# Chapter 2.

# analyzis and REQUIREMENTS SPECIFICATION.

## 2.1. Overview of system structure.

Full software system name: “Center of Ukrainian Aviation Training”.

Purpose of system: evaluation of pilots, maintaining pilot skills and provide information about flight simulation results.

Field of usage: flight simulators training centers with orientation on training semi-professional and professional pilots.

The objectives of application: make intuitive information resource that would help pilots perform their training and get evaluation, for instructors – to evaluate pilots.

The main idea of development this system is to create comfort, effective and secure system with intuitive interface and powerful back-end. Business logic is commonly easy. The functionality is to manage users and provide correct managing of data. One of structural ideas is to cut database rights of user portal for security reasons. That is why the connection user portal will be read-only. For database itself will be created two types for connection: for main application and user portal. The simulator will be working only with main application.

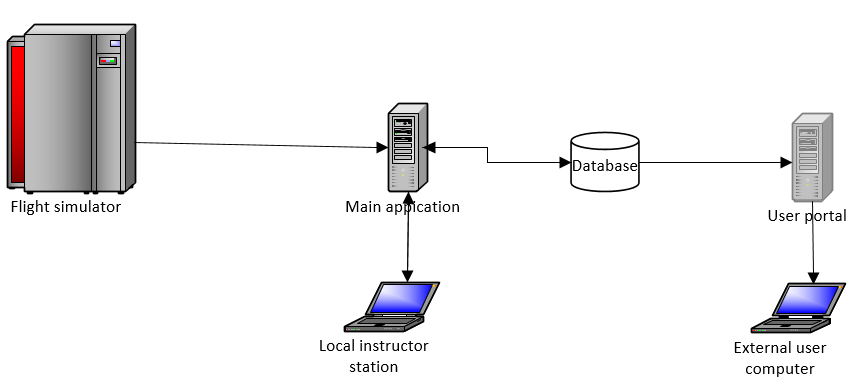


Fig.2.1. Structure architecture diagram

The whole system will be structured of four sub-systems that are described in Table 2.1.

Table 2.1.

Sub-systems description and purposes.

|  |  |
| --- | --- |
| Sub-system | Purpose |
| Flight simulator | Flight simulation is used to perform gathering data of pilot training process. |
| Main web-application | Web application stored locally and perform gathering data, calculation and user interface for comfort usage of application in web browser. Perform writing data and instructor evaluation to database. |
| User portal | Web site or portal for users to get to know the information about information center and evaluation of pilots. |
| Database | Storing the data for both main application and user portal. |

As seen from table all subsystems will be interconnected and working with database but in different ways. The structure of system is displayed on Structure architecture diagram (Fig.2.2.).

## 2.1.2. Functional requirements.

Main application functional requirements:

1. Ability to receive data from flight simulator.
2. Ability to process data from flight simulator.
3. Ability to expand data with additional information (username, plain model, etc.).
4. Ability to access database with read-write rights.
5. Ability to manage user accounts.
6. Ability to manage plain models.
7. Ability to manage metrics.
8. Ability to manage flight simulation records.
9. Ability to evaluate flight simulation records.
10. Ability to connect to flight simulator with socket.
11. Ability of authentication with session.
12. Ability of authorization with session.
13. Ability to authorize with administrator rights.
14. Ability for administrator to set roles.
15. Ability to manage data in database.
16. Ability of session management.
17. Ability to receive data from database.
18. Ability to visualize data received from database.
19. Ability to connect to application from only from local computers.
20. Ability to deploy application on different operational systems.
21. Ability to use application functionality with browser.

User portal functional requirements:

1. Ability to access database with read-only rights.
2. Ability to receive data from database.
3. Ability to visualize data received from database.
4. Ability of authentication with session.
5. Ability of authorization with session.
6. Ability to authorize with administrator rights.
7. Ability to deploy application on different operational systems.
8. Ability to use portal/site with browser.
9. Ability to perform visualization of possible data of flight simulator center.
10. Ability to perform easy visualization of record data gathered from database.

# 2.2. System architecture overview

Both main application and user portal use client-server architecture.

Client-server architecture is one of the architectural templates software and is the dominant concept in the creation of distributed network applications and provides for cooperation and the exchange of data between them. It provides the following key components:

• a set of servers that provide information services or other programs that appeal to them;

• set of clients using services provided by servers;

• network that provides interaction between clients and servers.

Servers are independent of each other. Customers also operate in parallel and independently of each other. No strict binding clients to servers. More than a typical situation is when one server simultaneously handles requests from different clients; on the other hand, the client can then apply to a single server, then to another. Customers should know about available servers, but may not have any idea about the existence of other customers.

2.2.2.