in output is critical and affects the ability of the dashboard to present results.

Example: Based on the combination of toxicological and hematological states, a patient may be advised to take medication, measure vitals at specific intervals, or seek specialist consultation (see Figure 21).

To update the knowledge base, domain experts should:

- Add new XML files under `backend/taks/` to define new abstractions.
- Author new JSON rule files under `declarative_knowledge/` to define additional patient states.
- Extend or modify JSON files under `procedural_knowledge/` to suggest new treatment recommendations.

Each rule file is loaded at runtime, so changes will take effect automatically upon application restart.

The KB folders are automatically monitored on execution and will prompt you on structural breaches.

4 DSS dimensions

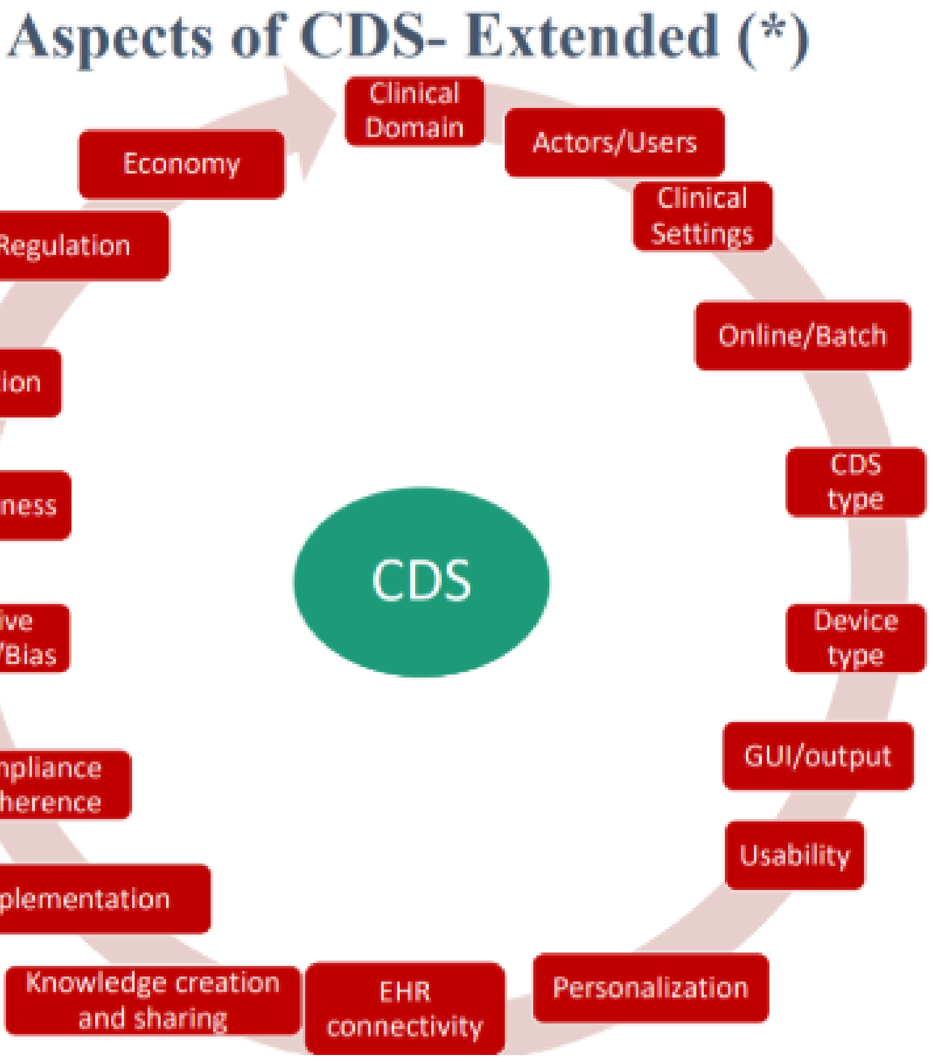


Figure 22: The DSS dimensions.

– **Clinical Domain:** While the system is designed to be LOINC-compatible and theoretically supports any clinical measurement, its current implementation and rule structure are explicitly tailored for hematological diagnostics. With that being said, it can be easily expanded to other domains as well.

– **Actors / Users:** The system is intended for use by clinical staff, including physicians and nurses.

– **Clinical Settings:** Given the system's current focus on hematological diagnostics and its standalone architecture, the hospital inpatient ward emerges as the most appropriate clinical setting for deployment.

– **Online / Batch:** The system operates in an online mode, performing on-the-fly data validation and retrieval with high connectivity to the database, allowing users to interact with patient records in real time through a responsive form-based interface.

– **CDS Type:** The system is a knowledge-driven CDSS that combines diagnostic support with elements of treatment guidance based on structured rule files. It uses predefined clinical logic to analyze hematological lab results and identify conditions such as anemia or leucopenia, while also referencing treatment-related rules to flag when certain interventions may be clinically indicated.

– **Device Type:** The system is designed to be operated on a desktop computer, though certain aspects of it (Insert Measurement or treatment recommendations) can benefit from deployment to the hospital staff's tablets.

– **GUI / Output:** The system presents clinical data through a structured, tab-based graphical interface. Standard operations such as patient search, measurement insertion, and data retrieval are displayed using clear text-box forms with real-time validation and feedback. In addition, the system includes a snapshot-based visual dashboard that displays the patient's clinical state—at a selected point in time—through interactive patient cards. Each card summarizes the patient's name, ID, hematological condition, systemic toxicity level, and treatment recommendations. Color-coded indicators (e.g., red for abnormal, green for normal) and tooltips are used to enhance interpret-ability. The dashboard also includes summary statistics, visualized with pie charts, and provides export options (Excel, JSON). This format allows clinicians and researchers to quickly triage patients, track recommendations, and analyze condition distributions across a population.

– **Usability:** The CDSS system achieves usability through a straightforward tabbed interface, where each tab corresponds to a specific clinical workflow such as patient search, measurement entry, or historical query, thereby minimizing navigation complexity. Context-sensitive tooltips with clinical examples support accurate data entry and reduce cognitive load in fast-paced clinical settings. Input validation ensures correctness while providing real-time feedback using familiar clinical terminology. In addition to the tabbed interface, the system now includes a snapshot-based dashboard that presents patients' clinical states visually, using color-coded indicators and tooltips for toxicity and hematological conditions. This enhances readability and allows for efficient triage and treatment review. The consistent use of standardized LOINC codes supports potential interoperability with other clinical systems, allowing the CDSS to function as a complementary interface within broader EMR workflows.

- **Personalization:** This system does not support client personalization it is not capable of recommending on personalized treatment.

- **EHR Connectivity:** Currently this system does not support external EHR connectivity and is designed to manage EMR records internally.

- **Knowledge Creation and Sharing:** The system utilizes a structured set of internal rule files—including `hematological_rules.json`, `toxicity_rules.json` and `treatment_rules.json` —which encode clinical knowledge in a modular, human-readable format. It does not yet incorporate external medical guidelines or enable automated sharing across systems.

- **Implementation:** This system is a stand-alone system, designed to replace other EMR records management systems, if exists.

- **Compliance / Adherence:** The system includes a rule-based recommendation layer that provides suggested diagnostic interpretations and potential treatment considerations based on patient data. However, it does not currently track whether clinicians follow these recommendations, and thus compliance or adherence is not yet monitored or enforced within the system.

- **Cognitive Models / Bias:** The system operates based on explicit, rule-based logic defined by clinicians, without any use of statistical or machine learning models. As such, it does not introduce algorithmic bias and applies the same logic uniformly across all patients.

- **Effectiveness:** The system offers an organized way of logging and accessing the hospitalization's records, allowing additional analytical systems to use this information for research or for managerial purposes.

- **Evaluation:** There are no regular user evaluations in the system, and the users are responsible for the proper use of it.

- **Regulation:** This system is not yet regulated in aspects of privacy or cyber security, thus does not meet the regulations by the health organization.

- **Economy:** This system is very cheap to deploy, uses local computational resources thus is very cost effective. The potential ROI from this system lies at better monitoring of the EMR records, which has the potential to improve the medical care as well as to standardize the billing process.