OBJECT-ORIENTED ANALYSIS AND DESIGN

Directed Emergency System

System Design Document

27 December 2022

Table of Contents

1 Introduction

We have defined all the requirements of the system in the scenario we have established. We designed a simple interface. We have designed our system to be fast and practical in case of emergency. We created the designs and drawings according to all the evolutions.

1.1 Purpose of the System

The aim of our system is to deliver patients to the hospital as soon as possible by providing fast and practical communication, especially in major emergencies.

1.2 Design goals

Design goals for DES;

Usability : It must be usable for everyone even with someone who never used an application before.

Reliability : There should not be any errors for updates will always occur.

Performance : The incident must be reported in 3 minutes.

Response time : Most of the incidents from anywhere in , anytime can be reported.

Supportability : At least 10 kinds of emergency situations can be reported and dealt with.

Quality : Usage of navigation must be without %95 error.

Fault Tolerans: DES should be fault tolerant to loss of connectivity with the routing service.

Modifiability: DES should be modifiable to use different routing service.

Performance, fault tolerans, response time and supportability are high level priority for DES. Usability, reliability, quality are medium level priority for DES. Modifiability is low level priority for DES.

If we look at the trade offs;

Cost vs Quality

If we want our system to work as fast and with as few errors as possible, this increases the cost.

Functionally vs Usability

We aimed to make our system easy to use. Thus, in general, people will not have problems with the interface while using the system but the more functions there are in the system, the more it conflicts with this situation. This again creates a trade-off.

2 Current Software Architecture

The 'SOS Emergency and Safety app' application is similar to ours. While our application notifies the appropriate hospital's location to the ambulance in an emergency, 'SOS Emergency and Safety app' sends the most appropriate police to the location in the application. The working architecture is as follows. User will download application first. Then the user will activate the SOS status in an emergency. Then the police will go to the user's location.

Advantages

+The biggest advantage is quick response to the emergency.

+Saves time and keeps other cops ready for other incidents.

Disadvantages

- Where there is no signal the application fails.

3 Proposed Software Architecture

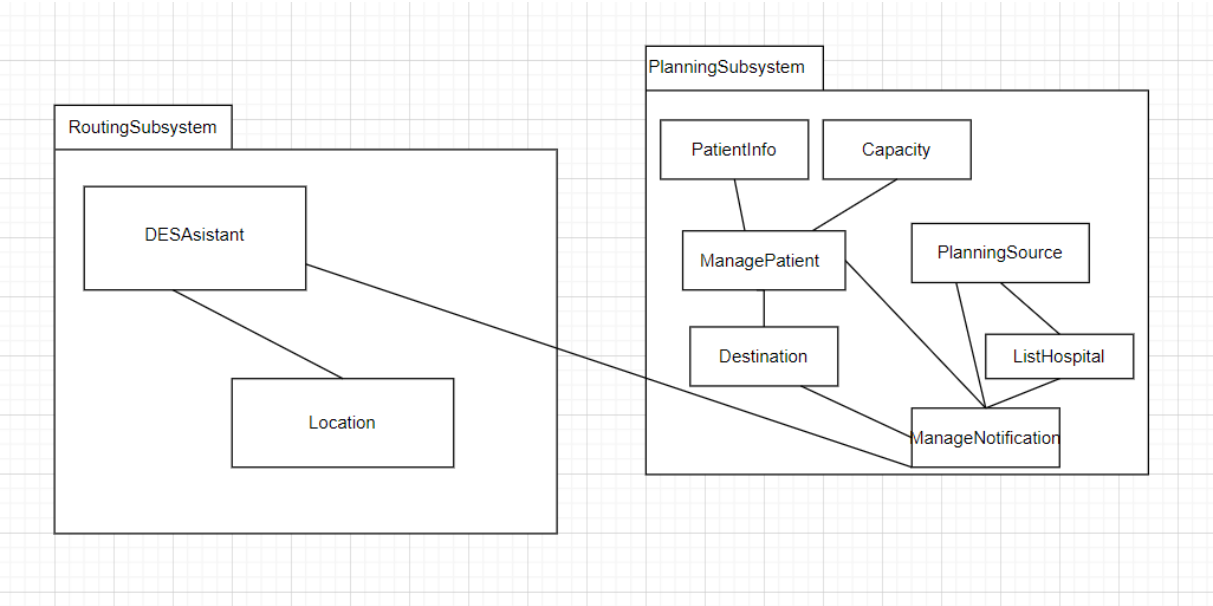
3.1 Subsystem decomposition

The PlanningSubsystem is responsible for the stages of finding a hospital according to the incoming emergency notification, notifying its location and delivering this information to the necessary people. The PlanningSubsystem is also responsible for notifying to RoutingSubsystem from ManageNotificiation.

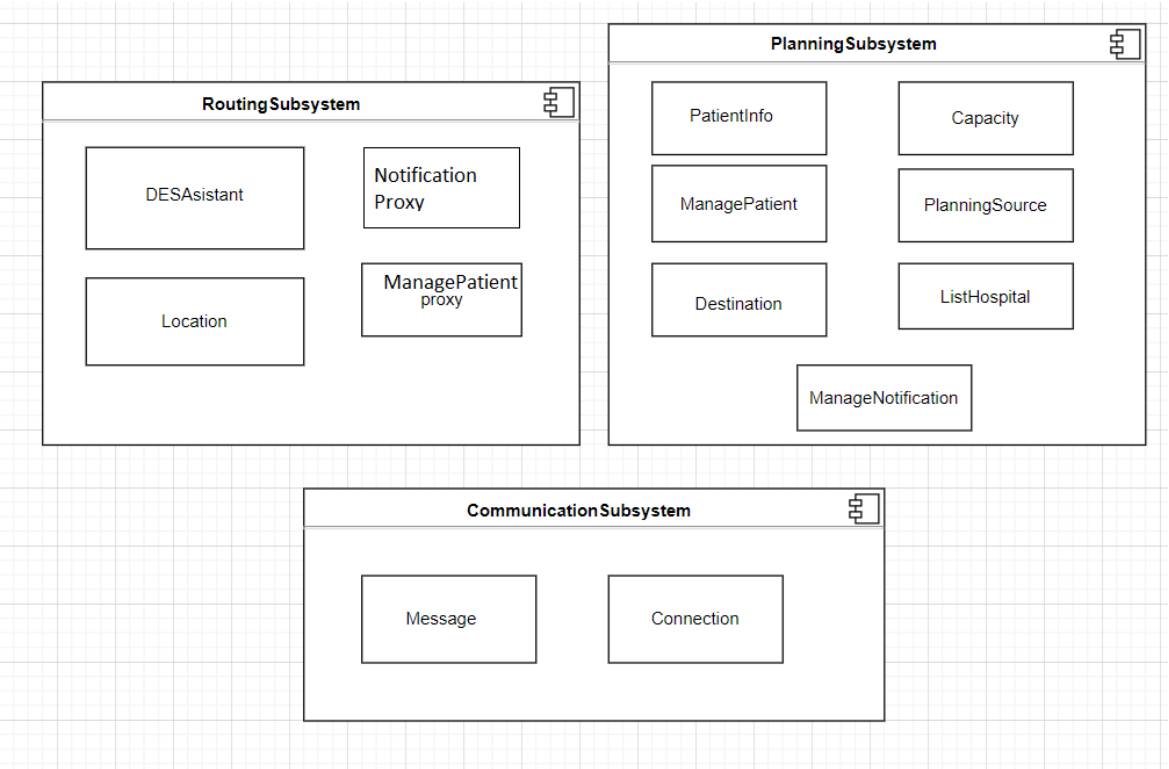
The RoutingSubsystem is responsible for give location information based on ManageNotification.

Our system uses client-server architecture. Client-Server architecture has a requester and a provider. The client is the party requesting the information. Client-server architecture, architecture of a computer network in which many clients (remote processors) request and receive service from a centralized server (host computer). Client computers provide an interface to allow a computer user to request services of the server and to display the results the server returns. Servers wait for requests to arrive from clients and then respond to them. Ideally, a server provides a standardized transparent interface to clients so that clients need not be aware of the specifics of the system (i.e., the hardware and software) that is providing the service. Clients are often situated at workstations or on personal computers, while servers are located elsewhere on the network, usually on more powerful machines. This computing model is especially effective when clients and the server each have distinct tasks that they routinely perform. In hospital data processing, for example, a client computer can be running an application program for entering patient information while the server computer is running another program that manages the database in which the information is permanently stored.

We chose it because we think it is the most suitable architecture for the analysis model that we have established and is frequently used in hospital information processing and management processes.

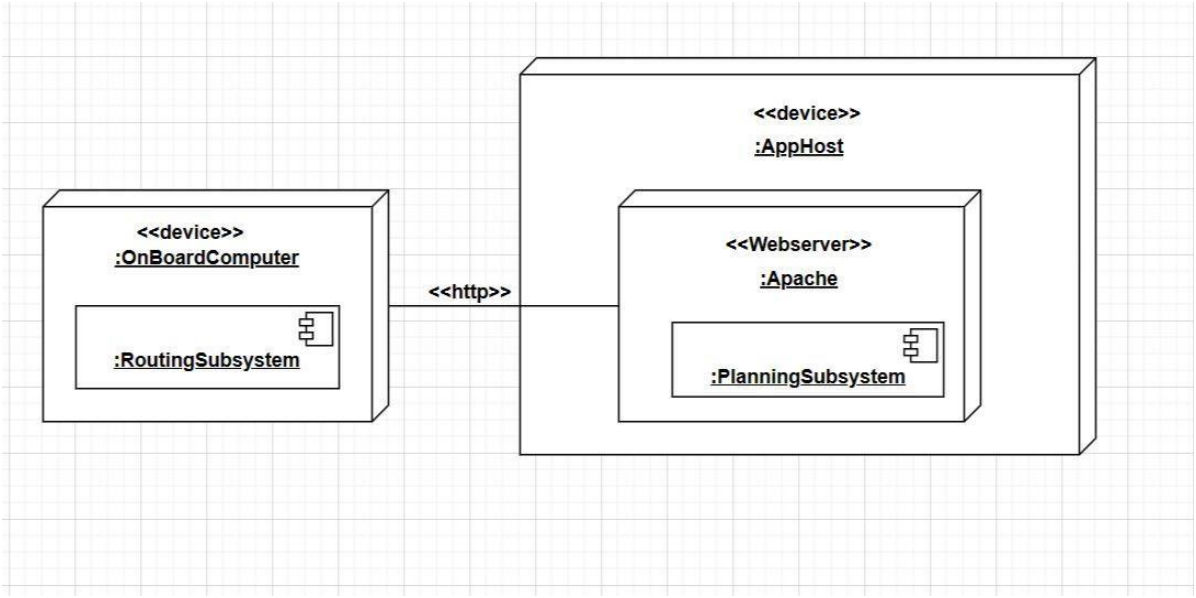


Initial Subsystem Decomposition



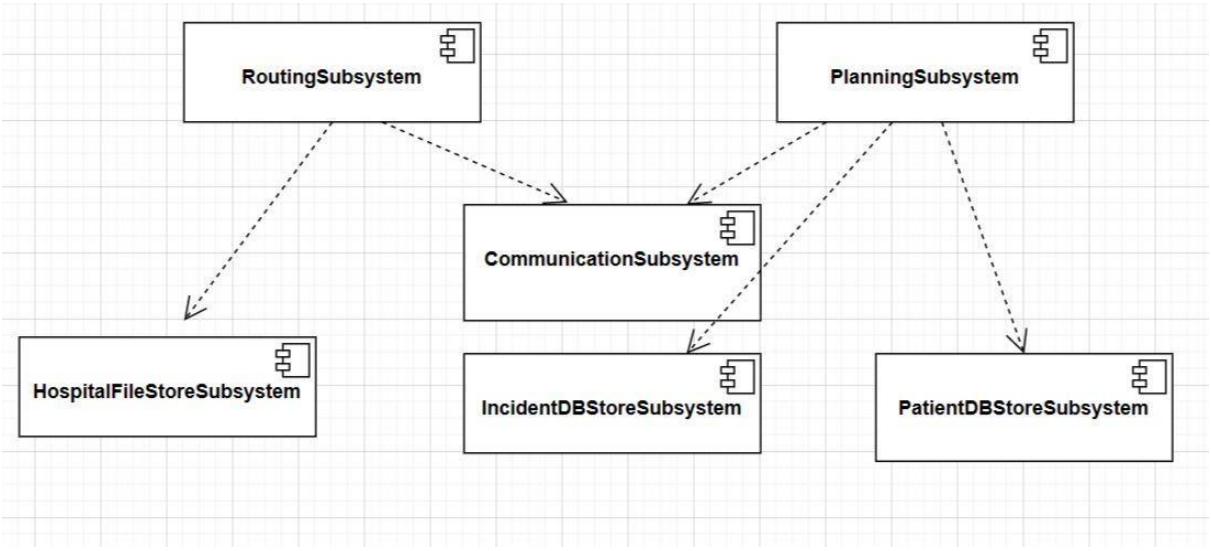
Revised design model for DES.

3.2 Hardware/software mapping



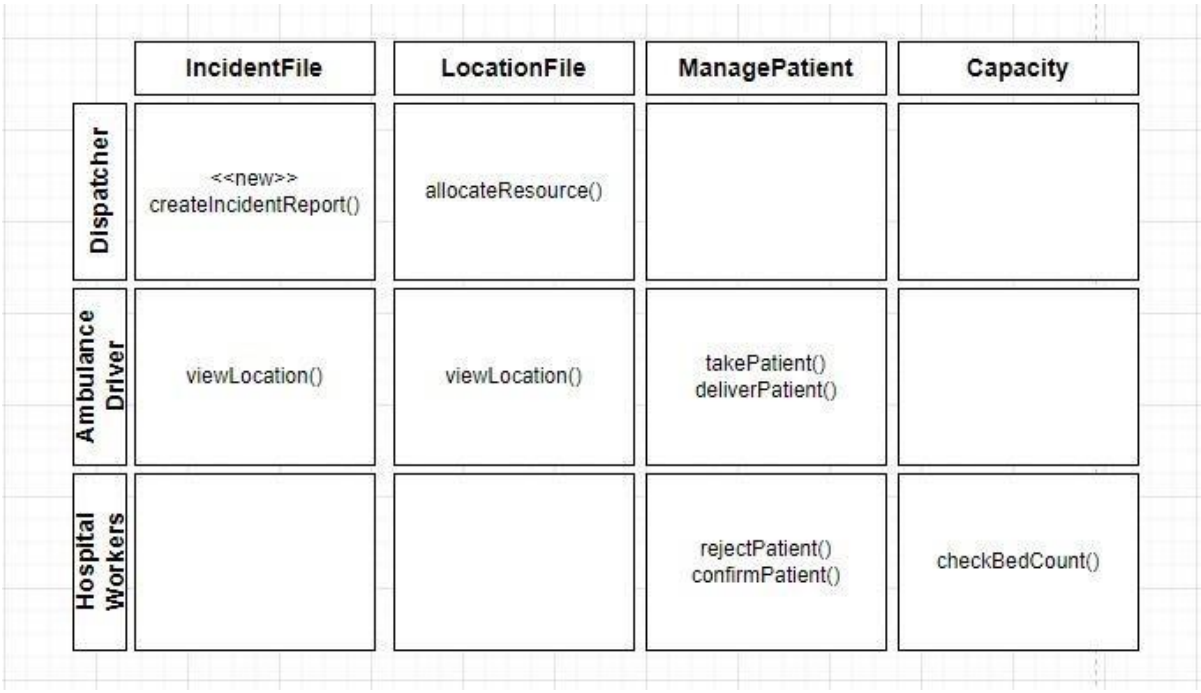
3.3 Persistent data management

After the system is turned off, we may want some information to be kept, which allows us to keep the information. We used both filesystem and database system in our project. Filesystem is more useful for storing small information. A database, on the other hand, stores more complex and large-sized information. So we thought it would make sense to keep the hospital list in filesystem. Because the hospital list information is generally similar and does not take up a very large area. But since the incident file and patient information take up a lot of space, if we keep it in the database, we can easily access it with queries when we want to access information.



3.4 Access control and security

Since there are more than one actor using our system and these actors have different functionality and different access rights to data, we used an Access matrix when describing the system.



3.5 Boundary conditions

ManagePatient: It is the new boundary use case that holds and manages patient documents. Includes patient information and locations.

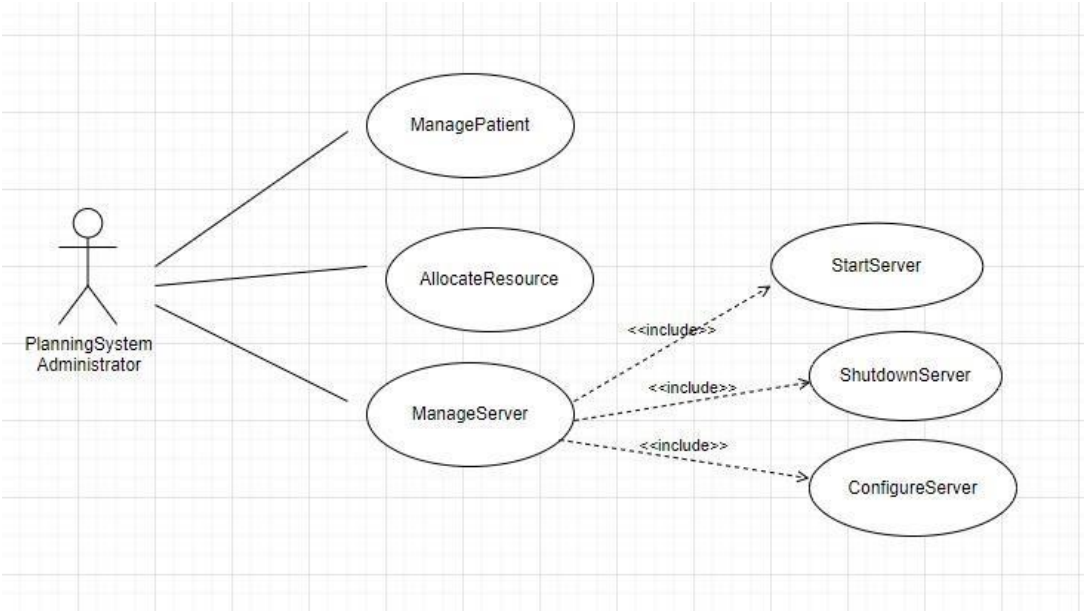
AllocateResources: It is the new boundary use case that directs and manages resources. Contains information on patient and hospital locations.

ManageServer: ManageServer includes all the functions necessary to startup and shutdown the PlanningSystemServer.

StartServer: It is used to start the server.

ShutdownServer: It is used to shut down the server.

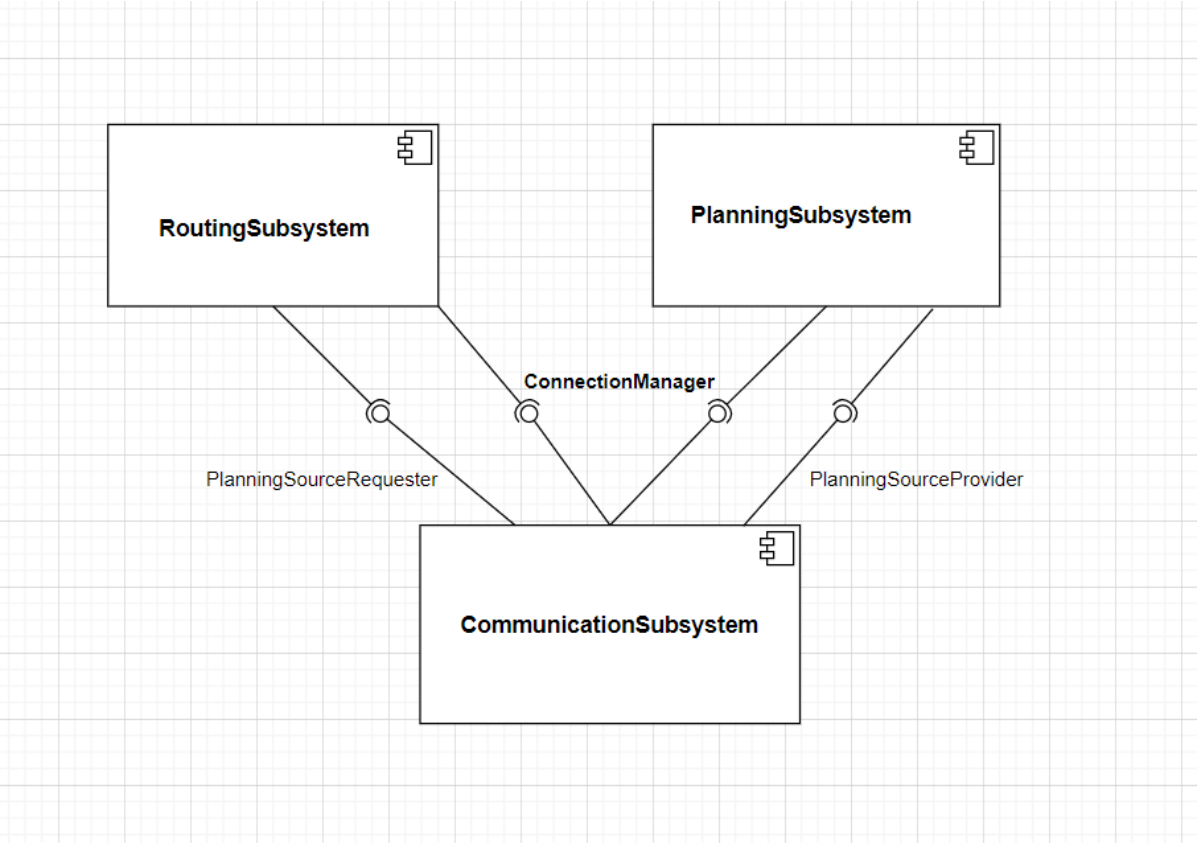
ConfigureServer: It is used for configurations that need to be made on the server.



4 Subsystem Services

UML uses a "lollipop" to denote an interface, which can be appended to classes and subsystems, among other things. We created the communication subsystem to establish the communication between the other 2 subsystems. Our routing and planning subsystems are consumer. Communication subsystem is provider. In the planning subsystem, the main part of the work is taken care of. It has processes such as routing according to the response from the process of communicating with the hospital, handling patient transport, providing resource management. It has the service of continuing the process by transmitting the information it has obtained to other systems. The communication subsystem provides the necessary messages and notifications from 2 subsystems to be delivered between each other. The routing subsystem, on the other hand, serves to show the location of the hospital where the assistant is used according to the future messages and to deliver the messages to the planning subsystem with the communication subsystem.

The CommunicationSubsystem provides three services for managing connections, managing patients and managing planning.



5 Glossary

DesAssistant	DES assistant is responsible for notifications management and hospital locations.
Location	The hospital location that the ambulance driver sees on the GPS screen.
Capacity	The number of vacant beds in the hospital
ManagePatient	It is the one that holds and manages the information of the hospital
PatientInformation	It is the class where the patient's information is kept.
Destination	The destination is the hospital location
ManageNotification	It is the class that manages notifications
PlanningSource	It is the class that manages the resources.
ListHospital	It is the class in which nearby hospitals are listed
AppHost	It is the host of the DES application
OnBoardComputer	Identifies the screens used in the system

RoutingSubsytem	It is the subclass in which routing occurs
PlanningSubsytem	Server that provides information from a Database
CommunicationSubsystem	It provides three services for managing connections, managing patients and managing planning.
AllocateResources	It is the new boundary use case that directs and manages resources. Contains information on patient and hospital locations.
ManageServer	It includes all the functions necessary to startup and shutdown the PlanningSystemServer.
StartServer	It is used to start the server.
ShutdownServer	It is used to shut down the server.
ConfigureServer	It is used for configurations that need to be made on the server.

6 References

https://codecanyon.net/item/sos-emergency-and-safety-app-for-everyones-safety-worldwide-with-phone-protection-feature/35884969?gclid=CjwKCAiAnZCdBhBmEiwA8nDQxdosohL4d1ceKKuL3Sh4fNQxmnthbf4TT7R2709r3bDmTUAOwNqIqRoCPaoQAvD_BwE

https://www.britannica.com/technology/client-server-architecture

draw.io