



Софийски Университет „Св. Климент Охридски“
Факултет по Математика и Информатика
Катедра „Софтуерни Технологии“
София

Курсов Проект

по

Архитектури, ориентирани към услуги, и уеб услуги

*Локално базирано софтуерно приложение за размяна
(купуване) на книги*

Изготвили:

Стефан Байчев, фн 24170

Александра Саликиряки, фн 23986

Магистърска програма: „Софтуерни Технологии“

Цел на разработваното приложение

Приложението има за цел да предостави локално базирана система за размяна/ купуване на книги/ списания (при нужда с възможност за дефиниране и на друг тип информационни носители) между различни клиенти. Предлагащите от системата на този етап на услуги обхващат както следва следните достъпни в обобщен вид програмно съставени дейности:

1. Възможност за автентикация (реалното съществуване) и ауторизация на всеки потребител в системата
2. Проверка за наличните/ предлаганите за размяна/ книги/ списания
3. Добавяне на книги/ списания от съответния потребител
4. Услуга предлагаща запазване и впоследствие търсене дали е налична книга/ списание което потребителя харесва, но в даденият момент на търсене не е налична
5. Външна услуга предоставяща проверка валидността/ коректността на ISBN номера на дадената книга/ списание

Анализ на изискванията към приложението (функционални и нефункционални изисквания)

Функционалните изисквания са както следва:

За всяка една от горе-посочените услуги потребителя е нужно да:

1. Да бъде проследявано дали потребителят има автентикация и/или ауторизация (както и да му се предостави регистрация, за да осъществи дадените) в системата като такъв.
2. Потребителя да има възможност да търси в системата(приложението) при въведени такива специфични за дадена книга/ списание критерии за търсене и да получава резултат отговарящ на зададените параметри.
3. Потребителя да има възможност да добави книга/ списание което иска да предостави за заемане, като има на разположение програмна функционалност даваща му възможност да дефинира параметрите ѝ/ му.
4. Потребителя да получи програмно дефиниран достъп до информация предоставяща му възможност да запази харесвана, но не налична в момента, книга/ списание, като в последствие със съответната програмно дефинирана функционалност, да провери дали е налична дадената/ то
5. Потребителя получава съобщение за грешка при невалидно/ некоректно въведени данни за ISBN номера на дадена книга/ списание, което бива въвеждано от системата на екрана на потребителя

Нефункционалните изисквания са както следва:

Съобразявайки се със стандартите за такъв род приложение изискващо надеждност, поддръжка и бъдещо развитие, сме се придържали при разработването на програмните дейности да използваме относителна програмна простота на писаният код, следвайки характерните метрики и принципи на ООП.

Обосновка на използваните програмни технологии

Използваните технологии за направата на софтуерното приложение са както следва:

Относно Услугите

- ✓ Програмните езици C# (.Net програмната среда) и Java (JRE)

Относно Приложението използващо Услугите

- ✓ Базирано е на .Net програмната среда, чрез програмният език C#, а по специфично използвайки .Net архитектурно базирани технологии

Осъществените в горе-посочените програмни среди са такива, следвайки изискването в документа с име „ SOA-Projects “ и са съобразени съответно с познанията и опита на екипа за тях. Съответно, предлагат при нужда от софтуерно надграждане сравнително бърз, лесен и адаптивен начин за постигането му.

Проектиране на услугите и приложението, което ги използва (архитектура, модули, услуги, протоколи за комуникация)

Използваните протоколи за комуникация са SOAP и REST посредством HTTP информационно-преносният протокол.

Приложението използващо услугите е реализирано посредством Windows Forms, базирано на .net Framework архитектурата.

Описание на програмната реализация (модули, услуги)

Следва да бъде дадено кратко описание на програмно реализираните модули и услуги:

Всяка една от съставените услуги има своя характерна точка(и) за достъп съобразена с възможната нужда на дадено използващо го клиентско приложение. Целта е да се осигури централизиран достъп до един общ ресурс за услугата, като по този начин се капсулира средата за комуникация с нейните вътрешни методи и състояния, предпазвайки я от нежелани/нехарактерни програмни влияния.

Следва на страница 3, Таблица със съответните характеристики, изисквани според горе-посоченият документ „ SOA-Projects “ за задължителните програмно-функционални Условия и следвайки ги, са посочени описанията на различните Услуги-Методи осъществени за всяка една от вече споменатите по-горе общодостъпни и смислово обособени Услуги:

Отговаря на точка от Критерии за оценка на проекта: ▾	Услуга ▾	Метод ▾	Език ▾	Комуникационен протокол ▾
1. Разработване на услуги върху една платформа с един комуникационен протокол	Authentication	public OperationResultValue<string> AddUser(string Country, string City, string PostCode, string Address, string UserName, string Password, string Email, string FirstName, string LastName, string Phone) public string IsValid(string userName, string password)	C#	SOAP
2. Разработване на услуги върху различни платформи	BookOfInterestService	public OperationResult AddBookOfInterest(BookOfInterest book, string userName, string guid) public OperationResultSet<Book> CheckBookOfInterest(int bookid, string userName, string guid) public OperationResultSet<BookOfInterest> GetAllBooksOfInterest(string userName, string guid) public OperationResultSet<BookOfInterest> GetAllBooksOfInterestByUser(string userName, string guid)	Java	REST
4. Използване на външна услуга (услуга, която не е разработена от екипа)	ISBNService	public bool IsValidISBN13(string sISBN) { object[] results = this.Invoke("IsValidISBN13", new object[] { sISBN }); return ((bool)(results[0])); }		SOAP
5. Разработване на услуга, достъпна при използване на различни комуникационни протоколи	BookManagementService	private bool CheckAuthenticated(string userName, string guid) public OperationResult AddBook(Book book, string userName, string guid) public OperationResult AddAuthor(Author author, string userName, string guid) public OperationResult EditBook(Book book, string userName, string guid) public OperationResult ReserveBook(Book book, string userName, string guid) public OperationResultSet<Author> GetAllAuthors(string userName, string guid) public OperationResultSet<string> GetAllTypes(string userName, string guid) public OperationResultSet<string> GetAllGenres(string userName, string guid) public OperationResultSet<Book> GetAllBooks(string userName, string guid) public OperationResultSet<Book> GetAllBooksByYear(int year, string userName, string guid) public OperationResultSet<Book> GetAllBooksByTitle(string title, string userName, string guid) public OperationResultSet<Book> GetAllBooksByAuthor(string authorName, string userName, string guid) public OperationResultSet<Book> GetAllBooksByUser(string userName, string guid) public OperationResultSet<Book> GetAllBooksByGenre(string genre, string userName, string guid) public OperationResultSet<Book> GetAllBooksByType(string type, string userName, string guid)	C#	SOAP + Rest
6.1 Интегриране на услугите в уеб сайт или десктоп приложение	LibraryExchange desktop application	LibraryExchange Main Form AddAuthor Form AddBook Form Authentication Form Register Form EditBookForm Form Edit Books Form Reserve Book Form Search Form	C# Windows Forms	

- Специфична особеност е OperationResult и OperationResultSet типовете, показани в колоната „ Метод “, за дадените такива, които се грижат след изпълнение на вътре-създадените в методите дейности, да бъдат прихванати и отразени набор от нежелани/ желани резултати от исканите и те да бъдат комуникирани по съответния разбираем начин до потребителя.

Дата: 02.06.2013 г.