USE CASE: Patient overview of medication status that guarantees a quick and individual medication adjustment.

Dynamic and interactive visualization of medication status for patients in a hospital.

Innovation and Complexity Management

Module HI-B-25

Deggendorf Instititute of Technology

European Campus Rottal-Inn (Faculty of Applied Sciences)

Bachelor of Health Informatics: Semester 3

Uchenna Chukwuma-Enyioji

Pfarrkirchen, 22.01.2023.

STRUCTURE FOR DATA VISUALIZATION PROJECTS

This visualization aims to provide a quick and individualized medication adjustment by giving an overview of all patients in a hospital with their associated medication status.

Background: Nurses are under a lot of pressure. Additionally, this has grown more so in Covid-19. Additionally, there is a general lack of trained personnel and a high rate of sick leave among the staff. The nurses are stressed and under pressure as a result.

General Objective: To combat this, we wish to make nurses' daily tasks simpler and more understandable and to increase their sense of security in crucial and intricate working processes. The process of administering medication is a critical component of a ward's operation. Patients may suffer severe effects if the wrong medication is administered or if you neglect to administer it. This places a great deal of duty on nurses.

Problem: There is no overall overview of the medication status of each patient in a ward or hospital in general.

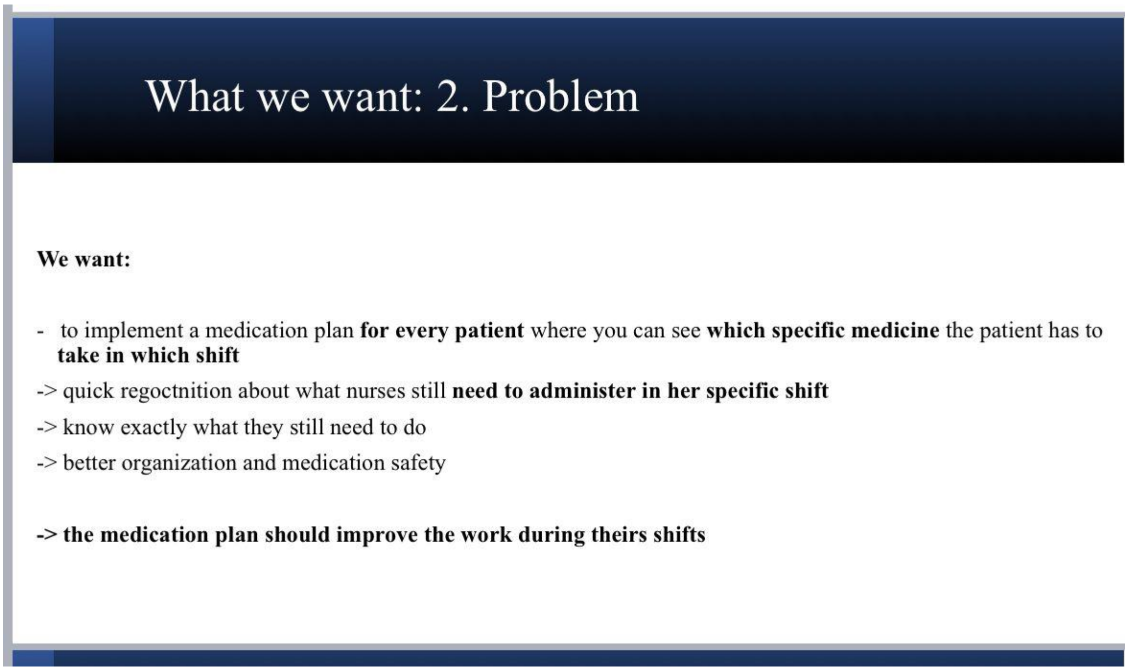
What we want: We want to implement a summary of each patient's medication status. Because nurses need to be able to identify situations rapidly, we wish to incorporate a traffic light system. So that nurses can quickly determine whether a patient's medication is finished or not. Every hospital operates in three shifts. We decided to divide the system into three shifts so that each shift could verify that the drug had been consumed. The entire system is designed to make it simpler for nurses to work.

The design choice for visual encoding and abstraction idiom in this project is a scatter plot. The scatter plot is chosen because it allows the display of multiple variables, such as room number and medication given, on a single graph. Furthermore, the scatter plot allows for the use of color to indicate the medication status of patients, **with red indicating that the medication has been given once or not been given at all, yellow indicating that the medication has been partially given (2 times), and green indicating that the medication has been fully given (3 times).**

1. Context: Domain Situation

The domain of this project is the healthcare industry, specifically the management of medication for patients in a hospital. The stakeholders in this project include patients, nurses, and doctors. The key user of this visualization is a nurse, who is responsible for administering medication to patients and ensuring that they are taking the correct medication at the correct time. The nurse's goal is to have a quick and easy way to view the medication status of all patients and to be able to adjust the medication as necessary if the medication shows that it was not completed.

These use cases were prioritized based on their importance to the key user because there have been problems with delayed medication administration and sometimes the patients end up not taking their medications at the stipulated time because there is no organized view of administration management in a hospital which has resulted in health complications and delayed recovery. That is why the use case was created with the ability to view the medication status of all patients being the highest priority, followed by the ability to adjust the medication for individual patients.



The above image is the need of my group members from Deggendorf.

2. Translation

The medication administration structure from FHIR resources showed the JSON template (https://build.fhir.org/medicationadministration.html):

JSON Template

```
{
  "resourceType": "MedicationAdministration",
  // from Resource: id, meta, implicitRules, and language
  // from DomainResource: text, contained, extension, and modifierExtension
  "identifier" : [{ Identifier }], // External identifier
  "basedOn" : [{ Reference(CarePlan) }], // Plan this is fulfilled by this administration
  "partOf" : [{ Reference(MedicationAdministration|MedicationDispense|
   Procedure) }], // Part of referenced event
  "status" : "<code>", // R!  in-progress | not-done | on-hold | completed | entered-in-error | stopped | unknown
  "statusReason" : [{ CodeableConcept }], // Reason administration not performed
  "category" : [{ CodeableConcept }], // Type of medication administration
  "medication" : { CodeableReference(Medication) }, // R!  What was administered
  "subject" : { Reference(Group|Patient) }, // R!  Who received medication
  "encounter" : { Reference(Encounter) }, // Encounter administered as part of
  "supportingInformation" : [{ Reference(Any) }], // Additional information to support administration
```

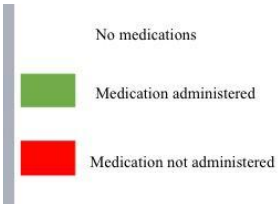
```
  // occurence[x]: Specific date/time or interval of time during which the administration took place (or did not take place). One of these 3:
  "occurenceDateTime" : "<dateTime>",
  "occurencePeriod" : { Period },
  "occurenceTiming" : { Timing },
  "recorded" : "<dateTime>", // When the MedicationAdministration was first captured in the subject's record
  "isSubPotent" : <boolean>, // Full dose was not administered
  "subPotentReason" : [{ CodeableConcept }], // Reason full dose was not administered
  "performer" : [{ // Who or what performed the medication administration and what type of performance they did
    "function" : { CodeableConcept }, // Type of performance
    "actor" : { CodeableReference(Device|Patient|Practitioner|PractitionerRole|
    RelatedPerson) } // R!  Who or what performed the medication administration
  }],
  "reason" : [{ CodeableReference(Condition|DiagnosticReport|Observation) }], // Concept, condition or observation that supports why the medication was administered
  "request" : { Reference(MedicationRequest) }, // Request administration performed against
  "device" : [{ CodeableReference(Device) }], // Device used to administer
  "note" : [{ Annotation }], // Information about the administration
  "dosage" : { // Details of how medication was taken
    "text" : "<string>", // Free text dosage instructions e.g. SIG
    "site" : { CodeableConcept }, // Body site administered to
    "route" : { CodeableConcept }, // Path of substance into body
    "method" : { CodeableConcept }, // How drug was administered
    "dose" : { Quantity(SimpleQuantity) }, // Amount of medication per dose
    // rate[x]: Dose quantity per unit of time. One of these 2:
    "rateRatio" : { Ratio },
    "rateQuantity" : { Quantity(SimpleQuantity) }
  },
  "eventHistory" : [{ Reference(Provenance) }] // A list of events of interest in the lifecycle
}
```

However, this was too complicated for the task we wanted to achieve. The data structure was broken down and simplified to contain patient information, including their name, address, patient ID, medication given, gender, room number, and medication given during different shifts.

The task and data abstraction is to **view the medication status of patients in a hospital.**

3. Implementation

The marks used in the scatter plot are circles, with the size of the circles indicating the number of patients in a room and the color of the circles indicating the medication status. The channel used in the scatter plot is the x-axis, which displays the room numbers, and the y-axis, which displays the medication given. This was used because we only need to show the name of the patient, room number, and medication status, then the scatter plot was filtered to show the room number and the medication status. In the visualization, when you click or hover over a circle that represents a room, you get to see the details of the room occupant which shows the name and details of the medications prescribed.



Name of the patient	Room number	Medication status		
		Early shift	Late shift	Night shift
Maria Huber	102			

The technology stack used for this project includes Python, the Pandas library for data manipulation, the Dash library for creating the web application, and the Plotly Express library for creating the scatter plot. The architecture of the solution includes importing patient data from a JSON file, creating a new column to store the traffic light status based on the medication given, creating a dropdown menu to select the room number, a scatter plot that updates based on the selected room number, and patient information displayed when clicking on a patient's name.

code.py > ...

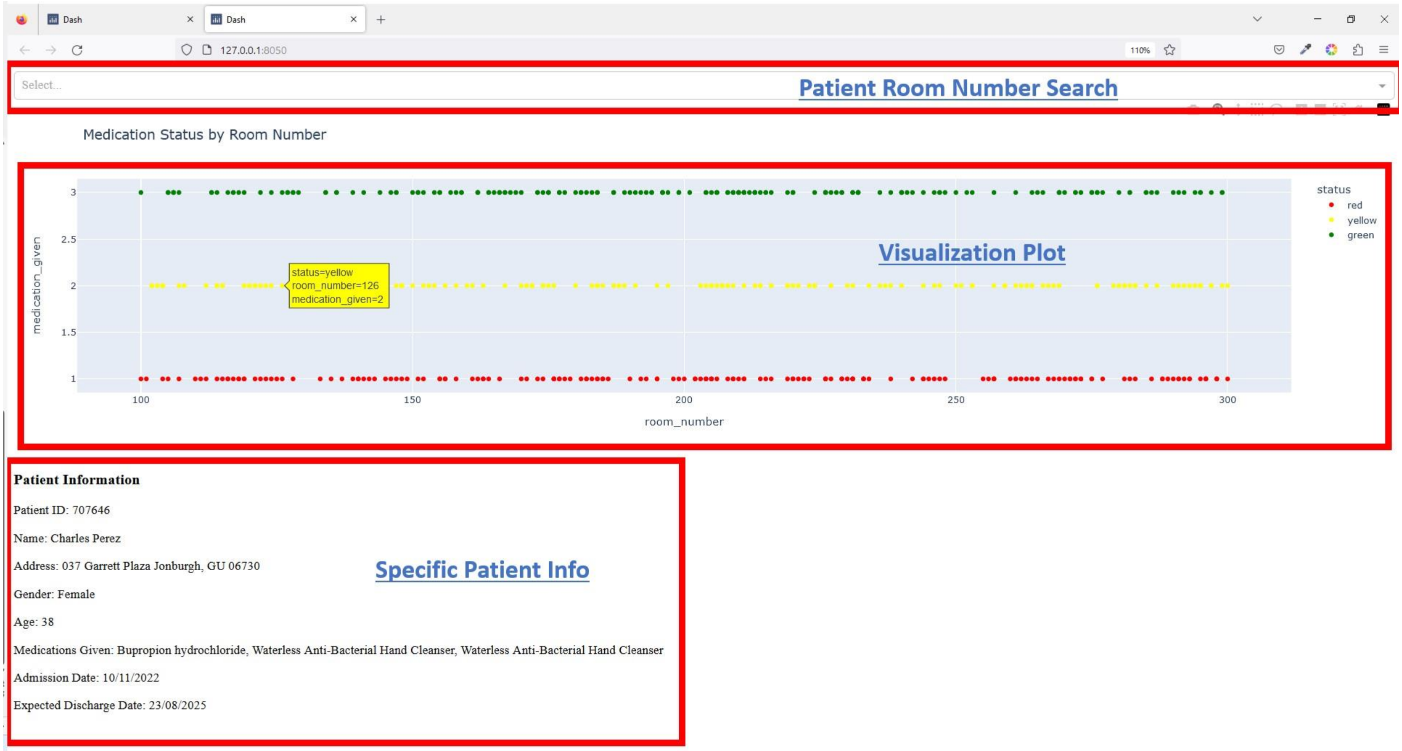
```
import random
import pandas as pd
import dash
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output
import plotly.express as px

import json
```

The scatter plot was chosen for the design choices as it allows for analyzing medication completed or not completed by separating the scatterplot into 3 colors of the traffic light. The red is already separated so that you can see that the circles(representing the rooms) have not completed their medication. This gives nurses insight to focus on those patients in the highlighted rooms.

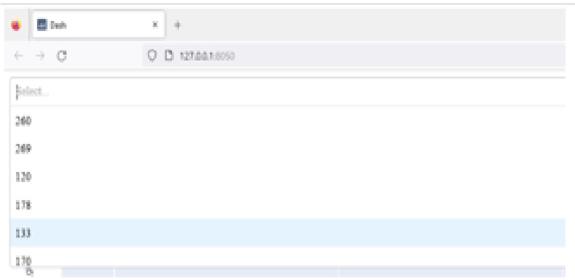
The traffic light status was chosen as it is a clear and easily understandable representation of the medication status. This is very helpful for the nurses to easily identify the rooms and patients that need to be administered their medication to.

The dropdown menu allows the user to filter patients by room number easily. The patient information displayed when clicking on a patient's name includes patient Id, first name, last name, address, gender, age, medications given, admission date, and expected discharge date from the hospital.

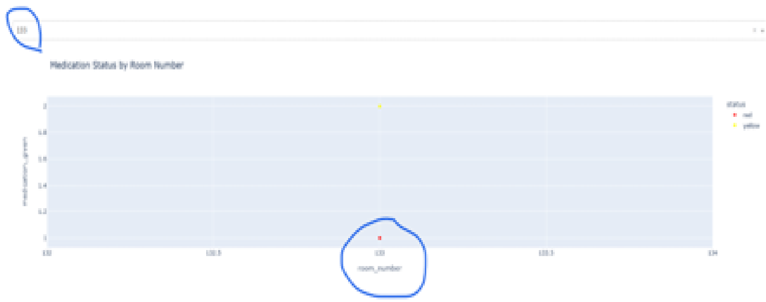
The final product is an executable web application that can be accessed through a web browser. Screenshots of the most important user interactions include the scatter plot with the traffic light status as the marker color, the dropdown menu to select the room number, and the patient information displayed when clicking on a patient's name.



Search Functionality:



Search Result:



Patient Information Hover after search:

Patient Information Hover on general visualization:

The scatter plot allows for the use of color to indicate the medication status of patients, ***with red indicating that the medication has been given once or not been given at all, yellow indicating that the medication has been partially given (2 times), and green indicating that the medication has been fully given (3 times).***

4. Testing

Unit tests were performed to ensure the correct functionality of the code, including testing the data import, data manipulation, and the functionality of the scatter plot and dropdown menu. Results of runtime and memory profiling were also performed to ensure the application is running efficiently. A load test was also performed, with a coverage of a minimum of 5000 and a maximum of 9000, to ensure the application can handle a large amount of data and user interactions.

5. Roadmap

Reflecting on the development process and its outcomes, it is clear that the use of the Visualization Framework by Munzner was effective in creating a dynamic and interactive visualization that met the project's goals.

The achieved goals were:

- The use of the Dash and Plotly libraries made it easy to create a web application that was user-friendly and easy to navigate.
- It was possible to overview the patient's medication status with a traffic light system.
- The web page was made interactive with the search functionality and hovering over the circles showed details of the room occupant.

However, some areas can be improved in future developments. One potential next step would be to add real-time functionality to the application, allowing it to update the visualization automatically as new data is added or as medication is given to patients. There is room for improvement in expanding the patient data to include more information such as adding display of medications not given.

The vision for the product and its architecture is to continue expanding the application's functionality and usability, making it a valuable tool for hospital staff to manage the medication of patients in the wards**. The measure for this goal would be a more detailed medication plan of patients grouped by wards rather than a just overview of all patients' medication status.**

In addition, one can consider the implementation of interoperability with hospital EHR via FHIR. This can help the hospital staff to have a more accurate and up-to-date view of patients' medication status, reducing the risk of medication errors and improving patient care.

Furthermore, one can also consider adding a feature that allows a blinking system to the traffic light to highlight a critical case of missed medication administration thereby showing priority to patients that should be attended to, to avoid health complications.

6. Summary & Conclusion:

In summary, this project aimed to create a dynamic and interactive visualization for managing medication for patients in a hospital ward. The Visualization Framework by Munzner was used to guide the design and implementation of the visualization, which was created using the Python programming language, the Dash library for building web applications, and the Plotly library for creating interactive visualizations. The data was generated using the Faker library, which allowed for the creation of realistic patient data.

The visualization was implemented as a scatter plot, with the size of the circles indicating the number of patients in a room and the color of the circles indicating the medication status. The scatter plot was filtered by room number, and a dropdown menu was used to select the room number. When clicking on a patient's name, patient information was displayed. The web application also allows for the export of patient data to CSV and JSON.

Unit tests were performed to ensure that the application functions correctly, and results of runtime and memory profiling were done. A load test was also performed to ensure that the application could handle a large amount of data and user interactions without any issues.

In future development, the application could be improved by adding real-time functionality, allowing it to update the visualization automatically as new data is added or as medication is given to patients. Additionally, the application could be integrated with an API to share data with other systems and stakeholders.

In conclusion, this project has successfully demonstrated the effectiveness of using the Visualization Framework by Munzner in creating a dynamic and interactive visualization for managing medication for patients in a hospital ward. Creating an online medication plan that guarantees a quick and individual medication adjustment was a success. The final product is an executable web application that allows hospital staff to easily view and manage the medication status of patients in the wards. However, there is always room for improvement and further development to make the application more efficient and useful for hospital staff and patients.