Архитектурен проект

Тема:  
Система за онлайн обучение

Група: Проект №4

Автори:  
Димитър Тодоров Тодоров Факултетен № 471219078

Иван Людмилов Виячки Факултетен № 471219005

Мартин Александров Христов Факултетен № 471219056

Николай Валентинов Каймакански Факултетен № 471219072

Павел Даниелов Иванчев Факултетен № 471219001

Съдържание

[Въведение 3](#_Toc85909719)

[Предназначение 3](#_Toc85909720)

[Обхват 3](#_Toc85909721)

[Актьори 3](#_Toc85909722)

[Архитектурен обзор 3](#_Toc85909723)

[1. Use-case изглед 3](#_Toc85909724)

[2. Логически изглед 6](#_Toc85909725)

[Class диаграма 6](#_Toc85909726)

[Диаграма на състоянието (State) 6](#_Toc85909727)

[Изглед на процесите 7](#_Toc85909728)

[Диаграма на последователностите (Sequence) 7](#_Toc85909729)

[Изглед на данните 9](#_Toc85909730)

[Изглед на внедряването 10](#_Toc85909731)

[Нефункционални изисквания 10](#_Toc85909732)

# Въведение

В последните години, станахме свидетели на една глобална епидимична обстановка, Породена от Covid вируса. Поради това бяха взети много мерки, каквито до сега не са били такава практика. Едното от тях и най-важното е обучението. Нашата програма има за цел да подобри и направи обучението по лесно и в същото време интересно за студентите. Приложението Система за онлайн обучение е реализирано с програмна реализация на C#.net. Чрез приложението ще бъдат решени важни проблеми свързани с разстояния между участниците и затруднената епидемична обстановка. Студентите, ще могат да надграждат своите знания бързо и лесно. Също така приложението ще улесни учебният процес и комуникацията между преподаватели и студенти в електронна среда. Софтуера ще предоставя бърз интерфейс, чрез който студенти и преподаватели ще обменят знания, ще бъдат провеждани тестови задачи, ще бъдат качвани материали за изпита, както и важни новини късаещи студентите.

# Предназначение

## Обхват

Документът обхваща фазата на събиране на изискванията и анализа на софтуерния продукт и оформяне на дизайна. Обхватът на тази документация е да обясни архитектурата на системата за онлайн обучение. Документацията описва различните аспекти на дизайна на системата за онлайн обучение, които се считат за архитектурно значими. Тези елементи и поведения са от основно значение за насочването на изграждането на системата за онлайн обучение.

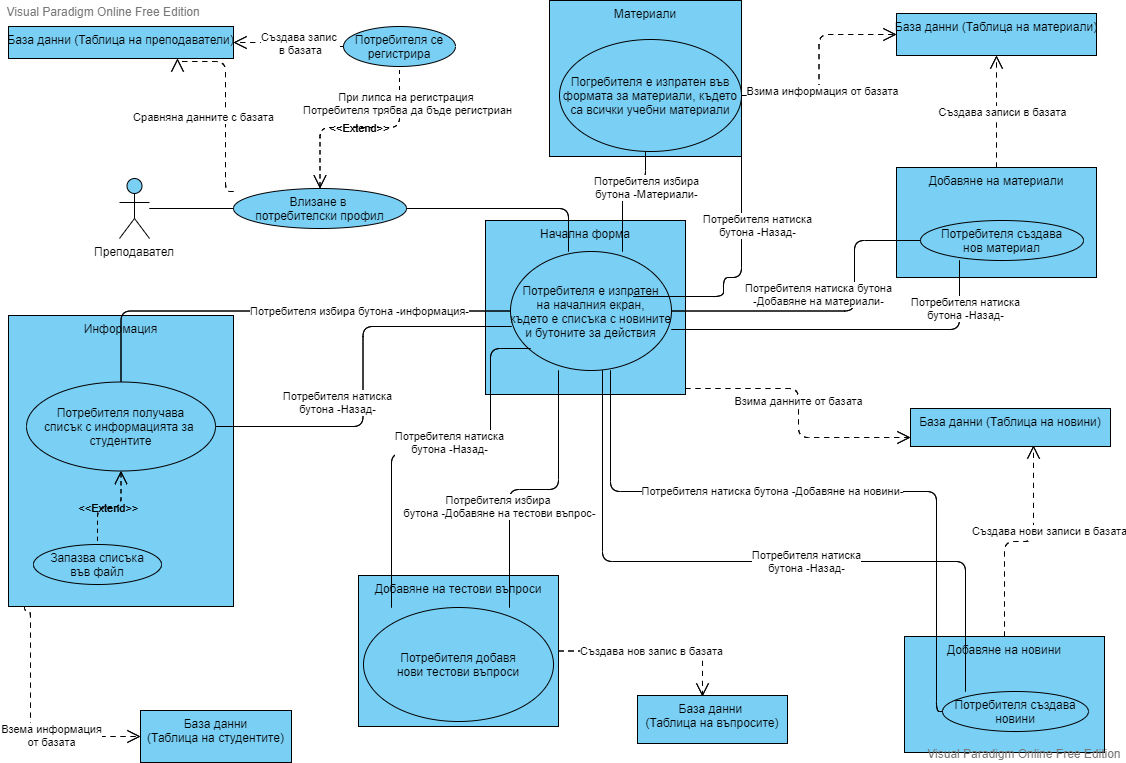
## Актьори

Преподавател – добавя новините, които са изложени на началния екран, добавя материали за студентите и въпроси за тестовете, може да преглежда материалите и да получи информация за студентите  
Студент - преглежда материалите, има достъп до информация за преподавателите, решава електронни тестове, генерирани от въпросите на преподавателите, има достъп до вече направените тестове и може да ги преглежда

# Архитектурен обзор

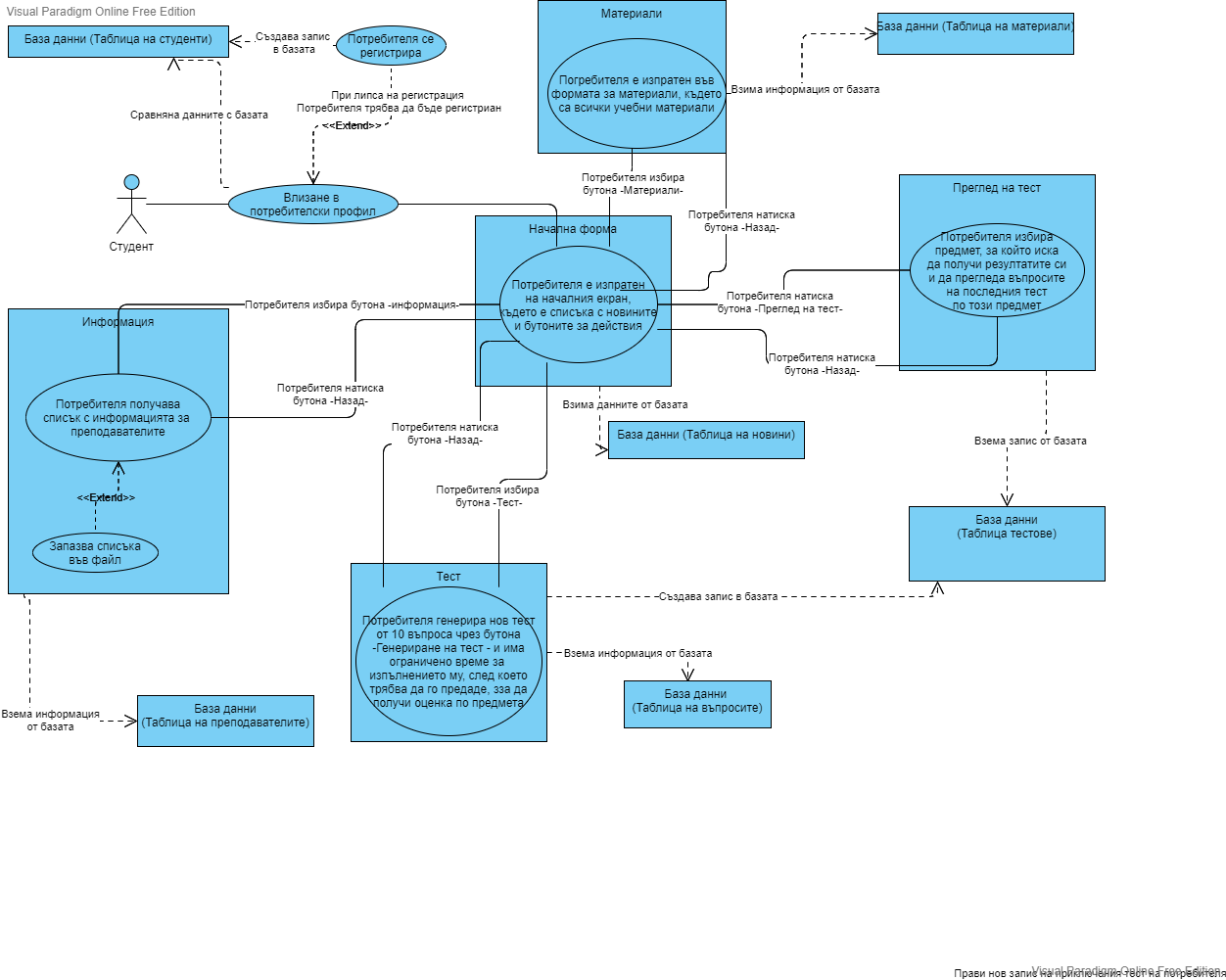
## Use-case изглед

1. Преподавател



Потребителят с роля на преподавател въвежда своята информация на първата форма, при грешно въведени данни потребителя е уведомен за несъответствията, при липса на такъв профил в базата, потребителя е уведомен, че трябва да направи нова регистрация. При регистрирането потребителят трябва да въведе всички задължителни полета и да се регистрира, при несъответствия с данните, потребителя е информиран. След като се впише в своя профил успешно, потребителя е изпратен на началната форма, къде са изложени всички новини от базата данни, както и бутоните за “Материали”, “Добавяне на материали”, “Добавяне на новини” , “Добавяне на въпрос” “Информация”. При натискане на бутона “Материали” потребителят е изпратен на нова форма, където са изложени всички материали от базата данни, при желание да се върне към началната страница може да използва бутона “Назад”. При натискане на бутона “Добавяне на материали”, потребителят е изпратен на формата за добавяне на материали. При желание за добавяне на нов материал, потребителят може да го направи след попълване на нужните полета, при несъответствие ще бъде уведомен, връщането на началната страница става с бутона “Назад”. При натискане на бутона “Добавяне на новини”, потребителят е изпратен на формата за добавяне на новини, където може да направи нов запис при правилно попълнени данни и също може да използва бутона “Назад” за връщане към началния екран. При избиране на бутона “Информация” потребителят е изпратен на формата за информация, където е представена информацията за всички студенти, преподавателят може да запази информацията под формата на файл и да се върне на началната страница чрез бутона “Назад”. При натискане на бутона “Добавяне на въпрос” преподавателят може да добави въпрос по конкретна дисциплина при правилно попълване на полетата и може да се върне в началния изглед.

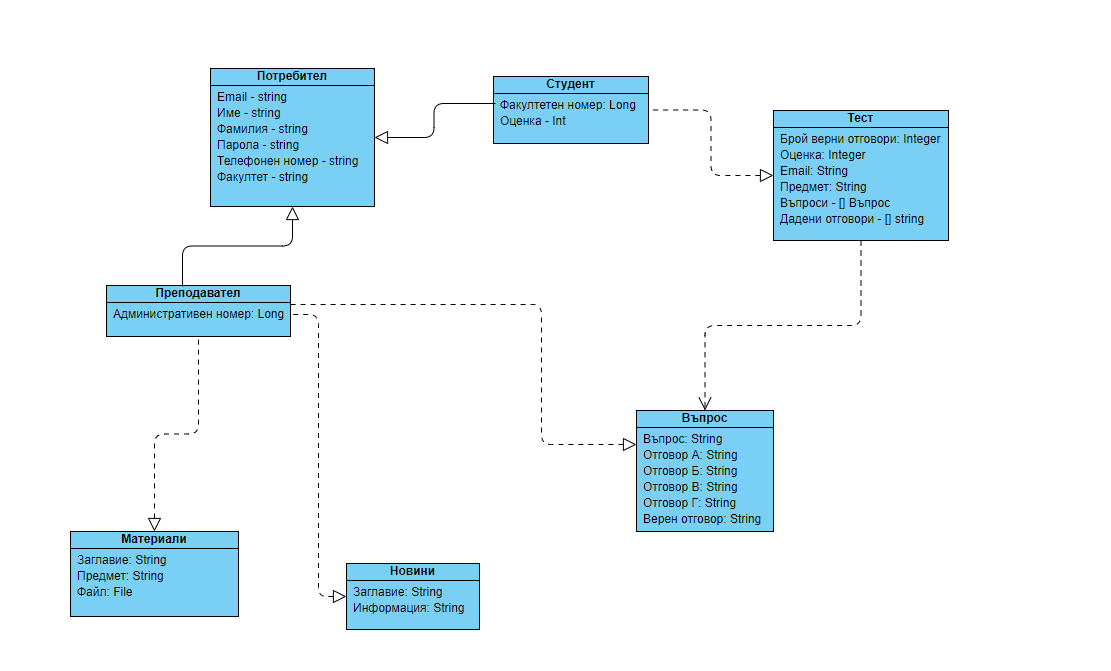
1. Студент



Потребителят с роля на студент въвежда своята информация на първата форма, при грешно въведени данни потребителя е уведомен за несъответствията, при липса на такъв профил в базата, потребителя е уведомен, че трябва да направи нова регистрация. При регистрирането потребителят трябва да въведе всички задължителни полета и да се регистрира, при несъответствия с данните, потребителя е информиран. След като се впише в своя профил успешно, потребителя е изпратен на началната форма, къде са изложени всички новини от базата данни, както и бутоните за “Материали”, “Тест”, “Преглед на тест” и “Информация”. При натискане на бутона “Материали” потребителят е изпратен на нова форма, където са изложени всички материали от базата данни, при желание да се върне към началната страница може да използва бутона “Назад”. При натискане на бутона “Тест”, потребителят е изпратен на формата за тестове. На тази страница потребителят може да генерира тест по даден предмет, след което има определено време за изпълнение и предаване, връщането на началната страница става с бутона “Назад”. При натискане на бутона “Преглед на тест”, потребителят е изпратен на формата за преглед на тестове, където може да избере предмет, по който има направен тест и да прегледа теста и също може да използва бутона “Назад” за връщане към началния екран. При избиране на бутона “Информация” потребителят е изпратен на формата за информация, където е представена информацията за всички преподаватели, студентът може да запази информацията под формата на файл и да се върне на началната страница чрез бутона “Назад”.

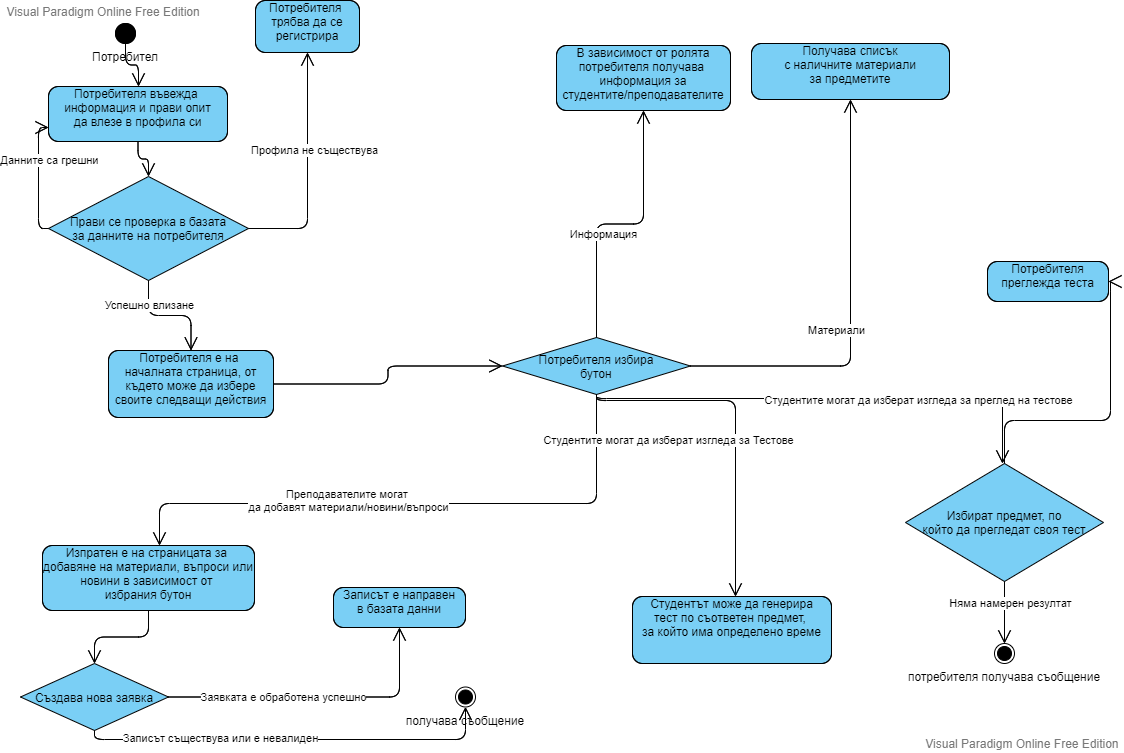
## Логически изглед

### Class диаграма



Студентът и преподавателят наследяват класа потребител.Преподавателят създава записи за Новини, Материали и Въпроси. Тестът използва въпросите като стойности, студентът генерира теста.

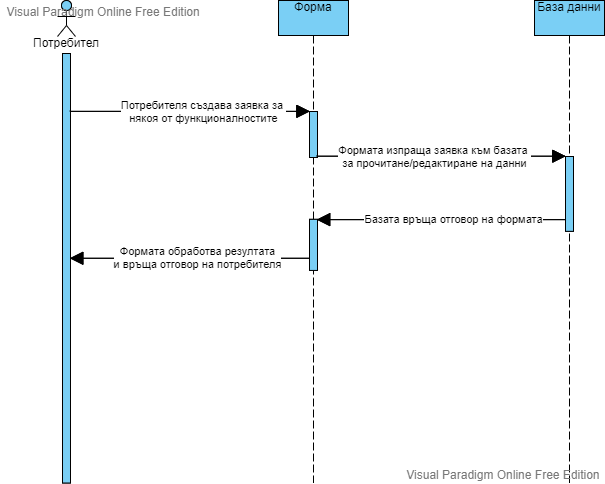
### Диаграма на състоянието (State)



Потребителят въвежда своите данни, за да влезе в профила си, при неуспех в зависимост от вида на проблема (грешни данни/ липса на такъв потребител) е уведомен за нужните от него действия. След успешно вписване потребителят е изпратен на началната страница, където има избор от различни видове действия (бутони). При избор на бутона “Информация“ е изпратен на нова форма, където са записани данни за потребителите от противоположната роля (преподавател / студент). При натискане на бутона “Материали” потребителят е изпратен на формата за материали, където са представени всички материали от базата данни. Потребителите с роля на студент могат да избират бутоните за “Тест” и “Преглед на тест”, където съответно може да направи тест за определено време по конкретен предмет във формата за тестове и да прегледа своя резултат в формата за преглед. Преподавателите могат да правят записи в базата на материали, въпроси и новини съответно чрез бутоните “Добавяне на материали”, “Добавяне на тестови въпроси”, “Добавяне на новини”.

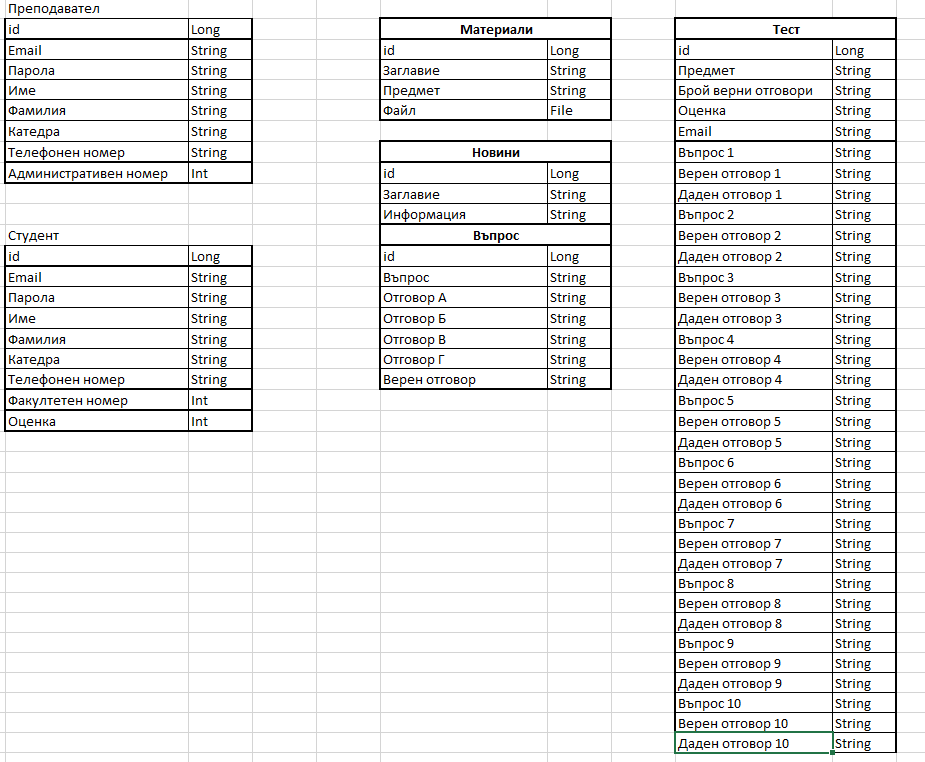
## Изглед на процесите

### Диаграма на последователностите (Sequence)

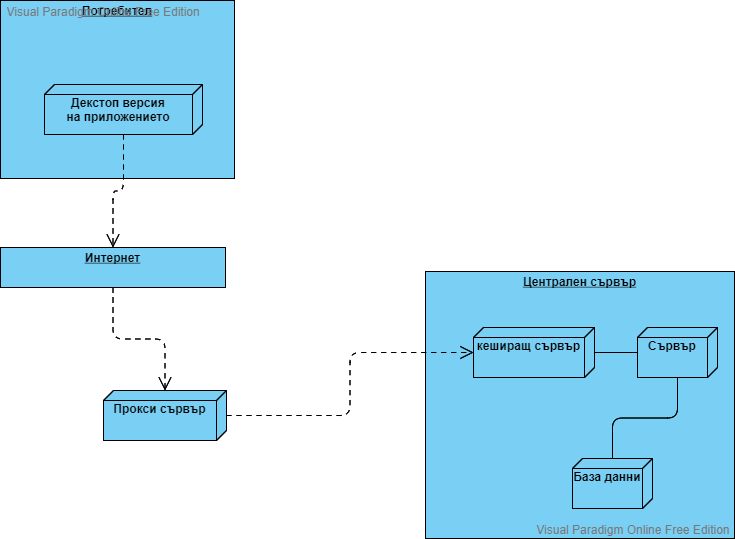


Потребителят прави заявка за функционалност към формата, тогава тя изпраща заявка към базата данни за извличане на информация или промяна на запис. След като получи отговор от базата формата обработва резултата и връща отговор на потребителя.

## Изглед на данните



## Изглед на внедряването



Клиента прави заявка за функционалност през декстоп версията на приложението, заявката е изпратена по интернет мрежата, след което е поета от прокси сървърът, който служи като ниво на защита за централния сървър. Прокси сървъра изпраща заявката към централния сървър и след това е изпратена към неговата база данни, когато бъде получен отговор от базата, Централния сървър връща резултата до декстоп приложението.

# Нефункционални изисквания

Производителност – системата извършва обработка на потребителски заявки в нормално възприетото за това време (2 секунди). Потребителите са разделени в две таблици, което ще намали броя на записите и ще ускори скоростта на търсене.

Достъпност – системата е конфигурирана локално на работната машина на всеки потребител, повреда при даден потребител няма да повлияе върху работата на цялата система, при повреда на базата данни системата ще бъде затруднена при някои функционалности, които изискват достъп до базата, но други ще продължат да работят с данните, които са били прочетени последно

Сигурност – Обменът на данни между приложението и базата е защитен, чувствителните записи от данни на потребителите са разделени в две таблици, което затруднява изтичането на конкретна информация

Възможност за разширяемост – добавянето на нови форми с нови функционалности е улеснено и няма да повлияе на досегашната функционалност ако тя не бъде засегната пряко

Надеждност – проблеми при даден потребител няма да рефлектират върху другите потребители на системата

Възможност за тестване –тестването на системата е улеснено, за всяка функционалност има отделна форма, която да бъде тестване, при проблем с дадена функционалност, тя може да бъде изключена лесно без да повлияе на останалите

Използваемост – интерфейсът на приложението е достъпен за потребителите, потребителите могат лесно да боравят с приложението, при неправилно въведени данни потребителя е уведомяван за нужните от него действия

 Хранилище за управление на версиите – GitHub <https://github.com/darcskull/STProject>