Software Design (CSE 564) MCS Portfolio Report

Kaden Brown
Arizona State University
Tempe, United States
ktbrown6@asu.edu

I. Introduction

Software design is a broad concept that encompasses a variety of techniques and practices used in the derivation and implementation of software systems. This course was focused on the development lifecycle of cyber physical systems — these are simply a collection of computing elements that interact with one another and their physical environment in a feedback loop. This definition is intentionally abstract as cyber physical systems are incredibly diverse and found in a variety of applications in the real world. Cyber physical systems are defined by five critical features, each of which must be present a system:

1. Reactive computation: the system must interact with its surroundings in both the physical and computational world continuously through both inputs and outputs. Through these inputs, the system can determine what behavior it should exhibit, and thus produce outputs.
2. Concurrency: the system must be capable of executing along multiple threads at a given moment rather than on a single, sequential line of computation. This allows for multiple types of input to be read at the same time, or, for multiple data streams to interact with one another, informing decision making.
3. Feedback Control of the Physical World: the system must have some ability to modulate its surroundings given input via its outputs. By reading inputs, the system must be able to affect some behavior reflected in its outputs that may change what its sensors read as input in future executions.
4. Real-Time Computation: the system must observe some connection to how time flows in the real world throughout its own execution. This is required in order to reflect that such a system might be able to perform an actual, physical task while accounting for delays and resource allocation.
5. Safety Critical Applications: the system must value safety over other design objectives. Before considerations such as design cost or computational efficiency can be prioritized, the reliability of the system must first be evaluated to strict standards. Errors encountered with safety critical applications can result in human injury or even death.

These guiding principles define what it means to be classified as a cyber physical system and each must be fully satisfied in its entirety[1]. The first phase of this project was to design a high-level description of a cyber physical system for proposal as a project for eventual implementation. This was open-ended with many potential ideas submitted encompassing topics such as air purification and traffic control, however only four were selected through a student-led voting process. My group prepared and submitted the concept of an automated emergency room triage system which was selected as one of four possible project topics. This system was intended to prototype a collection of medical sensors and processing units that would assist nurses in quickly taking patient vitals and other statistics in order to generate a risk score that could be used to rank patients by the urgency of their conditions. Throughout the semester, this cyber physical system was designed, its components specified using both formal and semiformal practices, implemented in Java, and its performance evaluated using a variety of test scenarios with patients exhibiting different conditions.

II. Solution Explanation

A. *Formal Design*

The formal design phase of this project was conducted through the construction of synchronous reactive components (SRCs). An SRC is a computational unit that has a given input set, output set, state set, initial states, and reaction set. The input and output set are the sets of all values that can possibly enter and exit the component, while the state set determines what variables control its state. Given the state of a component and some input, the output (and subsequent state following a transition) can be determined via the react set which maps inputs, states, and outputs. Each sensor unit was described with its own SRC. Computational units known as "profilers" were used to assess sensor data and score patient risk profiles; four of these were designated as SRCs. A pharmaceutical dispenser unit was designed to administer basic treatments to reduce patient symptom severity as a separate SRC. Finally, a notification system was developed to assist nursing staff based on the conclusions made by the profiler units, along with an application allowing for the quick summary of patient data and an interface enabling personnel to override the system's scoring parameters; each of these three were designated as independent SRCs.

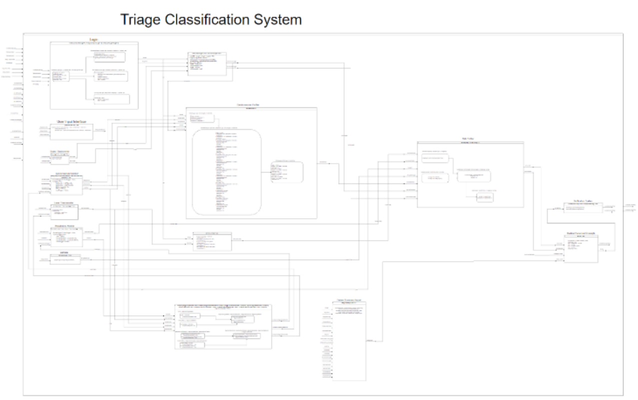


Fig 1. Complete formal design. All SRCs are visualized along with their input and output channels, demonstrating i/o relationships.

“Sensor” SRCs included the following: a login portal, a text-input interface, a cardiovascular monitor, a scale / stadiometer, a laser thermometer, a respiratory monitor, and a camera. Each of these was responsible for collecting one or multiple types of data from a presenting patient and transmitting this along to profilers. The first of these, the login portal, allows patients to choose between three options: logging into an existing account, creating a new one, or signing on as a guest. In each case, the important data here is the age of the patient and their gender which are logged as input regardless of an account’s existence; in the first two options both a username and password field are also required. The second sensor, a text interface, simply allows a patient to record pertinent information, such as how much paint they are in, what they believe they are suffering from, and any other details they wish to convey.

The next four sensors would, in the real world, correspond to physical equipment or devices that would take readings from the patient’s body. The cardiovascular monitor records the patient’s systolic and diastolic blood pressures as well as their heart rate. It simply records these values and pushes them to the rest of the system — this design is consistent among sensor SRCs and can be observed in figure 2.

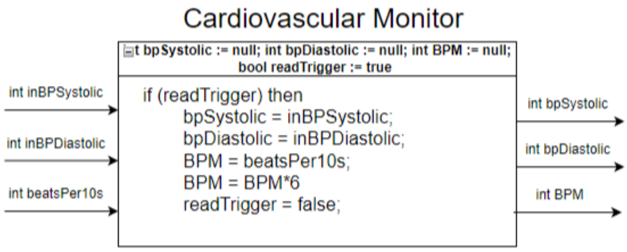


Fig 2. Cardiovascular Monitor SRC.

In this particular SRC, a conversion was necessary between the beats monitored in a 10 second window to how many would them be monitored in a 1-minute window. The second physical sensor, the scale / stadiometer, measures both the weight and height of the patient. This information is then passed on to several profilers. Both data streams are measured in imperial units: pounds and inches respectively. The laser thermometer is only responsible for taking the patient’s temperature using the unit degrees Fahrenheit. The respiratory monitor collects a patient’s blood oxygen partial pressure as well as the respirations per minute. The camera functions in a unique manner: the idea behind this SRC is that a physical camera would take an image of any affliction experienced by a patient for machine-learning recognition, however, due to the scope of this project it was not feasible to actually implement this. In order to replicate what an actual system might be capable of, this prototype system takes a string representative of a “filepath” as input which it then is able to pass to an exterior function in order ascertain the affliction type.

After the sensors of varying types read in all of their respective inputs, they are passed along to the profiler SRCs. The first of these is a BMI profiler which takes input regarding a patient’s age, weight, and gender to calculate not only their BMI, but additionally whether they are classified as overweight or underweight (based on their gender). The second is a cardiovascular profiler which takes data including age, diastolic blood pressure, systolic blood pressure, overweight status, and heart rate. These features are then used to compute a cardiovascular risk score which considers all of these various inputs to determine how likely an individual is to be suffering from a heart-related affliction. One theme in the decision logic here that remains consistent in the next profiler is the synergistic effect that certain traits exhibit; for example, if an individual is both older than 65 and overweight, they have a higher risk than both of these factors had they been simply added together[2]. The next profiler is a fever profiler which takes both age and temperature as input. Depending on the age of the patient, different temperatures will be different levels of cause for alarm. This is described in figure 3 which showcases how the severity is determined.

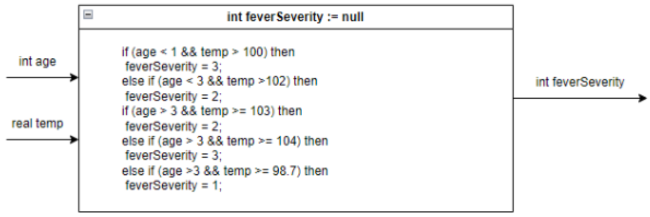


Fig 3. Fever Profiler SRC.

The final profiler is the Risk Profiler, which takes the outputs of the other profilers and joins them to some other values that have not yet been considered including breaths per minute, oxygen partial pressure, symptom type (from camera), and overweight / underweight status. Using these, two statistics are computed: severity and complexity. Severity refers to the risk associated with a patient’s condition and will ultimately be used to rank their priority as they exit the triage procedure. The complexity statistic represents the accuracy of the system’s risk prediction based on several of the inputs. This is necessary because of the safety critical nature of the system — it is unreasonable to expect that the tool will be accurate at all times and due to the wide variety of afflictions that patients present to the emergency room with, some will be more difficult to diagnose than others. The complexity score allows personnel to tell how accurate the risk assessment may be and can inform their decision making accordingly.

The standalone SRC that does not correspond to the sensor, profiler, or notification segment of the system represents the tool’s feedback capability. Known as the PharmDispenser SRC, this component utilizes logical decision making based on its input parameters to dispense medication or basic treatment in order to alleviate serious conditions[3]. Taking the inputs of partial oxygen pressure, systolic and diastolic blood pressure, temperature, and heart rate, the SRC is able to trigger the dispensing of an oxygen mask, blood thinners, water, antipyretic medication, epinephrine, and beta blockers. Each of these is aimed at reducing some potentially critical affliction symptoms. After dispensing one of these (if required), the sensors are re-triggered to monitor the patient to determine if any change has occurred in their symptoms. This happens simultaneously as their previous readings are used to generate a risk profile (which may or may not send a notification to personnel depending on severity). This allows the system to influence the patient’s symptom expression and hopefully lower their urgency level — this may trigger cyclically as a patient waits in order to update their risk value.

The final classification of SRCs were those belonging to the notification functionality. The first of these was the Patient Summary Report SRC. This functions to take all input data from each sensor and generate a readable string that can present the information to a healthcare worker who might want to review a patient case. The purpose of this is to yield some transparency to the system's decision-making process and inform nurses and physicians quickly about all of the acquired information. Next, the Medical Personnel Oversight SRC functions as an access point for triage nurses to assess what the system has decided and alter the risk factor that was automatically set. This is important because it cannot be assumed that the tool will always correctly diagnose patient risk profiles — it may be necessary for an assessment to be conducted manually by industry professionals. This SRC is shown below in figure 4.

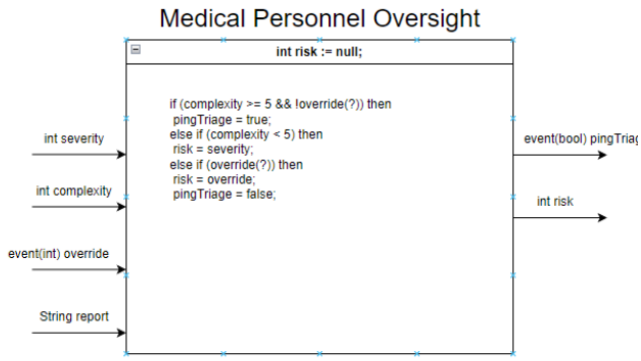


Fig 4. Medical Personnel Oversight SRC.

In the event that a high complexity score is generated by the Risk Profiler SRC, an automatic oversight is required to proceed further. This is because in this case it is assumed that there is a low probability the system has accurately assessed the patient and in order to ensure their safety, a nurse should take control of the case. Otherwise, personnel may choose to manually assess the patient information, but this is not required. The final SRC is the notification system itself which functions on simple logic that uses the risk calculated in previous steps to determine whether a doctor or nurse should be notified, and to what degree. A level one notification refers to an email-like update, while a level two includes a page and a level three results in a phone call.

B. *Semi-formal Design*

The semi-formal design formalism utilized in this project was UML diagraming conducted on the Astah platform. UML diagrams are an effective way of moving from formal SRCs into a format that is closer to how code might actually be implemented, building a framework for adaptation. This comes at the cost of losing some specification and gaining ambiguity — UML provides less in the way of hard constraints on design but gives more flexibility for how an implementation might build some functionalities.

The overall UML diagram for the project is shown in figure 5; this was built on top of the formal design. The structure of this diagram was nearly entirely one-to-one with the SRCs enumerated in the previous section. This means that for each SRC, a unique UML class exists that specifies each variable as well as methods used in that class.

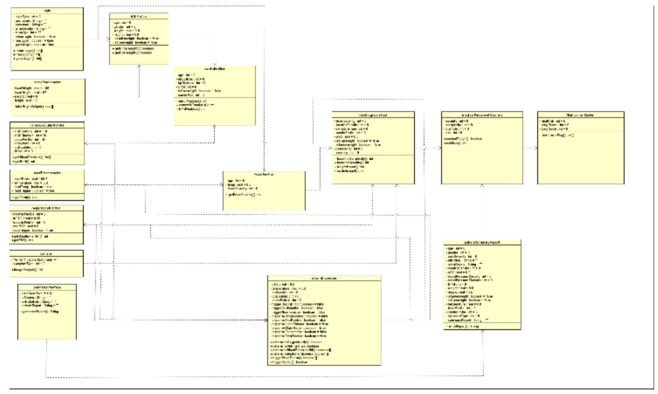


Fig 5. Overall UML Class Diagram.

Each of the connections used here is a dependency relationship as they reflect what needs to be executed, or at least present, in order for a given UML class to properly function. Given an SRC, its corresponding UML will have a private variable declared for all of its inputs and outputs, typically initialized to 0 for numeric values and to an empty string for string values (false for Boolean). These output values will be generated from input values using the methods defined in the input class which are generally designated as public. The level of abstraction can be observed in figure 6, which shows the UML class diagram of the Fever Profiler previously seen in figure 3.

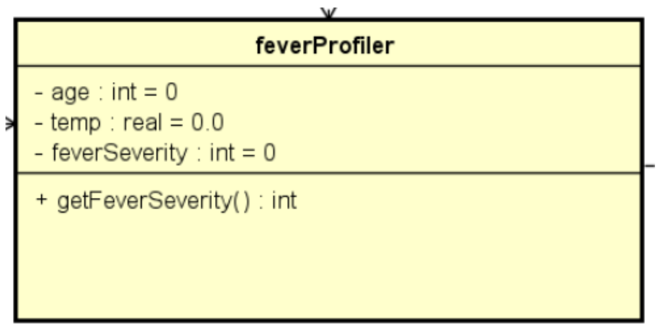


Fig 6. Fever Profiler UML Class Diagram.

Inputs and outputs are each identified as private variables initialized to zero while the logic behind determining the severity is concealed behind a method, "getFeverSeverity". This allowed for an easier conversion into Java code as the implementation section began, though it does not retain the same level of understanding that the SRCs did in the formal design phase.

C. *Implementation*

All of the above UML class diagrams (which in turn correspond to each defined SRC) were implemented in Java as classes. In order to expedite this process and maintain continuity from the semi-formal design, the Astah based diagrams were used to autogenerate framework code. This was then fully fleshed out to meet the expectations given in the requirements design phase and reflected by the individual SRCs. Patients were treated as objects in implementation, each with a substantial dataset corresponding to each of the required inputs to all sensor SRCs. These were supplied to the sensors who passed the values along to the profilers through a variety of method calls at each connection point. These, in turn, called methods to pass their analysis onto the notification segment which then sends out signal pings to either nurses or doctors.

The complexity of the classes varied significantly with most of the sensors being relatively simple and the profilers significantly larger. On average, the profilers required as much as 5-6x as much code in order to account for the complicated logic governing their decision-making processes. The pharmDispenser was perhaps the most complex to implement in that it had to have the ability to trigger several sensors independently (resetting a portion of the system) while simultaneously readying the risk diagnosis component. The notification section and its three components were similar to the sensors in complexity and did not pose significant complications.

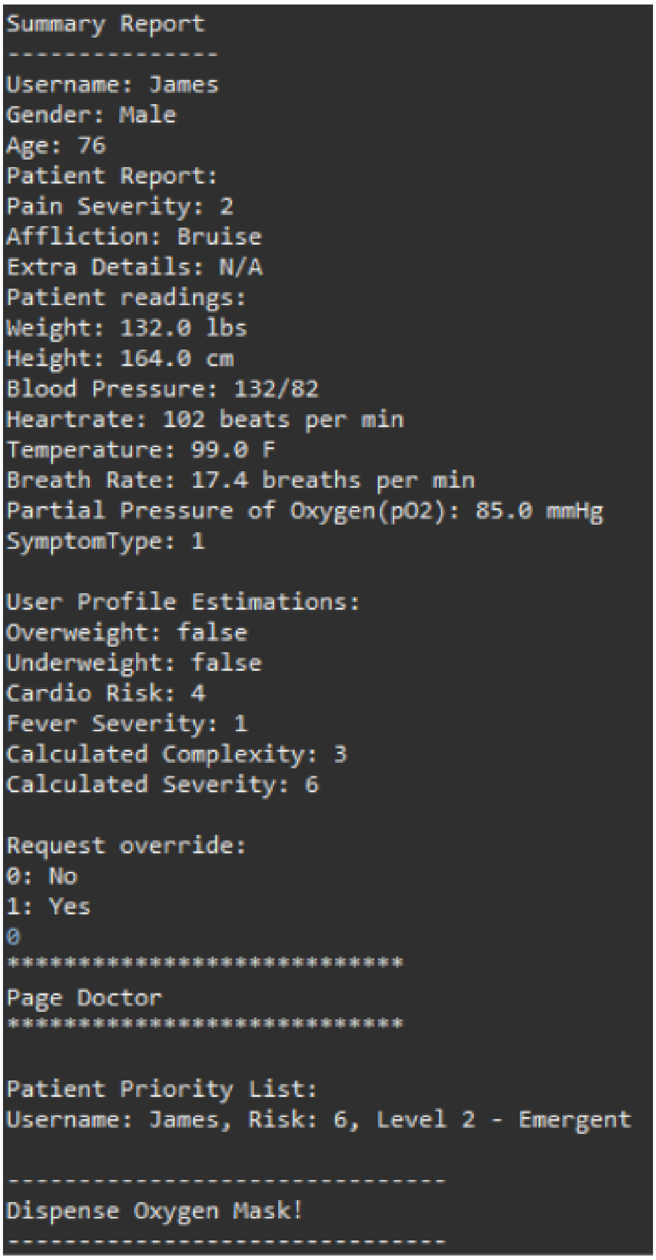
III. Results Description

In order to assess the performance of the system, seven input trial cases were drafted to see what risk factors and complexity factors would be calculated. It was also important to trial the pharmaceutics dispensing functionality to make sure that this was working correctly, granting the correct medications in the proper cases. Finally, the notification system needed to be validated in order to show that the correct staff were being pinged at accurate levels given the patient's presenting conditions.

The first trial case involved an elderly male at a healthy weight with high blood pressure and a low-grade fever. He also showed signs of bruising, though no difficulties breathing were observed. Most significantly, though, was a low partial oxygen pressure. The system output is shown in figure 7 along with the full patient report and output notification, which in this case was a page to a doctor. Given the partial pressure of oxygen falling at 85.0 mmHg, this is an emergent case and thus the doctor is notified with a tier 2 alert. Additionally, it can be observed that an oxygen mask was granted to the patient in an attempt to raise the pO2. This test case demonstrates the functionalities for diagnosing respiratory related conditions functions, as well as the notification and feedback features.

Another trial case involved a 3-year-old female patient weighing 30 pounds with a low blood pressure but normal heart rate. She presented with a severe fever and low pO2. The system was able to correctly dispense both an oxygen mask as well as an antipyretic. Though her blood pressure was low, due to her healthy weight and age a low cardiovascular risk was assigned, as it should have been. Given the moderate severity of the case and minimal complexity, a nurse tier 3 ping was issued, resulting in a phone call. This case demonstrates the ability of multiple factors to be simultaneously recognized by the risk profiler SRC as well as the pharmaceutical dispenser.

Seven full trials were executed in this manner, guaranteeing the functionality of each of the sensors independently of one another. The risk and severity profiles generated by the system were consistent with estimations observed in external research [4]. Furthermore, the profilers were each validated manually given sample inputs to determine that the expected values for BMI, cardiovascular risk, fever risk, and overall risk were calculated. The pharmaceutic dispenser was thoroughly tested to ensure that each possible treatment method could be reached. Finally, the notification components were assessed to guarantee that the correct pings were issued to the correct personnel on a case-by-case basis. The override feature was also simulated, though a random value was assigned in the case of an "override" occurring as no medical professional was available for consultation into the trail cases. This would obviously not be the case if the system were to be deployed, however the testing of the project here is simply to validate that the software components function as intended.



```
Summary Report
---------------
Username: James
Gender: Male
Age: 76
Patient Report:
Pain Severity: 2
Affliction: Bruise
Extra Details: N/A
Patient readings:
Weight: 132.0 lbs
Height: 164.0 cm
Blood Pressure: 132/82
Heartrate: 102 beats per min
Temperature: 99.0 F
Breath Rate: 17.4 breaths per min
Partial Pressure of Oxygen(pO2): 85.0 mmHg
SymptomType: 1

User Profile Estimations:
Overweight: false
Underweight: false
Cardio Risk: 4
Fever Severity: 1
Calculated Complexity: 3
Calculated Severity: 6

Request override:
0: No
1: Yes
0
****************************
Page Doctor
****************************

Patient Priority List:
Username: James, Risk: 6, Level 2 - Emergent

-------------------------------
Dispense Oxygen Mask!
-------------------------------
```

Fig 7. Example patient test case output.

IV. Individual Contributions

My primary area of contribution was towards the design, both formal and semi-formal, implementation, and testing of the login sensor, cardiovascular sensor, laser thermometer sensor, fever profiler, cardiovascular profiler, and PharmDispenser components. I fully designed the SRCs for each of these, writing their input and output sets along with their state sets and initialization values. I was responsible for determining their react sets and in what way they would be connected to the other SRCs that I did not directly design alone. In order to do this, I completed substantial research regarding acceptable vital standards for blood pressure, heart rate, and body temperature based on a variety of demographic characteristics [5] along with research on commonly prescribed drugs for conditions such as high blood pressure and low blood oxygen [3]. After building these components in their formal specification, I created their corresponding UML class diagrams. In the implementation phase I was responsible for writing the Java classes for each of these components along with their corresponding methods. Finally, I wrote and executed 3 of the 7 trial executions that were used to assess the effectiveness and functionality of the system at the conclusion of the implementation phase.

While I was not individually responsible for the writing of the BMI profiler and the risk profiler, I participated in the drafting of their logic. With regard to the risk profiler, I designed the formal specification involving the use of separate complexity and severity statistics that would ultimately be used to predict patient urgency at the end of system execution.

Additionally, I drafted the overall mappings of the entire formal specification, detailing which components would be connected to one another and what information / data would flow along these connections.

V. Skills Acquired

Throughout this project, I significantly expanded my understanding of the software design process. Prior to this, I had not worked on a system of such breadth requiring so many input types representing vastly different data. The integration of these various kinds of information was initially difficult, however through continuous research I was able to create a logical system for reconciling them with one another to produce sound output.

From the formal design phase, I was able to conceptualize the importance of solidifying requirements and assumptions early-on. While this was done in most development areas, there were several key places where this was not immediately nailed down, and it made development in the semi-formal phase very difficult. Eliminating complexity is the most simple at this stage, or so it seemed in the completion of this project — this demonstrates the need of effective design from the ground up in order to proactively counter requirements drift and implementation difficulties later on.

Another interesting point of learning throughout this project was how easy it was to lose sight of the overall system design when working in a small modular environment. For example, when designing the pharmaceutical dispenser, I did not initially consider how building such a feedback component would impact all of the processing occurring downstream — this required me to meet with the team members focused on the risk profiler SRC to decide how we wanted to alter the initial formal specification in order to allow for repeated re-analysis at this stage.

On a more technical level, I have certainly improved my ability to concisely write both formal design specifications as well as code. My understanding of how SRC react sets operate in relation to the state set was improved as a result of designing my own SRCs which was enormously helpful on the exam. Through the implementation of UML diagrams as actual code, I was able to see firsthand how the ambiguity of semiformal design can lead to inconsistencies in physical code from the initially intended formal design.

VI. Aknowledgements

This project was completed in groups of 2-4. My group contained three other team members including Parker Hoang, Ryan Hoang, and Animesh Gupta. The team worked well together and all members completed their share of the final work.

VII. References

[1] R. Alur, (2015), Principles of Cyber-Physical Systems, MIT Press.

[2] WB;, A. K. M. O. P. M. W. P. W. K. (1991, January). *Cardiovascular disease risk profiles*. American heart journal. Retrieved April 18, 2023, from https://pubmed.ncbi.nlm.nih.gov/1985385/

[3] Lacorossi L, Fauci AJ, Napoletano A, et al. Triage protocol for allocation of critical health resources during Covid-19 pandemic and public health emergencies. A narrative review. *Acta Biomed*. 2020;91(4):e2020162. Published 2020 Nov 10. doi:10.23750/abm.v91i4.10393

[4] Soster CB, Anschau F, Rodrigues NH, Silva LGAD, Klafke A. Advanced triage protocols in the emergency department: A systematic review and meta-analysis. Protocolos de triagem avançada no serviço de emergência: revisão sistemática e metanálise. *Rev Lat Am Enfermagem*. 2022;30:e3511. doi:10.1590/1518-8345.5479.3511

[5] Jatoba A, Burns CM, Vidal MCR, Carvalho PVR. Designing for Risk Assessment Systems for Patient Triage in Primary Health Care: A Literature Review. *JMIR Human Factors*. 2016;3(2):e21. doi:https://doi.org/10.2196/humanfactors.5083