

# SOP Report 1

## 1. Project title: Remote network management system

## 2. Team members:

- Chelariu Andrei DSWT1B
- Meleca Vasile DSWT1B

## 3. Short description of the project:

The project provides a way for the system administrator to manage remotely a network. More precisely, he will be able to add or remove devices from a network. To do this, he will manage the DHCP server via a web interface.

The project will also offer regular users the possibility to make requests to the system administrator to join the network. The users will be notified when their requests will be processed, via a web interface.

## 4. Application features:

- Single sign on: the system administrator will log in once on the system and then manage different services: DHCP, FTP, HTTP. For the time being, we will only implement management for the DHCP service. If time allows, we may add the possibility to manage other services.
- Secure management of the DHCP server: the management of the DHCP server will not be exposed to the whole network. To perform changes on the DHCP server, the administrator will issue commands on a RESTful service. The service will communicate with the DHCP daemon via inter process communication.
- Notifications: The system administrator and the users will be notified when there is a new request or when the status of a request has changed.

## 5. Application components:

### 5.1 TicketService

- service responsible for management of tickets in the system.
- when the user logs in, he receives a granting ticket. With this granting ticket, he can then request a ticket to access a particular service (user service → to access users; request service → to access the requests)

- POST /TicketService/grantingTickets → create a new granting ticket. The HTTP request will contain the user name and the password. If everything went OK, the user receives as response the granting ticket and a session key used to encrypt the communication between the user and the desired service.
- DELETE /TicketService/granting\_ticket → remove the desired granting ticket
- POST /TicketService/tickets → create a ticket for a particular service. The user will supply the granting ticket and the name of the desired service. If everything went OK, the user receives the ticket to access the service.

## 5.2 RequestService

- service responsible for the management of user services.
- when a regular user wishes to join the network, he will submit a request to the system administrator. The request will contain the MAC address of the device.
- provides a persistence layer to store the requests (MySQL, sqlite, etc).
- GET /RequestService/requests?ticket=t → gets the list of requests. The client of the service, must first get a ticket from the TicketService.
- PUT /RequestService/request/x?ticket=t → override the request with id x. If the request with id x does not exist, it will return an error.
- POST /RequestService/requests?ticket=t → appends a request to the list of requests. If everything went OK, the response will contain the URI of the newly added request.
- DELETE /RequestService/request/x?ticket=t → delete the request with id x.

## 5.3 UserService

- service responsible for the management of users.
- provides a persistence layer to store the user accounts (MySQL, sqlite, etc).
- GET /UserService/user?username=u → get the user with the specified user name. If the user exists, it will be returned, otherwise error will be reported (code 404).
- POST /UserService/users → add the specified user to the table of users. The user name, password and ticket will be provided by the service client in the HTTP request.
- PUT /UserService/user?username=u&ticket=t → update the user with the specified user name.
- DELETE /UserService/user?username=u&ticket=t → delete the user with the specified user name.

## 5.4 DHCPService

- service responsible for the management of the DHCP server.
- for security reasons, it will forward the requests to the DHCP daemon via an inter process communication mechanism.
- GET /DHCPService/config\_file?ticket=t → get the contents of the configuration file.

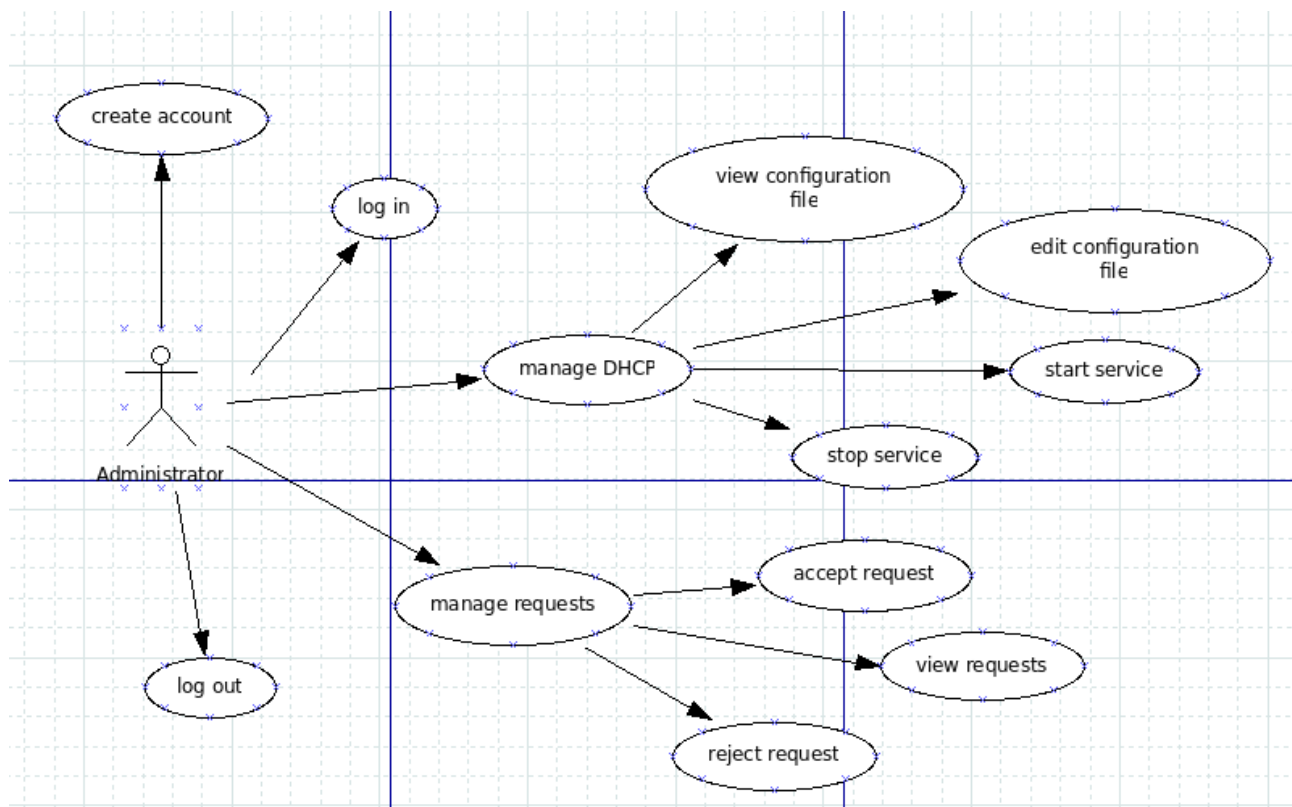
- POST /DHCPService/config\_file?ticket=t → append text to configuration file.
- PUT /DHCPService/config\_file?ticket=t → override contents of configuration file.
- GET /DHCPManager/commands?ticket=t → get the list of commands issued to the DHCP server (start, stop).
- POST /DHCPManager/commands?ticket=t&command=c → add a command to be executed by the DHCP server. The command will be executed, after the command in progress has been executed.

## 5.5 DHCP daemon

- the daemon will receive the commands from the DHCPService and execute them.
- to validate each command, the daemon will use a shared secret between him and the TicketService.

## 6. System diagrams

### 6.3 Use case diagrams



*Illustration 1: Administrator use cases*

The system administrator can perform the following operations:

- create account
- login / logout

- manage the DHCP server (view/edit configuration file, start/stop the server).
- Manage requests (view requests, accept/reject requests).

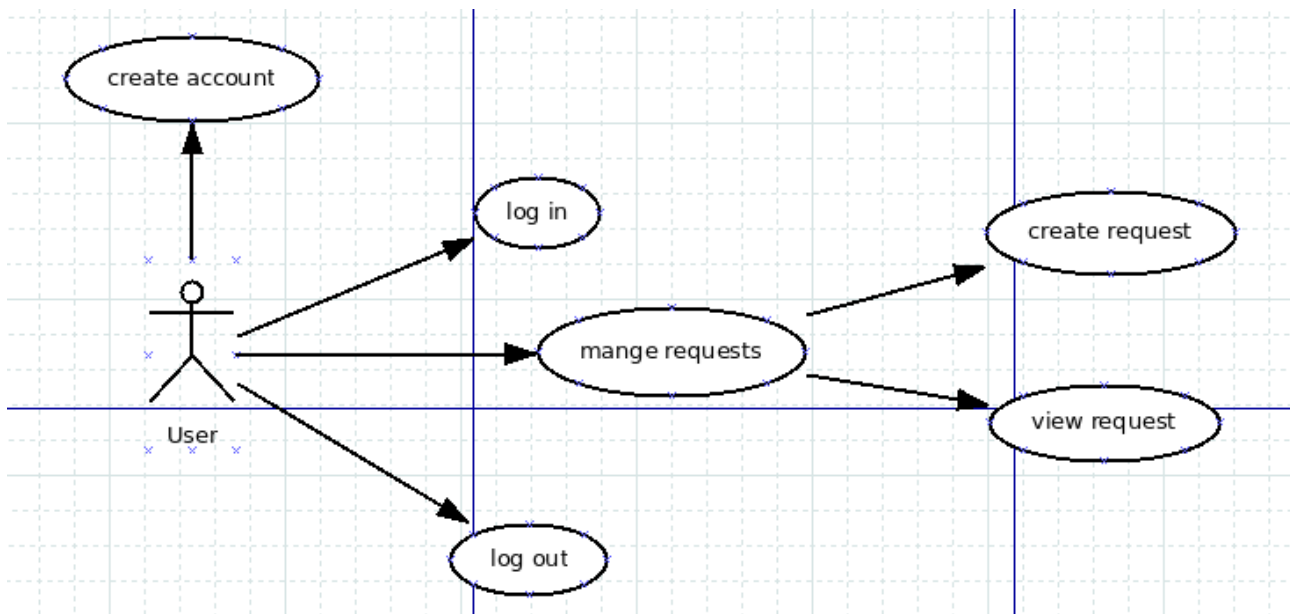


Illustration 2: Regular user use cases

The regular user can do the following actions:

- create account
- login / logout
- create / view requests

## 6.2 Sequence diagrams

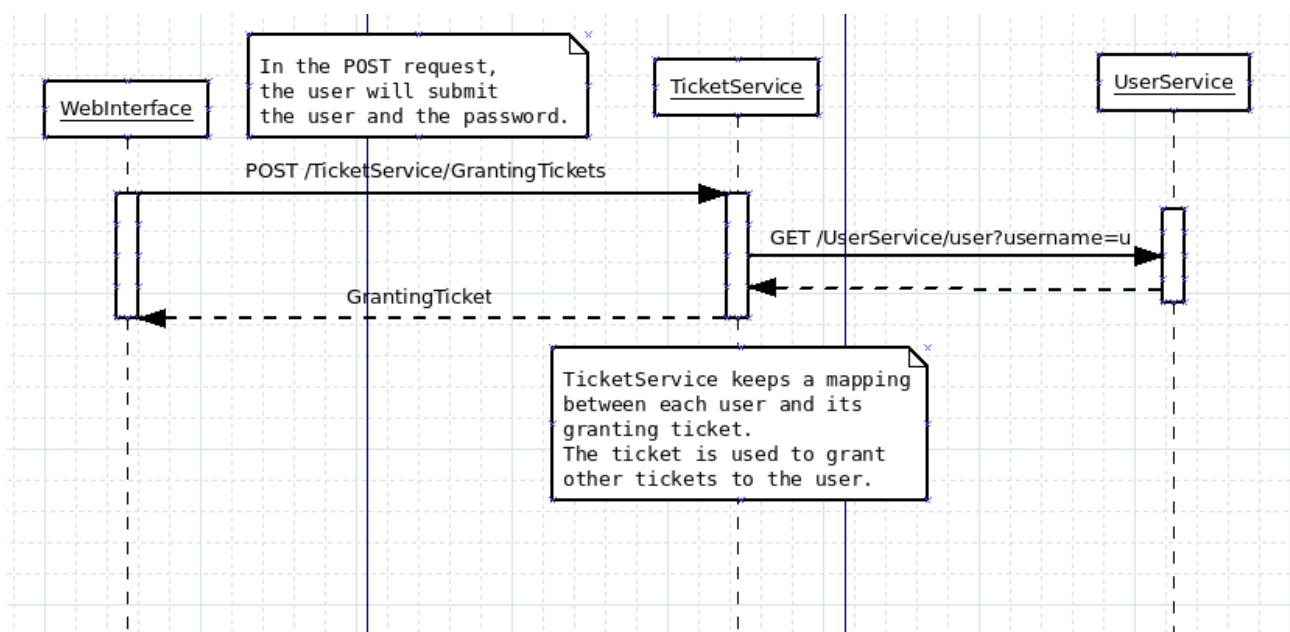


Illustration 3: Administrator login sequence diagram

When the user logs in, he requests a ticket from the TicketService. To verify the credentials of the user, the TicketService gets the correct user data from the UserService.

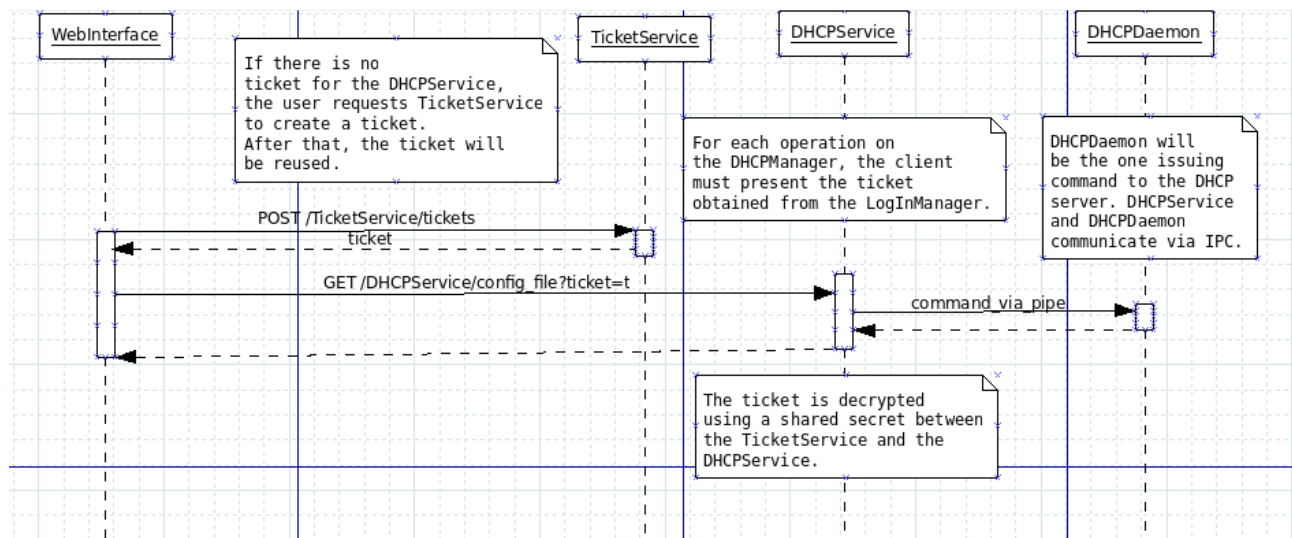


Illustration 4: Administrator view configuration file sequence diagram

When the administrator want to view the configuration file, he first requests a ticket from the TicketService. After that, he requests the contents of the configuration file to the DHCPService, presenting the ticket from the TicketService. The DHCPService forwards the command to the DHCPDaemon. After the daemon completes the command, the result is sent back to the DHCPService, and from there to the client.

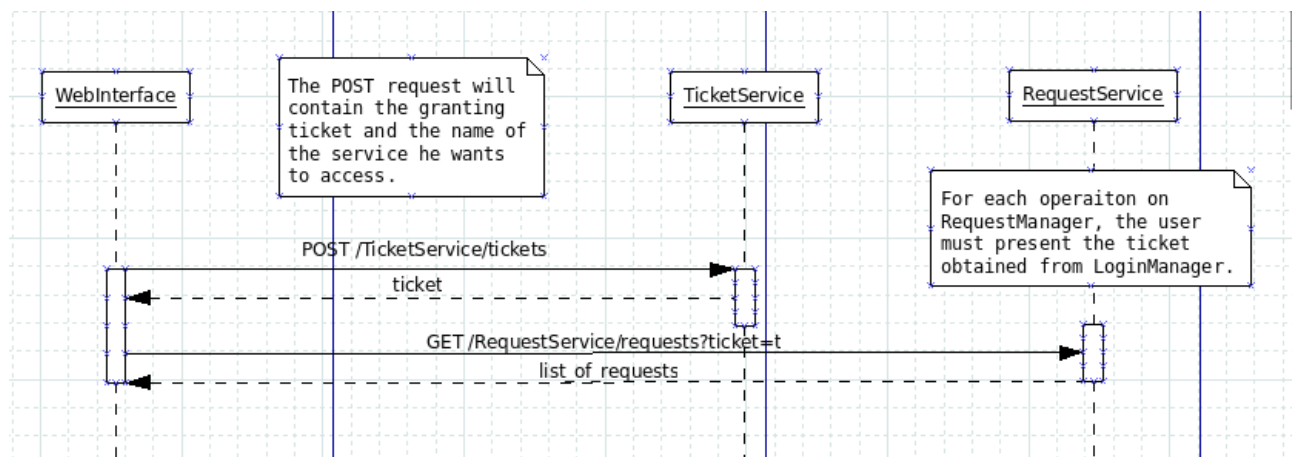


Illustration 5: Administrator view requests sequence diagrams

The same logic is applied when the administrator wants to access the list of requests. He first requests a ticket from the TicketService. After that, he requests the requests from the RequestService, presenting the ticket obtained previously.

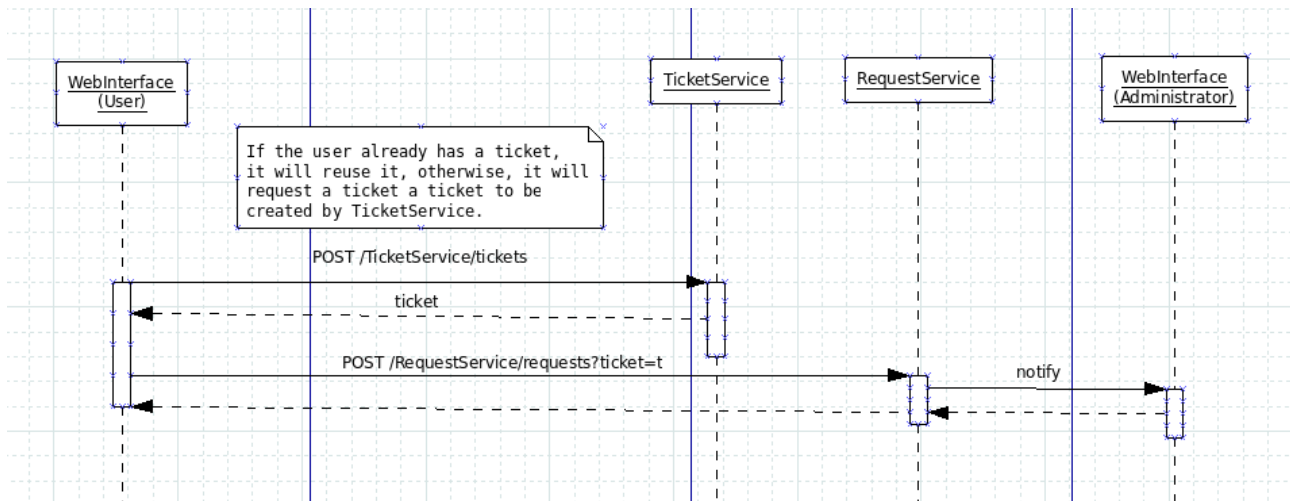


Illustration 6: User create request sequence diagram

When the user logs in, he requests a granting ticket from the TicketService. After that, he presents the ticket to the TicketService to gain access to the RequestService. When access is granted, he adds the request to the RequestService. The administrator gets notified of the new request.

## 7. Needed technologies:

- the project will be hosted on github: [https://github.com/andreichelariu92/proiect\\_sop](https://github.com/andreichelariu92/proiect_sop)

- needed technologies:

- C/C++ compiler (gcc > 5.0)
- lua 5.3
- MySQL
- sqlite
- libgcrypt
- virtual box
- Java
- Jersey