

IT System for Managing Hospital Emergencies

Software Architectures' Project

Tommaso Bredariol - 908883

Melany Brianese - 907343

Task 1: Katas for the IT system

Description

The IT system is designed to manage the emergency section of a hospital. The objective is to ensure rapid patient triage, efficient coordination between medical personnel, real-time patient tracking and resource management.

Users

Users will be on the order of hundreds among doctors, nurses, patients, administrative staff, and system administrators.

Requirements

The requirements for the IT system are the following:

1. Doctors and nurses need quick access to patient records, real-time updates on patient conditions and tools for task coordination. To address these needs, the system will include a centralized dashboard that displays patient queues, triage levels, and assigned tasks. Additionally, a task assignment tool will be integrated into the dashboard to facilitate communication and coordination.
2. Patients need real-time updates on their status, the ability to schedule appointments, and access to their medical records. A patient portal will be developed to provide a responsive and interactive user interface. Real-time updates will be delivered, ensuring patients are informed about their treatment progress.
3. Administrative staff require tools to monitor hospital efficiency, manage resources, and generate reports. An admin dashboard will be created, featuring analytics and reporting tools. A built-in admin panel will be customized to manage resources such as beds, rooms, and medical equipment.
4. System administrators are responsible for ensuring system uptime, security, and reliability.

Additional context

1. The system will include a form-based interface for rapid patient data entry.
2. The user interface will be designed with usability in mind. The task must be accomplished without difficulties even for first-time users given the sensitive matters they are dealing with.
3. Automated alerts will be implemented to notify doctors and staff about critical patient conditions.

Task 2: Architecture characteristics

Deployability

The system will be designed for quick and seamless updates. Continuous Integration and Continuous Deployment will be achieved with the use of GitHub, which will also allow testing and deployment. Docker containers will ensure consistent environments across development, staging and production (see requirements 1 and 2).

Reliability

System availability is critical for hospital operations. Redundant servers and database replication will be implemented to ensure 99% uptime (see requirements 1, 2, 3 and 4)

Scalability

Load balancers and distributed databases will be used to ensure scalability (see requirements 1, 2 and 3)).

Fault tolerance

The system will continue functioning despite hardware or software failures. Redundant systems and fail over mechanisms will be implemented (see requirements 1 and 2).

Simplicity

The system will be designed for ease of understanding and maintenance. Clear documentation and a modular design will ensure that developers can quickly understand and update the system (see requirement 4).

Modularity

The system will be built using a microservices architecture, with separate components for user management, data storage, and monitoring. API-driven interactions will ensure that modules can be updated independently (see requirements 1,2 and 3).

Testability

The system will be thoroughly tested to ensure functionality before deployment. Automated tests, including unit tests and integration tests, will achieve 90%+ test coverage (see requirement 4).

Architecture's details

- Frontend: HTML and css with Javascript and REACT
- Backend: Django with Python
- Database: SQL