Short problem description

The problem is about the development of a medical application which should facilitate and smooth the whole blood donation process.

The application may be composed of the following entities:

Donor -> Donation Center Personal -> Doctor -> Administrator -> Owner

The **owner** is a super administrator and above that he should be able to:

* Login
* Add administrators
* Remove administrators

The **administrator** is a manager of the application and is able to:

* Login
* Add doctors
* Disable doctors
* Add donation centers
* Disable donation centers
* Add personnel to donation centers
* Disable donation centers personal

The **doctor’s** work should be eased by assuring that the management of blood requests is a piece of cake and also by making the track of blood request come naturally even for a novice user. He is able to:

* Login
* Update profile
* View patients
* Add patients
* Choose patients
* Dismiss patients
* Change patients status (alive/dead)
* Request blood from donation centers
* Cancel blood requests from donation centers
* Track blood request from donation centers
* Return requested blood to donation centers

The **donation center personal’s** work is ensured to be lightened by easy management of resources (containers) and by utterly efficient management of blood requests. He is able to:

* Login
* Update profile
* Redirect blood requests
* Process blood requests
* Redirect blood requests
* Request blood donations
* Evaluate donation form
* Add blood supplies
* Remove blood supplies
* Send blood donation results

The **donor** is encouraged to donate by being shown the physical benefits of blood donation and moreover by the possibility of checking at any time their donation results by means of blood analysis. He is able to:

* Sign up
* Login
* See reasons to donate
* Update profile
* View donations history
* Apply for donation
* View application status
* Cancel application
* View analysis status

Overall the application will provide a ranking system for blood requests from donors,

a ranking system for blood requests from donation centers,

rank donation centers by relative distance and availability and

encourage donations.

The application assures: security(personal data stays private, users have specific permissions), accuracy(reliable ranking systems), performance(real-time interaction - max 5 seconds), low cost(respect the budget), usability(user-friendly UI/UX), reusability(well defined project structure, clean and maintainable code).

The project was developed in **five** iterations:

First iteration:

* Wide discussion of the project(problems we may encounter, technologies and tools which we will use, design of the application, architecture of the application and so on…)

Second iteration:

* Team brainstorming about diagrams learned
* Work at the required diagrams as sub-teams(tasks divided evenly)

Third iteration:

* Split evenly the tasks(in means of complexity) for the user-interface(Pages)
* Bind the user-interface with the business logic(View Models)
* Implement the interface fields validation

Forth iteration:

* Split evenly the tasks(in means of complexity) for the business logic(View Models)
* Implement functionalities of the business logic

Fifth iteration:

* Testing the most important methods of the application
* Writing the application’s documentation