Електронна библиотекa

Разработена от:

Проектна група №4

Денис Иванов, ф.н. 471218061

Димитър Нешев, ф.н. 471218024

Каролина Кирилова, ф.н. 471218062

Райчо Симидчиев, ф.н. 471218046

# Въведение

Предназначението на проекта е да предостави възможност на потребителите да достъпват разнообразие от книги по възможно най-лесен, достъпен и удобен за тях начин. Електронната библиотека ще предлага на своите потребители избор между заглавия от множество автори, жанрове и типове. Софтуерът ще имитира работата на реална библиотека с библиотекари. Те ще са във формата на администратори, които съдействат на потребителите и одобряват техните заявки за заемане на книги.

Основните предимство на този софтуер е, че огромно количество информация ще бъде достъпно за всеки регистрирал се потребител независимо от това в коя точка на света се намира той. По този начин всеки нуждаещ се от източник на информация лесно и удобно ще може да намери това, което търси. Използвайки предоставени филтри, потребителите ще имат възможността и да сортират книгите по избрани от тях множество параметри, което ще улесни използването на софтуера.

1. Предназначение

Обхват

В документа се представят етапа на планиране на проекта и части от началния етап. В началният етап на разработката на продукта се включват крайните цели, нуждите и средствата, които ще се използват, за да се изпълни тази цел. Втората фаза е планирането, която се изявява чрез диаграмите, показващи основната структура на проекта.

Актьори

Участниците в този документ са нерегистриран потребител(гост), регистриран потребител и администратор. Те се виждат в диаграмите, които са представени по-надолу в документа. В диаграмите са представени също така и техните възможности и операциите, които извършват.

Лицата, които ще бъдат заинтересовани от този документ, ще бъдат основно проектантите и програмистите. Те ще се водят по UML диаграмите за това как да изглежда структурата на приложението и това какви са крайните цели за функционалност.

Използвани термини и символи

ERD (entity relationship diagram) - диаграма, показваща връзките между различните елементи на базата данни.

UML (Unified Modeling Language) - състои се от множество различни диаграми, които дават възможност да разгледаме една система от множество различни гледни точки.

Източници

<https://lucid.app/> - Инструмент за изготвяне на Database ERD

<https://online.visual-paradigm.com/> - Инструмент за изготвяне на UML диаграми

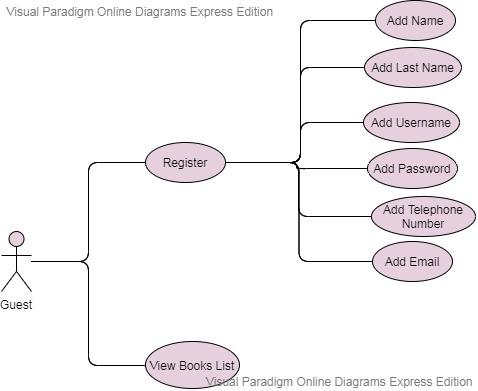
<https://fpmi.bg/moodle> - презентации от лекции и упражнения по софтуерни архитектури

1. Архитектурен обзор

Тук чрез Use-case изглед, Логически изглед и Процесен изглед ще бъде представена архитектурата на проекта и ще бъде описана информацията за всеки един от изгледите.

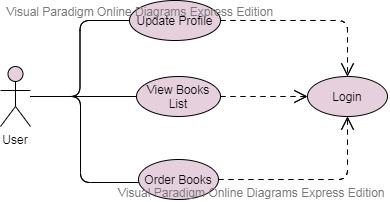
Use-case изглед –

Изглед 1 - Use case диаграма описваща възможностите на нерегистрирания потребител



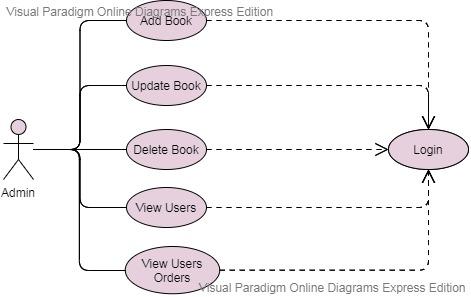
Чрез тази диаграма се показват какви възможности има един нерегистриран потребител. Той има два варианта – само да разглежда книгите или да се регистрира като си въведе данните: име, фамилия, потребителско име, парола, телефонен номер и имейл адрес.

Изглед 2 - Use case диаграма описваща възможностите на регистрирания потребител



Чрез нея се показват трите възможности на регистрирания потребител. Те са: да си обнови инфорацията в профила, да прегледа книгите и третата възможност е да си поръча книги. Като тези три възможности са зависими с входа в системата, т.е. ако потребителя не влезе в системата чрез потребителско име и парола, той ще има само една възможност – да разглежда книгите като гост.

Изглед 3 - Use case диаграма описваща възможностите на администратора

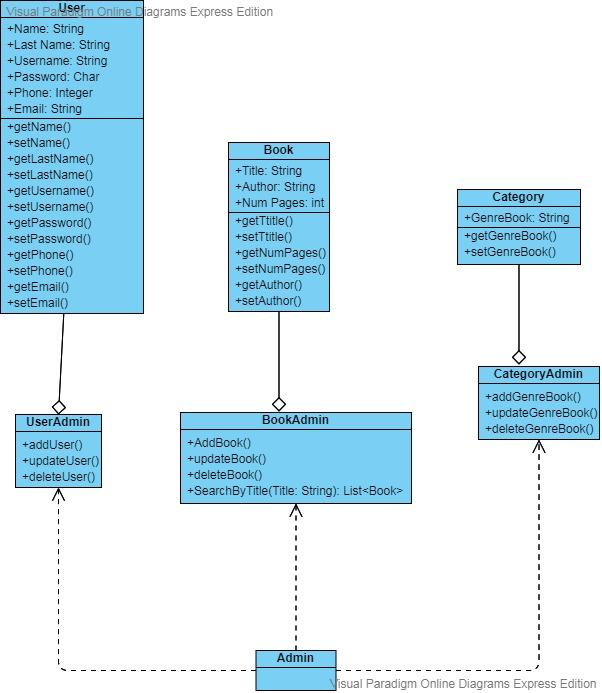


Чрез тази Use case са показани нещата, които един администратор може да прави. Той има възможност да добавя, променя или изтрива дадена книга, да преглежда регистрираните потребители и да преглежда и одобрява техните заявки за книги. Тези действия за зависими с входа в системата.

Логически изглед

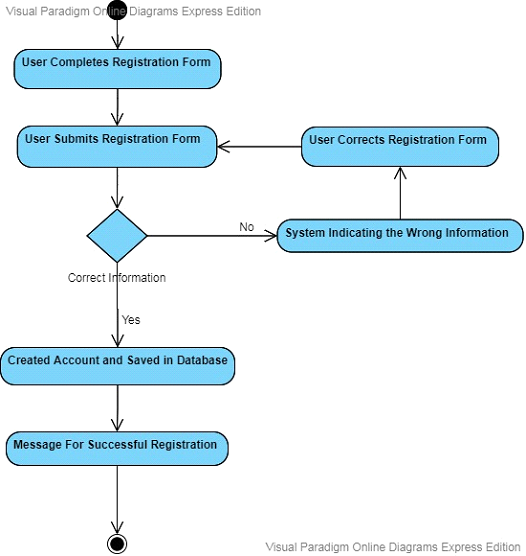
Изглед 1 – Class диаграма

Информацията в нея показва структурата, класовете и зависимостите между тях в дадената система. Чрез агрегация е показано, че обекти от класовете – потребител, книга и категория съдържат указатели към друг клас, а именно класовете – админ на потребителите, админ на книгите и админ на категориите. Тези три вида администратори за да съществуват и да извършват определени опрерации за зависими от класа администратор, като тази връзка винаги е еднопосочна.



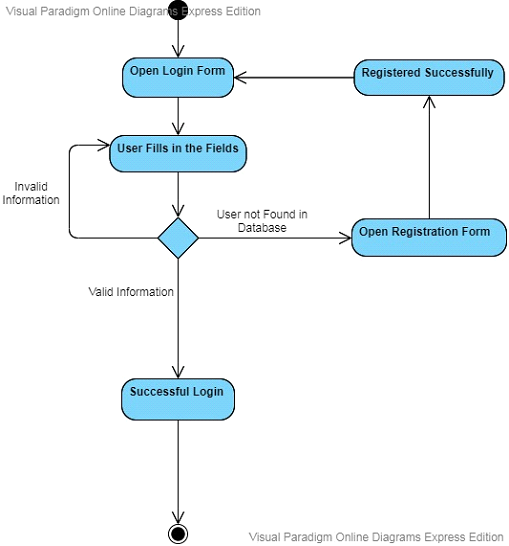
Процесен изглед

Изглед 1 - Activity диаграма за регистрация



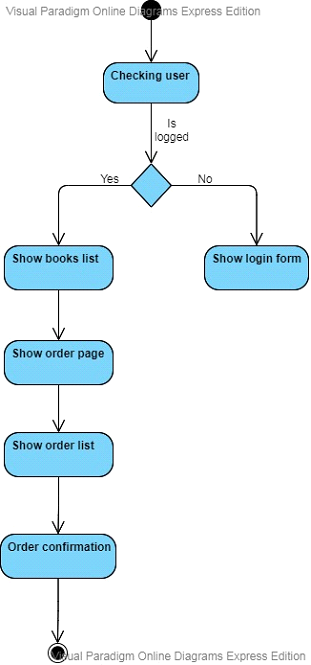
Тази Activity диаграма показва процеса по регистрацията на потребител. Първата стъпка е попълване на полетата с данните и след това изпращането им. После системата проверява дали въведената информация е правилна и ако е вярна се създава профил, данните му се запазват в базата данни и се показва съобщение за успешна регистрация, а ако има някаква грешно въведена информация или вече съществуващо такова потребителско име или потребител със същия email, системата му съобщава и го връща към формата за регистрация за корекция и изпращане на данните и ако всичко е наред се създава профил.

Изглед 2 - Activity диаграма за вход



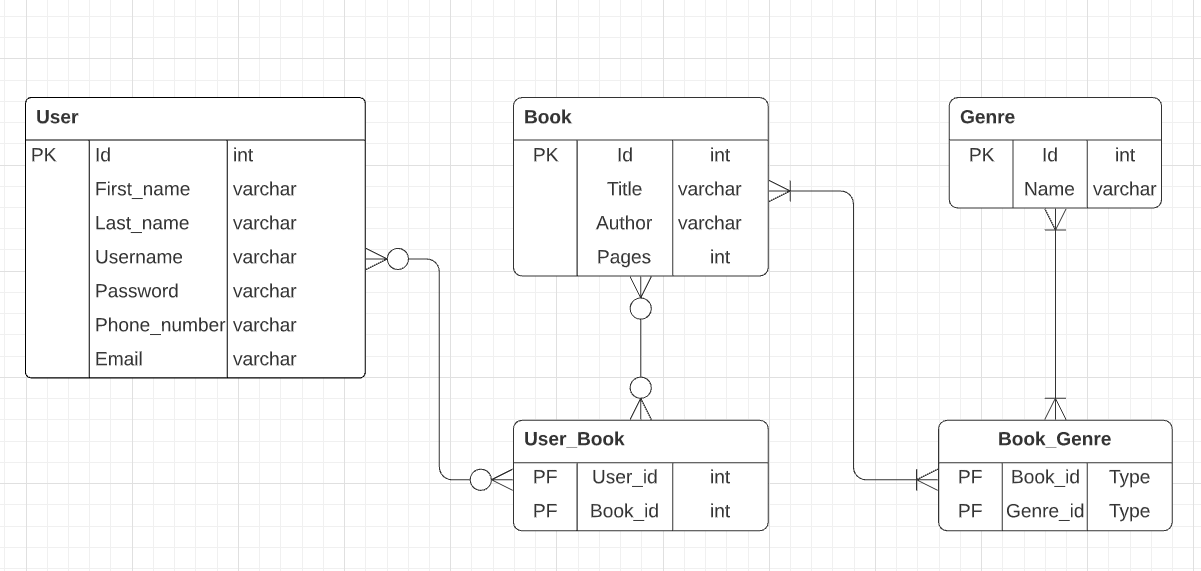
Чрез тази Activity диаграма е представен процеса по вход на регистриран потребител. Първо се отваря формата за вход и потребителят въвежда данните в полетата и системата проверява дали има такъв потребител и ако има дали въведените данни са правилни. Ако няма такъв потребител се отваря формата за регистрация и ако регистрацията е успешна, потребителя си въвежда данните за вход. При неправилно въведени данни системата го уведомява и го връща за поправяне, а ако всичко е правилно въведено входа в системата е успешен.

Изглед 3 - Activity диаграма за поръчка



Тази Activity диаграма представя процеса по правене на заявка за книги. Първо системата проверява дали потребителя е влязъл и ако не е влязъл му отваря формата за вход. Ако се е логнал му показва списъка с книгите, след това страницата за заявка и неговият списък за поръчка и накрая за потвърждение на заявката за книги.

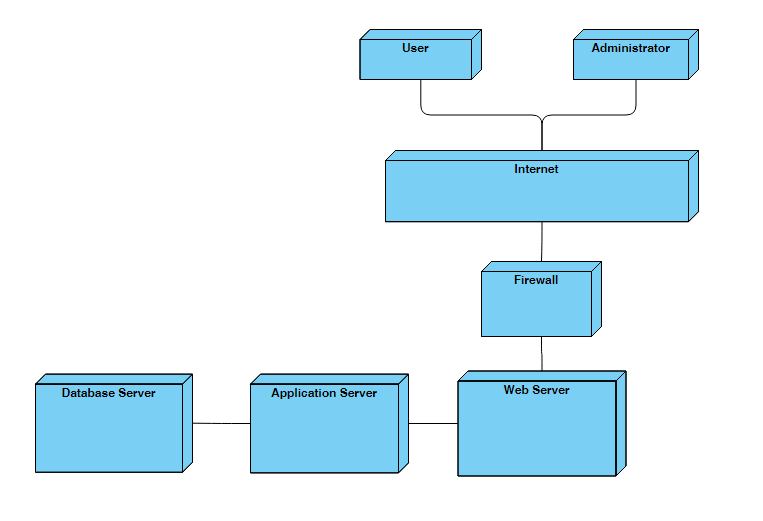
Изглед на данните



Тази графика изобразява различните типове данни, които ще бъдат съхранявани в базата, както и техните типове и релации. Връзките между User и User\_Book, Book и User\_Book са нула или много към нула или много. Това е така поради причината, че един User може да има нула или няколко книги, а един Book може да е назначен към нула или множество Users.

Връзките между Book и Book\_Genre, Genre и Book\_Genre са много към много с изискването да има поне една връзка. Това е така, защото една книга няма как да няма жанр и не се позволява да има жанр без книги. Понеже една книга може да има няколко жанра и един жанр съдържа множество книги, връзката е много към много.

Изглед на внедряването



Приложението ще бъде достъпно през интернет. Достъпът до него ще бъде ограничен от Firewall. Щом информацията достигне уеб сървъра, той я предава към application сървъра, който я обработва и я връща обратно към потребителя.

1. Нефункционални изисквания

1. Достъпност - Системата трябва да бъде сигурна с цел предотвратяване на спирания, което ще причини евентуална загуба на клиенти. Също трябва да притежава поддържащ екип, който да предотвратява сривове в системата и да се справя с евентуални проблеми, като претоварване на сървъра от много клиенти едновременно изискващи данни.

2. Разширяемост - Увеличаващият се брой клиенти може да доведе до срив в системата, затова използваната архитектура позволява системата да бъде лесно разширявана чрез добавяне на ресурси към сървърите и увеличаване броят на клиентите поотделно.

3. Производителност - Системата трябва да притежава голяма база от данни, която да съхранява големият брой книги и също се нуждае от бърз сървър за прехвърлянето на протоколи. Но за предотвратяване на забавяне при изпращането на протоколи системата се нуждае от добра мрежа.

4. Сигурност - Системата трябва да бъде защитена от злоупотреба с лични данни записани в регистрираните профили. Това се улеснява чрез централизираната му архитектура и чрез ограниченият достъп от Firewall.

5. Възможност за тестване - Системата не се нуждае от големи средства за нейното тестване. Трябва да се проведат тестове на ниво сигурност, и производителност.

6. Използваемост - Системата трябва да бъде лесно използваема поради големият възрастов диапазон на аудиторията използваща то приложения. трябва да е подходяща за малки деца чрез лесен и интерактивен достъп до дадено четиво, както и да е улеснен към аудиторията на по възрастните затрудняващи се да използват новите технологии.