Software Server for Automatic Generation of Audio Lectures (uListenSrv)

https://doi.org/10.3991/ijac.v10i1.6475

Milen Petrov
Sofia University, Department of Software Engineering, Sofia, Bulgaria
milenp@fmi.uni-sofia.bg

Asen Asenov
Sofia University, Department of Software Engineering, Sofia, Bulgaria
aasenov.bg@gmail.com

Adelina Aleksieva-Petrova
Sofia University, Department of Software Engineering, Sofia, Bulgaria
Technical University, Sofia, Bulgaria
aaleksieva@tu-sofia.bg

Abstract—Facilitation of new methods for learning materials delivery and adoption of new learning experiences and practices in e-learning is always a challenge. Using synthesis of digital audio learning assets and learning objects as one of main sources for conducting learning is not new, but research on using audio lectures or combined audio with presentation lecture is not well investigated and adopted in traditional online learning environments. The main goal of current paper is to present requirements elicitation, software analysis, design, construction and testing of secure and reusable software architecture for production and delivery of learning resources with audio elements in university programming courses. Paper presents different architecture styles for designing the system and finish with presentation of development and usage of contemporary Software Server for Automatic Generation of Audio Lectures (uListenSrv). Main difference here is support of languages, not only in English, but not so popular languages, like Bulgarian language.

Index Terms—Software Architecture, Speech Synthesis, Learning Management Systems, Security Management

1 Introduction

1.1 Overview

Current work represents analysis, design, implementation and testing of software server called uListenSrv - that is web–based, platform independent system that gives opportunity to transform wide variety of presentation and text files into voice readings, available for download. As input system takes existing provided presentation documents (lectures) or other learning resources, then transfigured into online based presentations with embedded sound. The system cope with information security management and works with user accounts, where registration and its use is entirely free of charge. The developed architecture satisfies following software requirements: multi-tenant, configurable, client-server, service-oriented, modular, and extensible in nature. Communication between client and server components in the system are protected with secure channel, using standard HTTPS protocol. End users communicate with the server and explore system functionality using *web client*. The server layer provides access to core functionality via software services through network, which are documented to facilitate usage from external software systems, and learning management systems (LMS).

Table 1. Existing systems for text to speech generation [15, p.14]

Criteria for evaluation	**Appearance**			
	Google Translate	***YakiToMe!***	***iSpeech!***	***SpokenText***
Remote access	8	10	10	0
Offline use	0	0	8	0
Open source	3	2	2	0
Free use	10	3	3	5
Extensibility	0	0	0	0
Search in files	0	0	0	0
User friendly interface	10	9	6	10
Voice quality	10	10	10	10

Main functionality, provided by the proposed software server system, can be summarized as follows:

- Reading of HTML, TXT, PDF, DOC and many other file types, generating and playing audio recordings (WAV).
- Parse PPT and PPTX presentation formats and visualize them in web based presenter, with embedded slides reading.
- Adding file ratings.
- Enable searching in uploaded files name and content.
- Maintain user accounts, allowing registration, login, logout, file sharing, and custom settings.

System is divided into subcomponents and each subcomponent is designed as subproject in order to foster reuse of software and architecture decisions.

1.2 State of the art

In the literature audio lectures widely are used in general in language learning, for example [1]. In other research [2] audio readings were introduced into two courses (ten offerings) using Apple Computer's iPod audio player for delivery and total of 20 audio files were created to use in the courses which results in suggested three significant factors: preference, usability, and experience of end-of-course surveys indicated that there is significantly higher satisfaction scores for those courses with audio readings compared to those courses without audio readings.

Interesting study [3] which contributes to personalization in multimedia learning by evaluating if there are differential effects between using audio that is instructor-narrated (i.e. all the students personally know the narrator) versus expert-narrated (i.e. none of the students know the narrator). In this study there are created two identical instructional audio files about a difficult conceptual topic. The only difference between these two treatments is that one audio was done by an "expert" that the students did not know and the other was done by their instructor whom they saw in class at least once a week. In both cases the content of the audio was exactly the same. Students in a large undergraduate nutrition course listened to two audio case study analyses as part of an online module about vitamins and minerals and their satisfaction is studied.

As next step to previous study we propose to measure satisfaction of students with adding generated audio lectures with the lecture which they know very well, and using lecturers words, but audio is produced by usage of audio-generation software, known as text-to-speech software. Finding a good free software for non-English language is a challenge, especially for not popular languages, such as Bulgarian language.

Good support for audio and other multimedia resources in W3C recent recommendation HTML 5 [4] in browsers, as well as other devices such as e-book readers like Kindle Fire, smart watches [5] becomes more and more popular. Common usage of audio-visual lecture today is by facilitating YouTube platform [6] or similar general purpose media services with live-recorded lectures or screencasts.

Using **recorded audio materials or audio materials with images is not** that common in university courses. For example in Software Programming courses and courses in Software Engineering stream or learning in corporate environment, workplace or by lifelong learners. In our opinion in general: using audio resources as learning experience is under used in general education systems, if used at all. There are several factors for such situation:

- It is **not easy to create only audio lecture** as that cannot be fully replaced by presentation materials and live lectures, given by human instructor.
- **Audio resources in general are not searchable**, if transcript is not created (which is not easy/cheap to achieve). It is even harder to automatically generate transcripts in non-English languages.
- It is huge and **time-consuming task to create recorded audio lectures**. Moreover if there is need to search, update, edit or delete such a resources-can be a challenge.

We come to the idea to design and implement secure software architecture, which addresses these challenges:

1. Fast and easy creation of audio resources from text slides, using open source components.
2. Make resources searchable.

3. Being time consuming task - once done - it is important to make easily re-used audio lectures in Personal Learning Environment (PLE) [7, 8] and as general learning resources in Learning Management Systems (LMS) [9] such as Moodle.

Making resources easy reusable is very important goal as tools, extracting course content can **perform repackaging of the content for reuse** in different learning environments [10].

In table 1 are compared several existing tools for text-to-speech transformation such as “Google Translate” [11], “YakiToMe!” [12], “iSpeech!” [13], and “SpokenText” [14]. Selected criteria for evaluation are: remote access, offline use, open source, free license, extensibility, search (in files), user friendly interface, and voice quality. As result we concluded, that there is need for new server software system, which satisfied initial user requirements, which are described in details in section III below.

1.3 Structure of paper

In next section of paper, named "*Analysis of Web-based Software Architectures*" there is presented one attempt for **categorization of different type of web-based Software Architectures** which are candidates for design of the developed software server solution with key advantages and disadvantages. In third section, named "*Architecture for Audio Presentation System*" all modules and structure of the designed architecture are discussed. In forth section "*Implementation of Audio Lecture Generation Software*" are detailed discussion on implementation and common usages of developed server. In fifth section named “*Usage of the system*” practical usage scenario is used when navigated trough steps to use the system. In last section conclusion about usages of the system are analyzed, results and further improvements are discussed.

2 Analysis of Web-based Software Architectures

2.1 Software architecture categorization

Here we made categorization of several software architectures, which are elicited as possible software design for our server system according to number of servers and number of web clients and their disposition (local or remote), for both client and server – as in Table 2. Different architecture characteristics has their positive sides and drawbacks, discussed here.

Table 2. Web architecture classification.

	Single client	**Different clients**
Single server/single database	Traditional client-server	Traditional client-server
Multiple server (single database)	1	2
Multiple server (Multiple database)	3	4

On Fig. 1 is depicted first scenario (1) - **one local server**, which **accepts connections from multiple clients**. Multiple 'local' clients, connects to same local 'server' and share same data; can produce and share audio lectures asynchronous/sequential to potentially different local users. This **software architecture can be called kiosk system** – any user can be served, but only one user in same time.

On Fig. 2 is depicted use case scenario (2) - **remote server** is on the left on the figure **and several clients** on the right. Mixed mode means remote and local clients are allowed. We can have synchronous and asynchronous client-server communication. Usage scenario here can be called Personal Learning Environment (PLE). Allows roaming client user to different places with multi-homing hardware and/or software environments.

On Fig. 3 is depicted scenario (3) - typical server model-view controller (MVC) deployment architecture. This means, that it cannot be used offline as PLE. This can limit some of the usage scenarios.

On Fig. 4 are depicted packages and on Fig. 5 is software architecture of **multi-tenant scenario** (4) - which allows same client to connect to different servers – one at a time. Thus with one client can achieve both – having possibility for using as PLE or LMS.

Looking at different possible architectures, we decided to implement last architecture, as giving great flexibility, security and reusability.

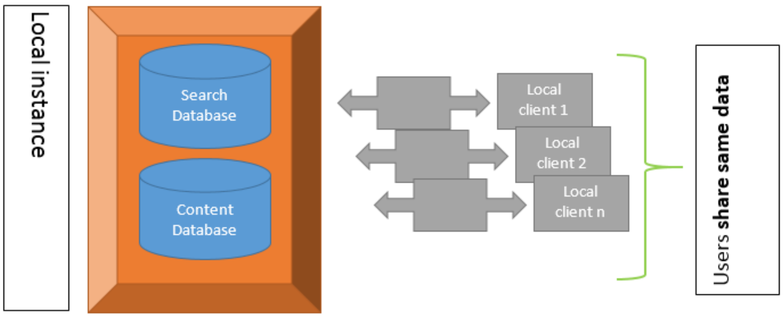


Fig. 1. Software architecture: local mode (kiosk)

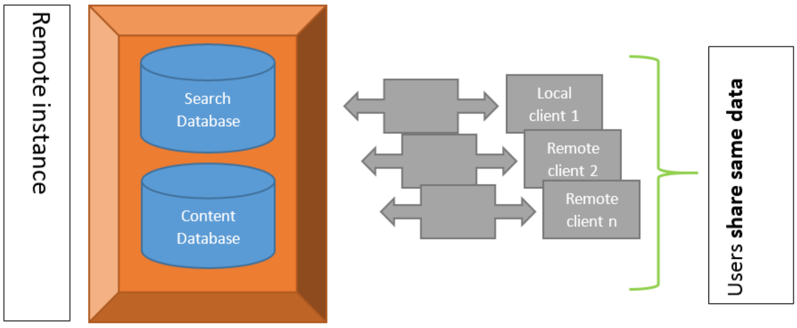


Fig. 2. Software architecture: mixed mode (PLE)

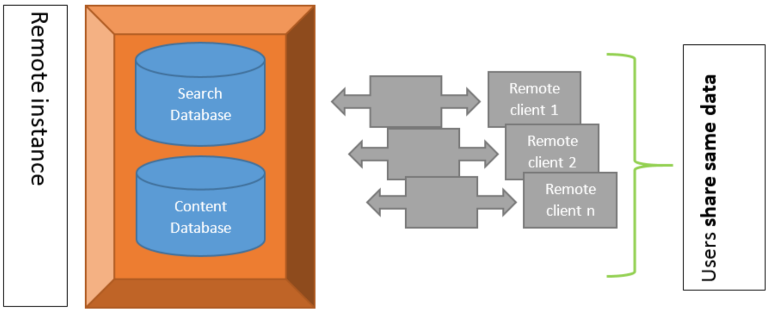


Fig. 3. Software architecture: traditional server-only deployment

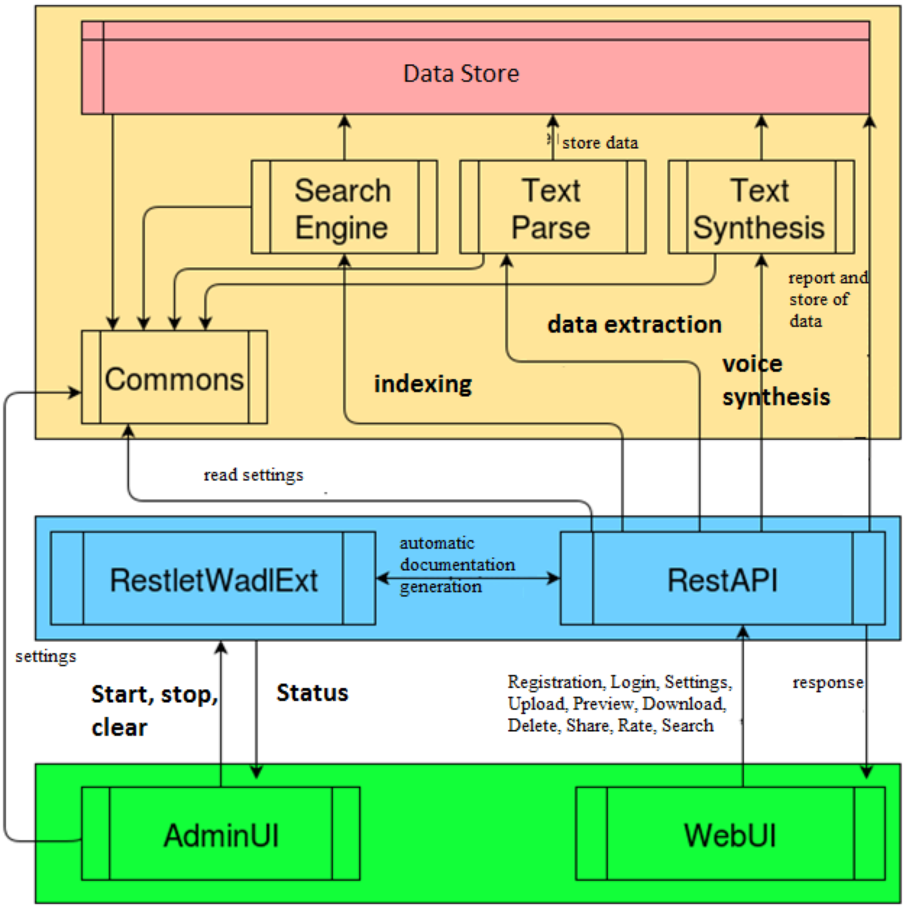


Fig. 4. Package dependencies and operations

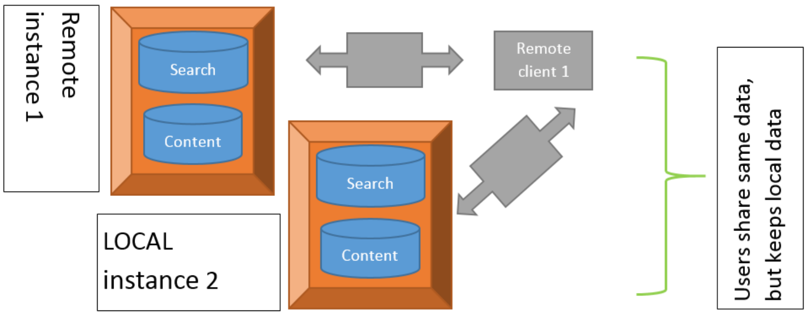


Fig. 5. Software architecture: Multi-tenant software architecture

3 Architecture for Audio Presentation System

3.1 Introduction

As initial data for the system is needed existing presentation documents (for example ppt). Next these are transformed into online based presentations with embedded sound. Selected software architecture is multi-tenant software architecture (on Fig. 5).

3.2 Software functional requirements

Current work represent elicitation, analysis, design, implementation, and testing of web based software, platform independent system, with main feature to transform wide variety of text files and file formats into voice lecture. In current sub-section main functionality provided by the system can be summarized as follows – functional requirements, which corresponds to the software requirements of the system:

- *REQ1: File processing* - reading input files. Reading of HTML, TXT, PDF, DOC and many other file types, generating and creating files with audio speech synthesis (WAV).
- *REQ2: Presentation processing* - parsing and pre-processing presentations. Parse PPT and PPTX presentations and visualize them in web based presenter, with embedded slides reading.
- *REQ3: Resource ratings* - adding file ratings.
- *REQ4: Searching in content* - enable searching in uploaded files name and content.
- *REQ5: User preferences* - maintain user accounts, allowing registration, user login, user logout, file sharing, custom settings.
- *REQ6: Multilanguage support* – support at least English and Bulgarian languages.

3.3 Software non-functional requirements

Non-functional requirements are specified as well:

- *NFREQ1: Modularity,* system is divided into subcomponents and each subcomponent is implemented as subproject in order to foster reusability of software and architecture decisions.
- *NFREQ2: Multi-tenant user-interface*, allowing one user with client software is able to configure and use different content provider server.
- *NFREQ3: Service-oriented decomposition* allowing to evaluate different quality characteristics of services.
- *NFREQ4: Easy to distribute and install* - for lower newcomers to use the server.
- *NFREQ5: User-friendly and simplicity in graphical management interface* allowing good user experience, without heavy settings for administration.

4 Implementation of Audio Lecture Generation Software

4.1 Module structure of software

Main identified modules for implementation are as follows:

- Module 1 Content extraction.
- **Module 2*** Auto-correction.
- **Module 3*** for manual review and process improvement (enrich dictionaries and tools, which filters, compacts and/or enrich presentations) . Also possible cross-cutting concern for modules injection using IoC software principle.
  - dictionary module interceptions;
  - terms dictionary;
  - abbreviations dictionary;
  - quiz and pre-/post- assignments.
- Module 4: Searching and indexing of extracted text (Elasticsearch).
- Module 5: Web services layer (Restlet);
- Module 6: Web user interface layer (SPA);
- Module 7: Database layer and database design (SQLite).

Requirements for modules 2 and 3 was discovered after completing the system, discovering gaps and defects in initial implementation and elicited need for advanced quality tools.

The implemented system uses few third party libraries and resources. For file content extraction is used Apache Tika [16]. For searching in extracted text is used Elasticsearch [17]. Web services are developed using Restlet platform [18]. Used database is SQLite [19].

All third party software libraries and platforms are successfully combined, used, extended and refined to accomplish the goal for maximum user satisfaction. The system can be successfully used by education institutions, as tool in training and learning on workplace, serious games, by disabled people, and regular users. Such working server software can easily be used for making research and experimenting with different quality characteristics of software systems (such as availability, security and so on). User-interfaces are designed as multi-tenant software and currently are tested and adopted for using on one server, but system is designed to be distributed via different 'tenants'. Each tenant can select any server available. One of the aims of the system is to be easy-to-install and easy to use software system. On table 3 are given classification of different of architectures that can be achieved with current implementation. As conclusion using last scenario is hard to achieve. It give to user flexibility to use learning environment in the way they want – either as traditional Learning Management System or as Local/Personal Learning Management System.

Table 3. Web architecture classification

	Server		**Case name**	**Access different devices**	**Multiple users**	
	Local	***Remote***			***Async***	***Parallel***
Local One client (1)	YES	NO	Personal Learning Environment (PLE)	NO	NO	NO
Local many clients (2)	YES	NO	Group Learning Environment (GLE)	NO	YES	NO
Remote One/many client/s (3)	NO	YES	Remote Learning Environment (RLE)	YES	NO	YES
Remote many clients (4)	YES	YES	Multi-tenant learning (MTL)	YES	YES	YES

4.2 Module dependenciues

On Fig. 4, named "*Package dependencies and operations*" above - main modules and their dependencies are depicted.

We have three layers:

- 1st layer: data-store and server middleware – responsible for parsing presentation, indexing content, produce text-to-speech generation.
- 2nd layer: web services layer – with RestletWadlExt module and RestAPI – responsible for reuse of functionality. It decouples client from server, and thus give us possibility to change implementations of client or server in any given time, without breaking functionality of the other layer.
- 3rd layer: client layer – contains of AdminUI module – which is "fat" client (desktop application) and second module is initial prototype of Client UI software (named WebUI).

4.3 Implementation

Authenticated resources case implementation is depicted on Fig. 6 below. Building blocks here are start component, basic state of client software application, authenticated clients, decision module for resources that requires authentication and finally two kinds of audio resources- with public access and with access which requires authentication and authorization processes.

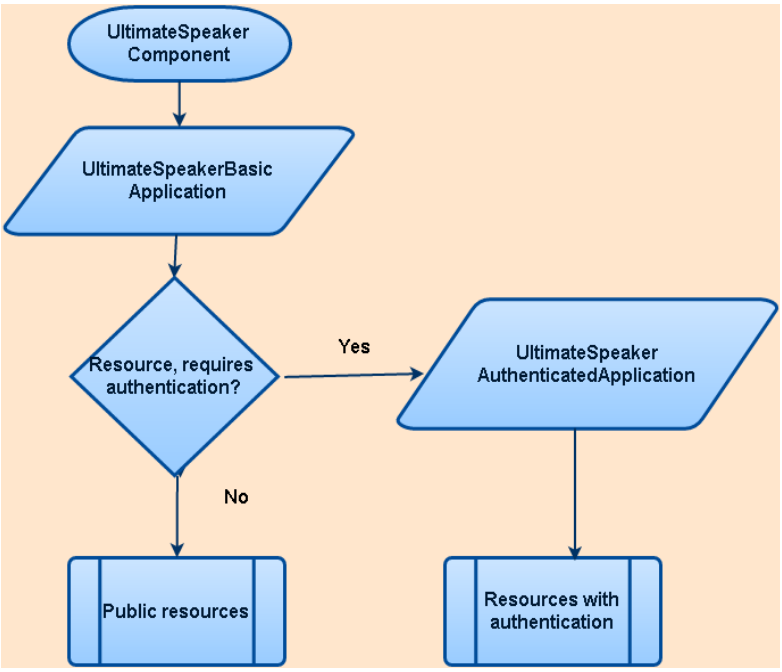

UltimateSpeaker Component
UltimateSpeakerBasic Application
Resource, requires authentication?
Yes
UltimateSpeaker AuthenticatedApplication
No
Public resources
Resources with authentication

Fig. 6. Public and authenticated resources scenario

Here we have main *UltimateSpeakerBasic Application*, which checks if requested audio resource requires authentication. If yes, then go to *UltimateSpeaker AuthenticatedApplication*, authenticate first, and then access the resources. If no, then get access to public audio resources.

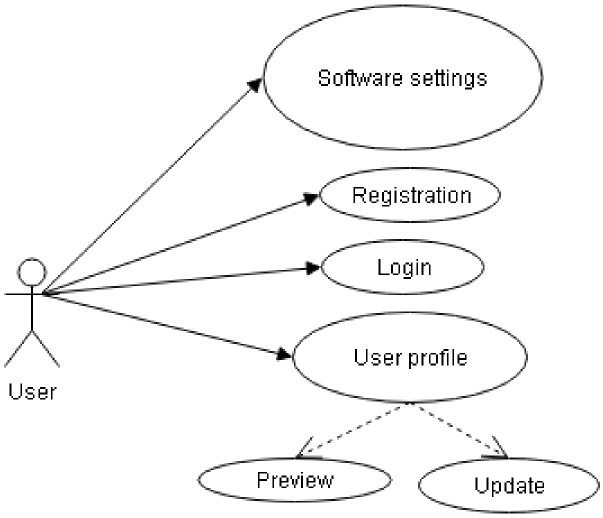


Fig. 7. Main user management use cases

Main scenarios of user management use cases are depicted on diagram Fig. 7. Alongside with common user management use cases such as registration, login, logout and profile creation and update - there are Software Settings, available even before the user is logged to the system. This allow user to change configuration where desired server is located. For example servers can be at local computer (localhost), or at remote IP address.

Depending on the server, there can be located different audio resources – for different courses. Or even resources for the same course, but with different level/skills pre-requests and different intensity of the learning. Scope of the each of the servers can be logically defined and recommended for different types of learners.

Main scenarios of authenticated (registered and logged) user are depicted on use case diagram on Fig. 8. User story here include use cases such view online presentation use case, preview presentation use case, download original file use case, download speech synthesis use case, rate resources use case, and share resource to other registered users.

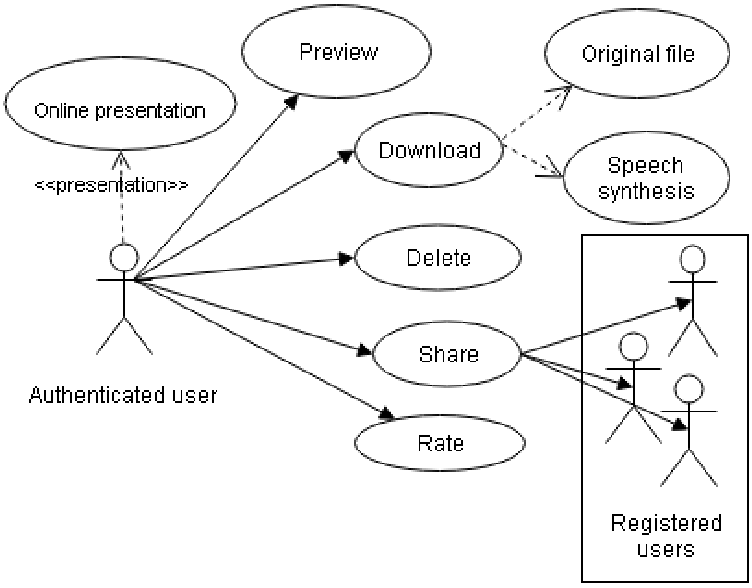


Fig. 8. Main authenticated user use cases

Developed Software Server, called uListenSrv supports all of the described server functionality. How this looks like and how it is tested – is described in next section of current paper.

5 Usage of the system

Here are provided description of basic steps for creation and sharing audio lecture.

5.1 Server-side – start and configure server

First we start server (Fig. 9). Then configure server settings (fig 10), where we follow requirements *NFREQ4 and NFREQ5 (as* described in section IV) *– for simple and user-friendly customer interface*.

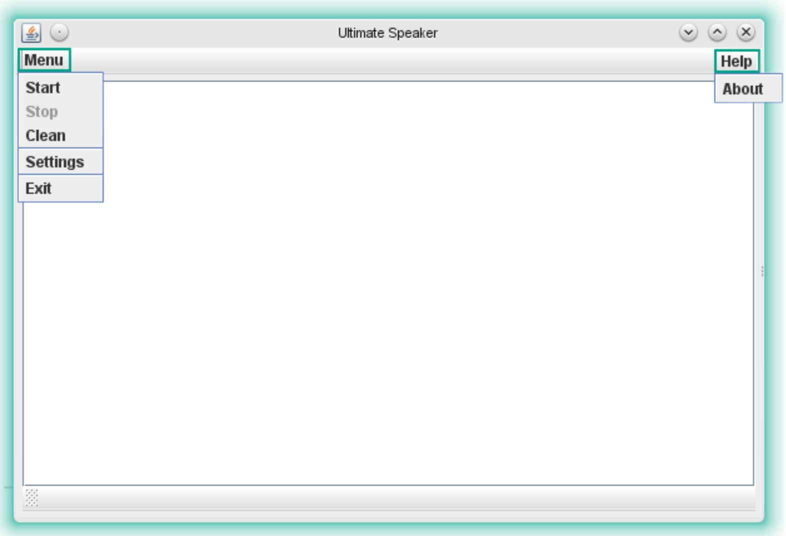
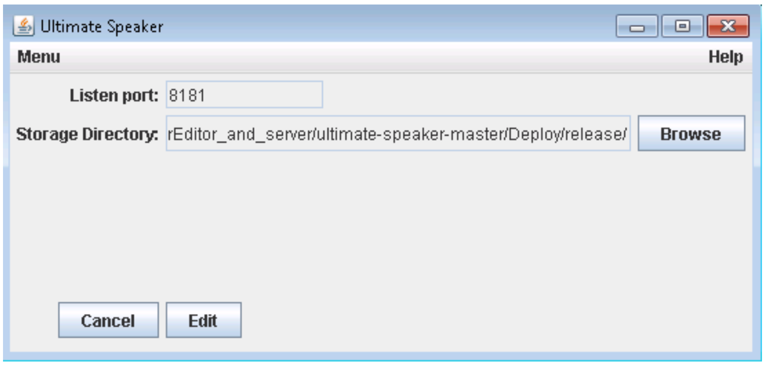


Fig. 9. Screenshot of admin client

Administration client, which itself contains all the server software and management client has modest, but powerful set of options. From here it is possible to conduct major operation with the server software – to start and stop server, to clear cache, to manage server settings, and monitor server log.

Server setting dialog is depicted on Fig. 10 below. There is option to edit server port (by default it is 8181, but can be easily changed. Last, but not least – support of different client is done by configuration of server directory.



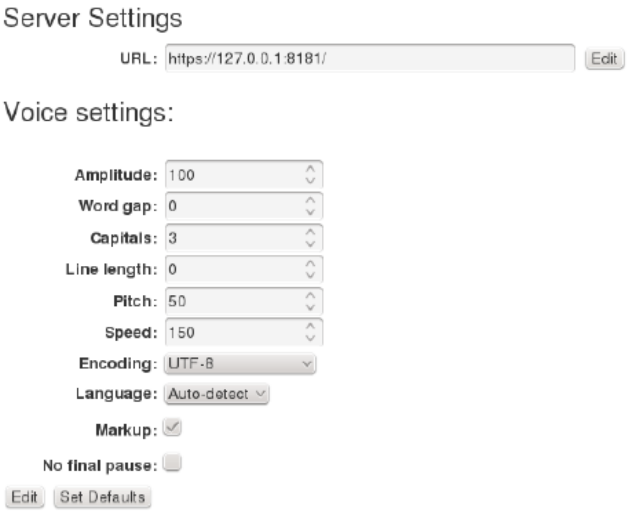
Listen port: 8181

Storage Directory: rEditor_and_server/ultimate-speaker-master/Deploy/release/

Fig. 10. Screenshot for server settings

5.2 Client-side software prototype – create audio lecture

When user is authenticated he or she may or may not change settings of speech synthesis and document processing details (where is appropriate). As available settings are default document encoding (important for pronouncement of digits, numbers and dates, support of markup in the learning material text and many others, mainly related to speech synthesis options. On Fig. 11 is given screenshot from settings panel of client. You need to setup server address, port and audio generation preferences. Here port and URL/IP of the server are combined as one setting.



Server Settings

URL: https://127.0.0.1:8181/ Edit

Voice settings:

Amplitude: 100

Word gap: 0

Capitals: 3

Line length: 0

Pitch: 50

Speed: 150

Encoding: UTF-8

Language: Auto-detect

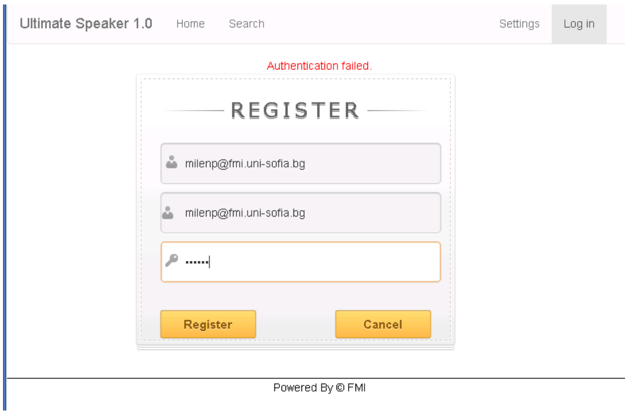
Markup: ☑

No final pause: □

Edit Set Defaults

Fig. 11. Screenshot for web client settings

Before start usage of the system, the user is required to register first (Fig. 12), using appropriate email and password.



Authentication failed.

REGISTER

milenp@fmi.uni-sofia.bg

milenp@fmi.uni-sofia.bg

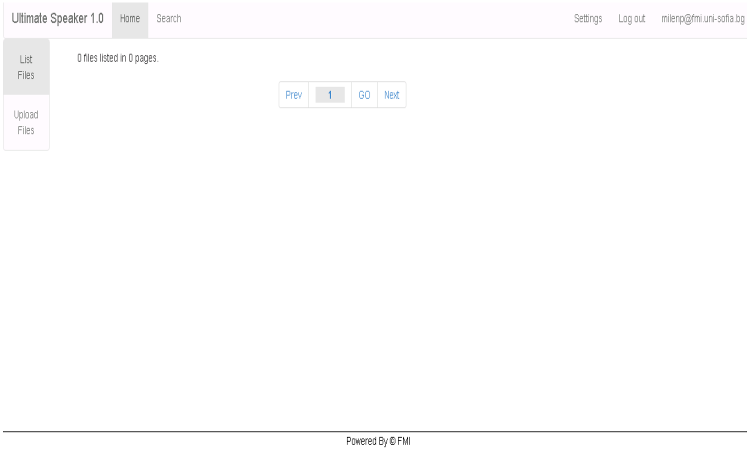
......

Register Cancel

Powered By © FMI

Fig. 12. Screenshot of registration form

After successful registration the user will be logged into the system (Fig. 13), then is needed to go to upload tab page (Fig.14), and to upload desired learning material presentation file (commonly power point format). Newly registered user has no available files. Next they can log in with old user with already existing files (Fig. 15).



0 files listed in 0 pages.

Fig. 13. Screenshot of *new user* list files screen

Before registering a new user or login into the system – it needs to configure URL address and port of the desired server. When user is registered for first time it needs to configure its settings. After registration there are no available files (see Fig. 13).

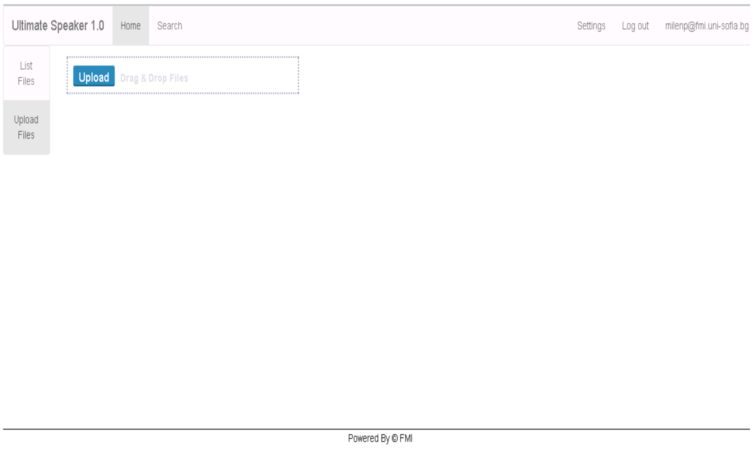
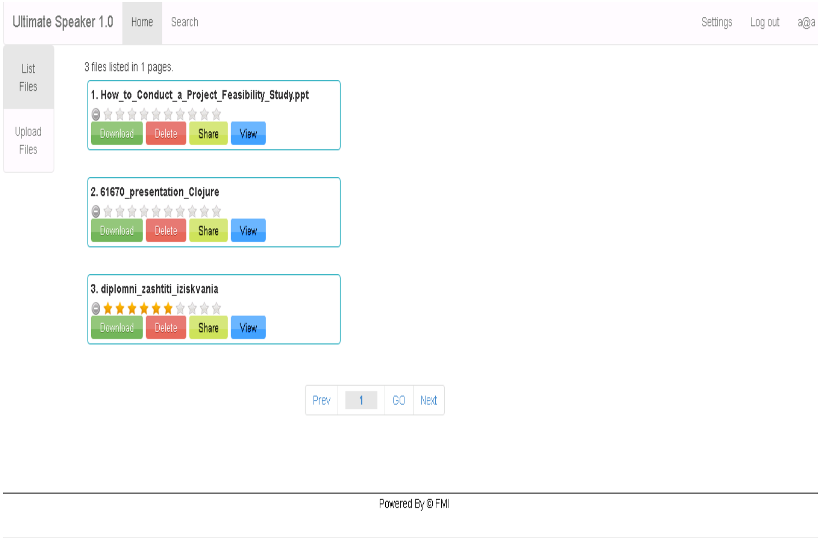


Fig. 14. Screenshot of upload screen

When uploaded – resources are already available and listed in convenient way. You can navigate to the next or previous page or directly to any specific page with resources.



3 files listed in 1 pages.

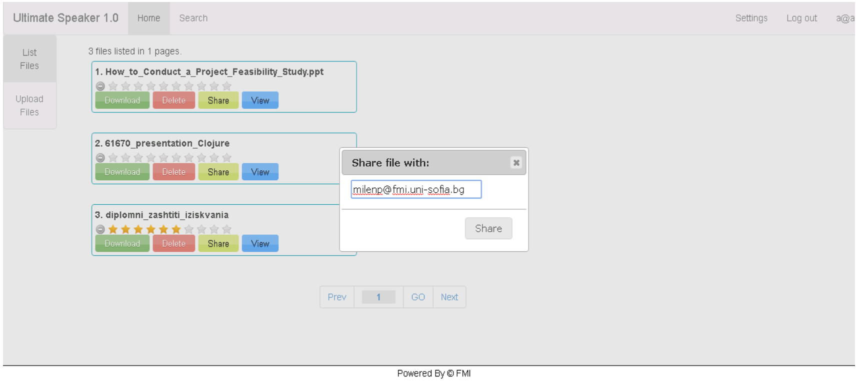
1. How_to_Conduct_a_Project_Feasibility_Study.ppt

2. 61670_presentation_Clojure

3. diplomni_zashtiti_iziskvania

Fig. 15. Screenshot list available resources screen for other *existing user* with several resources

In order to make available generated audio lecture to other external user of the system it is need to share it (see Fig.16). Press *Share* button for the audio resource for sharing, new *Share File Dialog* appears and then enter email address of the user or users and press again *Share* button. As result selected resource is shared to specified user.



3 files listed in 1 pages.

1. How_to_Conduct_a_Project_Feasibility_Study.ppt

2. 61670_presentation_Clojure

3. diplomni_zashtiti_iziskvania

Share file with: milenp@fmi.uni-sofia.bg

Fig. 16. Screenshot for share resources screen

These steps are enough to start using shared document. Usage of shared documents are described in more details in next section.

5.3 Client-side – use audio lecture

Login with another account, which already has shared resources. Example from the client prototype on Fig. 17 is depicted screenshot from rating component in client module in the system. When rate any given resource – you change general rate index of that resource.

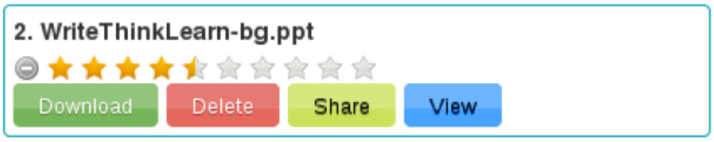
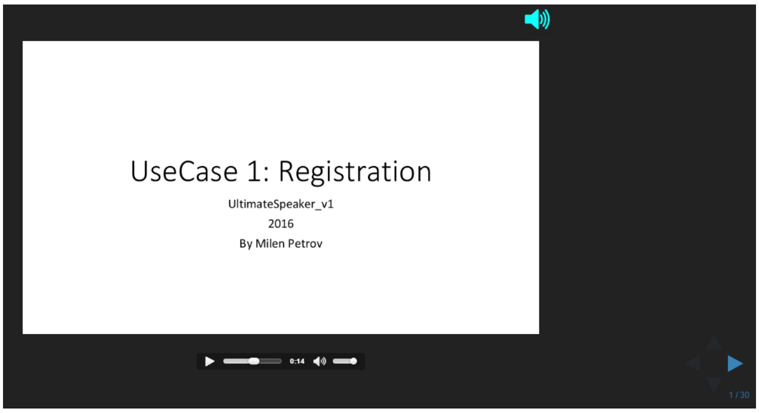


Fig. 17. Screenshot of view and rating by user in web client

First use-case for shared resource can be started by pressing *View* button, then new window is opened and presentation is appeared (Fig. 18). Then you can navigate through the slides, start, stop, pause and mute presentation with standard controls. Moreover there are two modes – for *auto play of slides* when open new slide, or by *manual starting audio*.



UseCase 1: Registration

UltimateSpeaker_v1
2016
By Milen Petrov

Fig. 18. Screenshot of audio-lecture as presented to user in web client

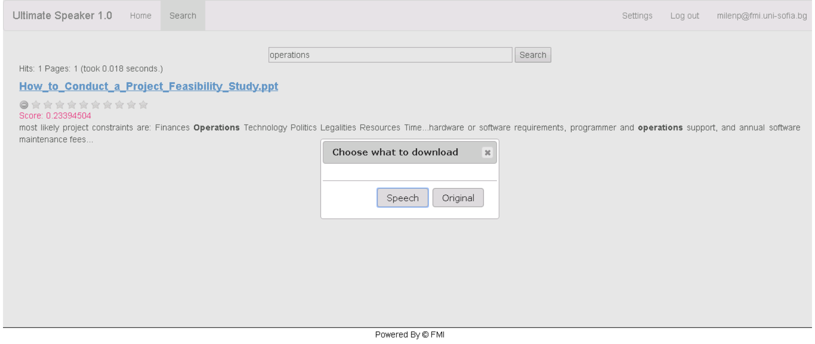
Next you can delete audio presentation or download original presentation (in initial ppt format). You can press *Download* button and system will ask if you prefer to download original file, or speech file (Fig. 19). If you select *Speech* button – the file will be saved locally. Then user can open it with default audio player of the operating system or on mobile device. Use case scenario for downloading only audio file describes a new approach for sharing and using learning materials.

Next you can search all available presentations in search tab. There are available two scenarios – searching in the file name – when stat typing – it shows autocomplete dialog (Fig. 20) – if file name starts with first letters. Or second scenario – you can do full-text search (see Fig. 21).

When resource is located as a result from the search – preview of content is given and keywords are highlighted.

User can view all documents, where matches searching criteria, ordered by their rank. Other feature such as rating of the resource is used to see the rating, credibility or quality of the resource. Searching and free rating of the resources can help different users to create and share lectures.

Last feature, presented here is packaging resources in single page and offline use. That can be achieved as in web client you can press *CTRL+S* keyboard shortcut or save from the browser menu, point download folder and audio presentation with all resources will be saved to local storage for offline use.



Hits: 1 Pages: 1 (took 0.018 seconds.)

How_to_Conduct_a_Project_Feasibility_Study.ppt

Score: 0.23394504

most likely project constraints are: Finances **Operations** Technology Politics Legalities Resources Time...hardware or software requirements, programmer and **operations** support, and annual software maintenance fees...

Fig. 19. Screenshot of download audio file dialog

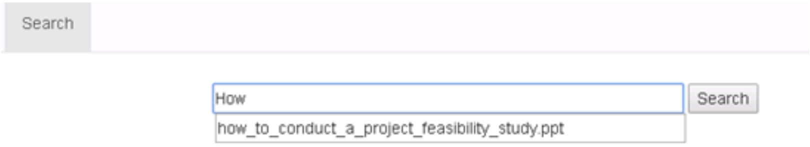
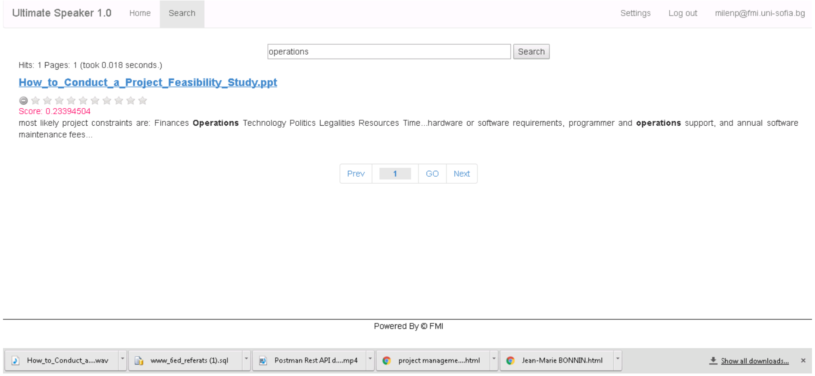


Fig. 20. Screenshot of search for resource and file name auto-complete



Hits: 1 Pages: 1 (took 0.018 seconds.)

How_to_Conduct_a_Project_Feasibility_Study.ppt

Score: 0.23394504

most likely project constraints are: Finances **Operations** Technology Politics Legalities Resources Time...hardware or software requirements, programmer and **operations** support, and annual software maintenance fees...

Fig. 21. Screenshot of full text search for resource

6 Conclusions and Future Work

Defined requirements for design and building software server are achieved. Main activities for future work are planned as follows: Firstly we need to use created software architecture of server and tools to measure satisfaction of students with adding generated audio lectures with the lectures, where are known very well, and using lecturer words, but audio should be produced by usage of audio-generation software, (known as text-to-speech software) compared to recorded audio lecture, by the same known lecturer, and compared to unknown 'expert' reader.

Second we can improve and use proposed software architecture to develop tools for measuring quality characteristics of services (QoS) by using different servers – local and remote, and to measure satisfaction of end user of the system. Proposed architecture is developed in order to investigate and provide a base for experimentation with quality of provided services. As one user (or client) can connect to different providers of functionality (in our case audio lectures) - further building blocks for measure quality of a services can be proposed. Also building quality modules 2 (for auto-correction of incorrect pronounced words) and module 3 (for manual review of audio lecture), as mentioned above.

Modules for measuring quality of services can be added on both sides:

1. In **client side** – each client can decide for themselves what quality of the services are provided, namely:
   (a) Time to response;
   (b) Price - for server-side services: hosting of service, and required hardware or software;
   (c) Quality of learning materials – such as audio, preferences - quality of voices, configurability, etc.;
   (d) Isolation of learning materials in means of securing the materials from external users;
   (e) Sharing of materials (is possible materials to be shared across different users/learners);
   (f) Required hardware or software for client.

With presented scenario there is no need for server-side module – data from the real usage are collected locally. As main advantage here is we know origin of the data and as main disadvantage – it is possible to have small amount of data for each of the services and different characteristics. This can lead to incorrect results.

2. **Server side mode** – we can have centralized approach for gathering quality characteristics. Clients only deliver satisfaction data for services. Services are activated according to their quality with their functionalities. Decision of which service to be provided is taken on the server-side. Advantage: approach is implemented centrally. Disadvantage: complexity of the transmission protocols – as there is need to negotiate quality of the service properties between multiple clients.
3. **Mixed mode** – client and server side modules. Client module communicates with server module (which collects quality characteristics of services) from the client

modules. Advantage: can collect potentially huge amount of data. Disadvantage: complicates server-side implementation has security implication - possibly users are sensitive to sharing their data, which can lead to less data gathered for the improvement of quality of the service.

7 Acknowledgment

Research, presented in this paper was partially supported by the DFNI I02-2/2014 project, funded by the National Science Fund, Ministry of Education and Science in Bulgaria and FNI-SU-2016/192 project - Sofia University “St. Kliment Ohridski” (Bulgaria) - “*Secure and re-usable software architectures for Technology-enhanced learning*”.

8 References

[1] S. M. Jyothirmayee, P. Eliah, and K. M Reddy, "Enhancing Communication Skills through Mini Projects in Engineering Education Using Audio-Visual Aids-A Study.", *International Journal of Scientific and Research Publications*, vol. 4, Issue 3, March 2014, ISSN 2250-3153

[2] W. Jackson, "*Publishing Digital Audio: Content Delivery Platforms*" Digital Audio Editing Fundamentals. Apress, 2015. 131-138.

[3] M. Miller and M. Piller, “*Principal factors of an audio reading delivery mechanism: Evaluating educational use of the iPod*”. In P. Kommers & G. Richards (Eds.), Proc. EdMedia: World Conf. Educational Media and Technology, AACE, 2005, pp. 260-267.

[4] R. Berjon, S. Faulkner, T. Leithead, E. D. Navara, E. O'Connor, S. Pfeiffer, I. Hickson; "*HTML 5, A vocabulary and associated APIs for HTML and XHTML*" W3C Recommendation 28 October 2014, URL:https://www.w3.org/TR/html5.

[5] R. Godwin-Jones "*Emerging technologies: New developments in WEB browsing and authoring*." Language Learning and Technology 14.1 (2010): 9-15.

[6] YouTube Platform, 2016, URL:https://www.youtube.com.

[7] D. Gillet, "*Personal learning environments as enablers for connectivist MOOCs*," Proc. 12th Inernational Conference on Information Technology Based Higher Education and Training, pp. 1–5, 2013. https://doi.org/10.1109/ithet.2013.6671026

[8] R. Kop, H. Fournier, and S.F. J. Mak, "*A Pedagogy of Abundance or a Pedagogy to Support Human Beings? Participant Support on Massive Open Online Courses.*" The Int. Review of Research in Open and Distance Learning. Vol 12, No. 7, 2011. https://doi.org/10.19173/irrodl.v12i7.1041

[9] Documentation of Moodle LMS, URL:https://docs.moodle.org/,

[10] M. Petrov, A. Aleksieva-Petrova, "DESIGN OF REST CLIENT ARCHITECTURE FOR COURSE RESOURCES DOWNLOAD AND PACKAGE", INTED2016. 10th Int. Tech, Edu. and Dev. Conf., 7-9 March 2016, Valence, Spain, ISBN: 978-84-608-5617-7, ISSN: 2340-1079, pp.6513-6521.

[11] "Google Translate." Google Translate. Web. 11 Sept. 2016, URL:https://translate.google.com.

[12] "YakiToMe!", 11 Sept. 2016, URL:https://www.yakitome.com.

[13] "iSpeech.", 11 Sept. 2016, URL:http://www.ispeech.org/.

[14] "SpokenText." Online Text to Speech (TTS) Converter. URL: 11 Sept. 2016, URL:http://www.spokentext.net/.

[15] A. Asenov, "*System for synthesis and visualization of text from wide variety of sources*", (In Bulgarian), Master Thesis, Dept. Soft. Eng., Sofia Univ., Sofia, BG. supervisor: M.Petrov, 2015.

[16] C. Mattmann, and J. Zitting, *Tika in action*, Manning Publications Co., Greenwich, CT, USA, 2011.

[17] C. Gormley, and Z. Tong, *Elasticsearch: The Definitive Guide"* O'Reilly Media, Inc., 2015.

[18] J. Louvel, T. Templier, and T. Boileau. *Restlet in Action: Developing Restful Web Apis in Java,* Manning Publications Co., Greenwich, CT, USA, 2012.

[19] M. Owens, and G. Allen. SQLite. Apress LP, 2010.

9 Authors

Milen Petrov is assoc. prof. at Sofia University, Faculty of Mathematics and Informatics, Department of Software Engineering, J.Baucher 5 blvd, Sofia, BULGARIA (e-mail: milenp@fmi.uni-sofia.bg).

Asen Asenov is Master Thesis Student at Sofia University, Faculty of Mathematics and Informatics, Department of Software Engineering, J.Baucher 5 blvd, Sofia, BULGARIA (e-mail: aasenov.bg@gmail.com).

Adelina-Aleksieva-Petrova is assoc. prof. at Technical University of Sofia, BULGARIA (e-mail: aaleksieva@tu-sofia.bg).

This article is an extended and modified version of a paper presented at the International Conference on E-learning in the Workplace 2016 (ICELW 2016), held in June 2016, at Columbia University in New York, NY, USA. Article submitted 30 November 2016. Published as resubmitted by the authors 23 January 2017.