České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra počítačů

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Tkachenko Danylo

Studijní program: Otevřená informatika
Obor: Softwarové systémy

Název tématu: Repozitář a server webová aplikace pro nahrávky dialogů postav

Pokyny pro vypracování:

Realizujte vhodný back-end (repozitář a server) pro webovou aplikaci pro správu a pořizování nahrávek dialogů postav počítačových her, rozhlasových her a animovaných filmů, využitelnou nezávislými tvůrci těchto uměleckých děl na základě konzultace se zadavatelem. Zaměřte se i na vhodné technologie streamování zvukových záznamů a na techniky ochrany dat. Back-end prodrobte zátěžovým testům.

Funkce back-endu:
1) Registrace a přihlášení uživatele
2) Vytváření a editace projektu
3) Vytváření a editace dialogu
4) Tvoření hlášek k nahrání
5) Pořizování a ukládání nahrávek
6) Vytváření komentářů k nahrávkám
7) Informace přes e-mail o pořízení komentářů, registrace, obnovení zapomenutého hesla
8) Export hotových nahrávek do archivu
9) Import projektů z nástroje Chat Mapper

Seznam odborné literatury:

[1] Peter Ladefoged: Elements of Acoustic Phonetics

Vedoucí: Ing. Adam Sporka, Ph.D.

Platnost zadání do konce letního semestru 2017/2018

[redacted]
vedoucí katedry

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta elektrotechnická
L.S.
-4-

[redacted]
děkan

V Praze dne 18.1.2017

Czech Technical University in Prague
Faculty of Electrical Engineering

Bachelor Thesis

“Voicer project” — Repository and Server Back-end for Voice-over Acquisition and Management

Danylo Tkachenko
Supervisor: Ing. Adam Sporka, Ph.D.

01.03.2017

Acknowledgment

I would like to thank Ing. Adam Sporka, Ph.D. for his support and collaboration in the preparation of this thesis.

In addition, I would to thank my mother Victoria, who provided me this incredible opportunity to study abroad.

Author’s Affirmation

I hereby declare that the submitted thesis is exclusively my own work and that I have listed all used information sources in accordance with the Methodological Guideline on Ethical Principles for College Final Work Preparation.

Prague 02.05.2017 ....

Abstract

This thesis describes a process which I went through developing an application for improving project collaboration. It specifically targets audio-recording related projects, which require remote teamwork. First of all, a problem analysis is prepared. Based on that analysis the application architecture is designed. After that, the application is implemented and tested. In conclusion, there is an overview of the project goals completion and evaluation of the results.

This thesis is connected to the user interface design part of the “Voicer project” of Sergey Krasotin’s bachelor thesis which he defended in June 2016 [16].

Keywords
Audio recording, Video games, Voice-vers, Collaboration, Java, Hibernate ORM, Spring Framework

Abstrakt

Tato bakalářská práce popisuje proces, kterým jsem prošel během vývoje aplikaci na zlepšení spolupráce na projektech. Zaměřuje se konkrétně na projekty související s nahráváním zvuku, které vyžadují dálkovou týmovou práci. Především byla připravena analýza problémů. Na základě této analýzy je navržena aplikační architektura. Aplikace je poté implementována a otestována. Závěrem této prací je přehled o dokončení cílů projektu a vyhodnocení výsledků.

Tento projekt se navazuje na bakalářskou práce Sergeye Krasotina, který ji obhájil v červnu 2016, a která se zaměřuje na design a implementáce uživatelského rozhraní [16].

Klíčová slova
Nahrávaní hlasu, Video hry, Voice-overs, Kolaborace, Java, Hibernate ORM, Spring Framework

Table of Contents

1. Introduction

Working in a software development company, I noticed that people, working in a team, do not have the right tools for their teamwork and collaboration. It is simple to work within a small team, as there is only a little amount of inputs coming from each team member, and everybody seems to keep up with the project pace quite effortlessly. However, as the team grows, it gets harder to organize their work, and efficient teamwork becomes a challenge. In big teams, there always should be a framework, that describes the process, how everything in the project should be done. This process outlines a set of activities that the team must follow to achieve the success. Even though, there are different modern methodologies, how to do it properly, there are still many industries, where this process is yet to be adopted.

This is the exact case of audio capturing businesses, such as game development studios, animated movie studios or dubbing companies. Those are the companies, that are creating a lot of media content, and require multiple people to work on the same project simultaneously. It also happens, that the team members are not sitting in the same space but are working remotely, and are located in different cities or even countries. In those cases, the lack of the right toolset and necessity of legacy software usage raises huge difficulties for the team to succeed.

Those are the types of problems that led this thesis to see the light.

1.1 Project goals

The main goal of the project is to create a robust tool, that can be used by the teams which require remote project collaboration. It should provide them the right instruments to gather all work in one place and synchronize their results. Having such an application would tremendously simplify the process and cut project expenses by creating new opportunities for smaller businesses. This would result in a significant market growth and creation of a better content for the consumer.

Project goals highlight:

A. Simplify audio content management.
B. Provide a project collaboration platform.
C. Create a single storage for the whole project.

This project has been a collaboration between Sergey Karotin and me. My role was designing and implementing of the back-end project, and his role was creating the user interface for the project. Sergey Krasotin’s part has been described in his thesis which he defended in June 2016 [11].

The goals of this thesis reflect my role in the project and may be outlined as follows:

- Design and implement the application architecture

- Propose an appropriate database model
- Create the application back-end and connect it to the front-end

2. Problem analysis and solution proposal

To understand better the problem and find out who are the users, I first analysed current market situation. Having done the research, I found out that there are 2 types of studios, where this project can be adopted.

Game development studios
Game development studios are software development companies. They can range from very small, that have only few employees, to big businesses, that have hundred of thousands of employees.

Most games have in-game characters and if they have voices, they require voice actors to record their lines. Many game development studios work with voice actors on site only and do not support remote cooperation. The reason is that it is hard to manage all the materials, which are being produced and to communicate all the changes effectively. This puts huge constraints on the project, as it is often required, that the character has a specific accent or even speaks a different language. If the budgets allows, studios invite voice actors to travel from abroad to work on the project locally. This can be very expensive, as the company has to cover accommodation and travel expenses. This option is suitable mostly for big studios, as the cost of inviting voice actors from abroad is very high.

Smaller studios usually don't have enough money to cover expenses of the voice actors coming from abroad, so they are trying to recruit freelance voice actors over the web, who are in many cases part-time workers or amateurs. This can be quite inefficient because every voice actor uses their own tools for recording and editing the audio, which causes confusion in work synchronization.

There are approximately 200 game development studios, which are currently active worldwide, this doesn't include very small studios [1].

Animated movies and cartoons studios
Companies, which are creating animated movies and cartoons, often need a native speaker to work on a project, which leads to hiring people from abroad to contribute on the project. This again can drastically inflate the project's budget.

There are approximately 250 animated movies studios, which are currently active across the globe [2].

2.1. Scenarios and opportunity assessment

To find the areas of possible improvements and get better understanding of the workflow, I studied how those target audiences currently work and described it in scenarios. It helped me to understand their goals and pain points and to create an opportunity assessment. Pain points are those, that frustrate the users. Opportunity assessment is a process of evaluation of an idea to find out whether there is a sufficient value in it.

Game development studios' workflow

1. Define scenes, where dialogs between character are needed.
2. Describe the scene and create the script.
3. Distribute the script to the voice actors.
4. Voice actors prepare and record their lines.
5. Voice actors send recordings back to the project manager.
6. Project manager analyzes the recordings and decides if they are acceptable.
7. If the recordings are not accepted, the project manager prepares corrections and sends them to the voice actor. Otherwise, the project manager sends the recordings to the sound engineer.
8. The sound engineer cuts the recordings and aligns them to fit the timing in dialogs.

Animated studios workflow

1. Prepare the script with line timing.
2. Make an audition and find a voice actor to contribute on the project.
3. Send the script to the voice actor.
4. Voice actor prepares recordings.
5. Voice actor sends recordings back to the studio.
6. Project manager reviews the recordings and adds corrections.
7. Project manager sends the corrections to the voice actor.
8. Voice actor corrects and records the lines, that need correction.
9. Voice actor sends recordings back.
10. Project manager reviews the recordings and if they are correct, passes them down to a sound engineer.
11. Sound engineer prepares the audio mix and cuts the pieces, so timing fits the frames.
12. If the recordings are accepted and fit the frames, project moves on to the next scene/dialogue, otherwise, the process repeats from the step 6.

It is clearly visible, that there is a great inefficiency in communication between the studios and voice actors, which leads to a slow progress and increased project expenses. Those are the areas that the project attempts to solve.

2.3. State-of-art approaches

In order to understand what areas should be improved in the current process, I analysed the tools, that are currently used by the studios to run a project. I outlined positive and negative sides of each tool to find out what an ideal solution for their job should contain.

Emails — it is one of the legacy tools, which despite having many better alternatives, still remains one of the most used mean of communication between remote team members.

Advantages	Disadvantages
+ Everyone has a mailbox + Simple user interface + Free	- Hard to keep track of the conversation with multiple users - Not suitable for a large data transfer - No instant response - No commenting system

Table 2.0: *Advantages and disadvantages of emails*

Google Drive[1] — it is a modern tool for cloud storage solution. You can upload and share your files with anyone.

Advantages	Disadvantages
+ Easy way to manage your files + Fast and free for small storages	- You can't preview specific audio file formats - No commenting system - You have to manually manage your project structure. - You can't share a part of a single document

Table 2.1: *Advantages and disadvantages of Google Drive*[1]

[1] https://www.google.com/drive/

Google Docs[2] — it is a text collaboration tool.

Advantages	Disadvantages
+ Instant collaboration for multiple users + Free and fast + Commenting system	- Audio attachments are not supported

Table 2.2: *Advantages and disadvantages of Google Docs*[2]

Trello[3] — simple task tracking application.

Advantages	Disadvantages
+ Easy task managing + Tasks can be assigned to the team members + Supports Kanban project methodology + Free	- Multiple projects are hard to maintain - Granular project access cannot be provided. - Attachments preview is not supported.

Table 2.3: *Advantages and disadvantages of Trello*[3]

[2] https://docs.google.com/
[3] https://trello.com/

Atlassian Jira[4] — a software development tool for agile teams.

Advantages	Disadvantages
+ The tasks are clearly visible. + It is possible to assign the tasks to people + Project statistics and reporting + Commenting system	- Expensive - Too complex for a regular using without previous Jira experience - Granular project access cannot be provided

Table 2.4: *Advantages and disadvantages of Atlassian Jira*[4]

It is clear, that currently, there is no tool on the market, that would provide the right tool set and combine all the user needs. “Voicer project” is focused on the advantages of the currently available tools and tries to combine them to create a better user experience.

2.4. Requirements

Based on the previous analysis of the state-of-art tools, scenarios and user interviews, following list of requirements has been created which describes what an application should contain to fulfill the user needs. User interviews are described in the Sergey Krasotin’s bachelor thesis [11].

Functional requirements

Functional requirements are focused on the users’ perspectives and describe how the system should behave and what it should allow.

User should be able to:

- share projects with other users;
- record audio for a line;
- invite another user to collaborate on the project via email;
- create dialogs in a project;
- login to edit project;
- export a project to an archive;
- assign a character to a voice actor.

[4] https://www.atlassian.com/software/jira

System should support:

- project creation, editing and deletion;
- line creation, editing and deletion;
- project import from a file;
- user account creation;
- resetting account password;
- logging in and logging out;
- characters creation, editing and deletion;
- line acceptance or declination;
- adding comments to a line;
- displaying the current progress of the project. System should reflect how many dialogs in the projects are done. How many lines it is needed to evaluate;
- reordering of the lines and editing;
- displaying the number of unassigned actors.

Non functional requirements

Those requirements contain different limitations that are applied to the project.

- All the data are to be stored in a relational database compatible with JDBC.
- The application will be accessible via a web browser using HTTP connection.
- Presentation layer will be communicating with backend using JSP (Java Server Pages)
- Application will be tested by using unit and stress tests.
- System should save recorded files locally.

3. Implementation

After research and analysis are done, I proceeded to implement the application. In this chapter it is described how the application was built and developed.

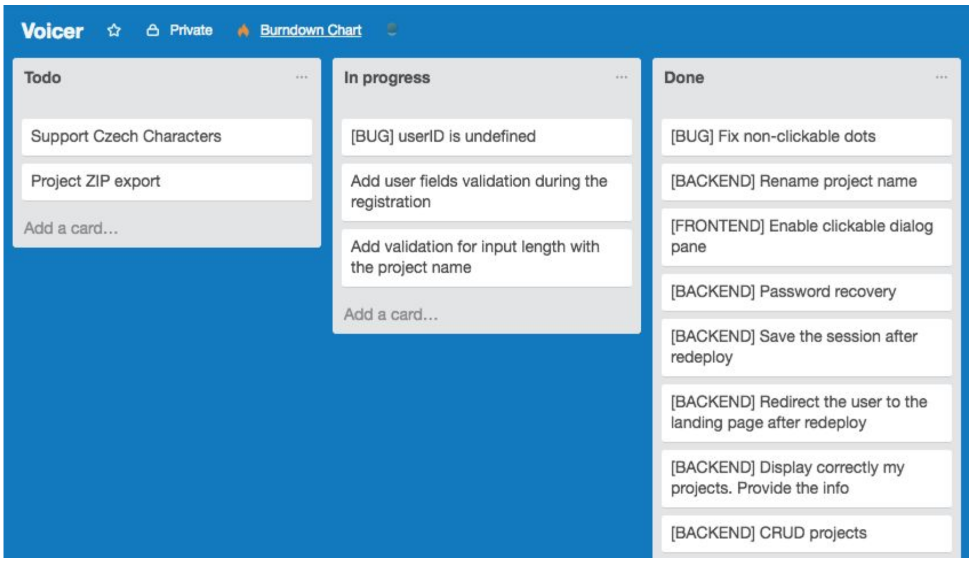
The back-end implementation of the application is connected to the front-end part of Sergey Krasotin's bachelor thesis, who designed the user interface and application interactions [11]. This thesis focuses on the implementation of the back-end part of the project only, and doesn't cover any UI related work.

3.1. Development method

For this project I chose to follow Kanban methodology as it best fits project requirements and supports easy work tracking [3].

The work of all kanban teams revolves around a kanban board, a tool used to visualize work and optimize the flow of the work among the team. While physical boards are popular among some teams, virtual boards are a crucial feature in any agile software development tool for their traceability, easier collaboration, and accessibility from multiple locations.

I created a virtual Kanban board using Trello[5] application. I divided my project board into 3 parts: Todo, In progress, Done. Then I created tickets which represent the tasks, that are needed to be done. As I progress, tasks were moving from the left to the right, indicating project progress. Thus, I've been able to track the work properly and to communicate the project's progress very easily.



Todo
- Support Czech Characters
- Project ZIP export
- Add a card...

In progress
- [BUG] userID is undefined
- Add user fields validation during the registration
- Add validation for input length with the project name
- Add a card...

Done
- [BUG] Fix non-clickable dots
- [BACKEND] Rename project name
- [FRONTEND] Enable clickable dialog pane
- [BACKEND] Password recovery
- [BACKEND] Save the session after redeploy
- [BACKEND] Redirect the user to the landing page after redeploy
- [BACKEND] Display correctly my projects. Provide the info
- [BACKEND] CRUD projects

Figure 3.0: *Kanban board*

[5] https://trello.com/

3.3. Architecture overview

The application is based on the Spring framework and implements main design patterns of a 3-layer web application [5]. The “Model-View-Controller” architecture is the main skeleton of the application. This design pattern allows to separate presentation layer from the implementation one. That supports low coupling and high cohesion, which greatly improves project’s sustainability.

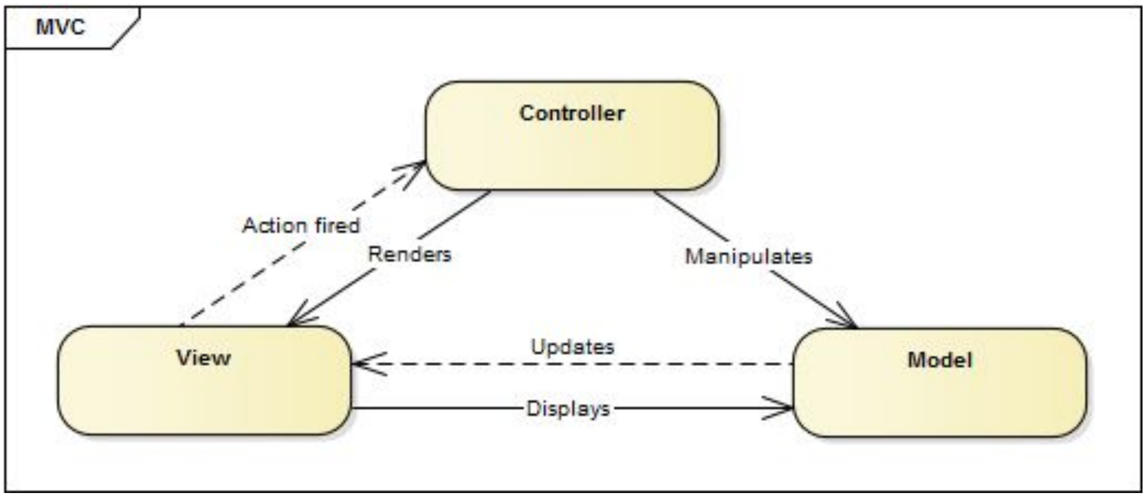


Figure 3.1: Model-view-controller architecture

Models

A model stores data that is retrieved according to commands from the controller and displayed in the view. They indicates the application's behavior and they are independent of the user interface.

Views

Views are what we see, when the application is opened in the browser. It is done using JSP pages, that are extended form of HTML with java support inside of them. Using JSPs it is possible to create dynamic web pages.

Controllers

Controllers are used for request handling, and are part of the MVC design pattern [5].

Basic algorithm is that controllers detect request mapping and delegate the request to services. After all business logic is done, it updates the view or redirects to the next view based on the incoming request.

Using the controllers we separate the business logic from the input processing and support high cohesion and low coupling.

Services

Services implement business logic of the application, they handle all the input requests received from controllers.

Example: When the user requests to login, controller gets the input values and sends them to the login service. Login service checks the data using DAO implementation and returns the result of the operation. Based on the result, controller changes the view. Detailed example of service-controller process can be seen on the figure #4.

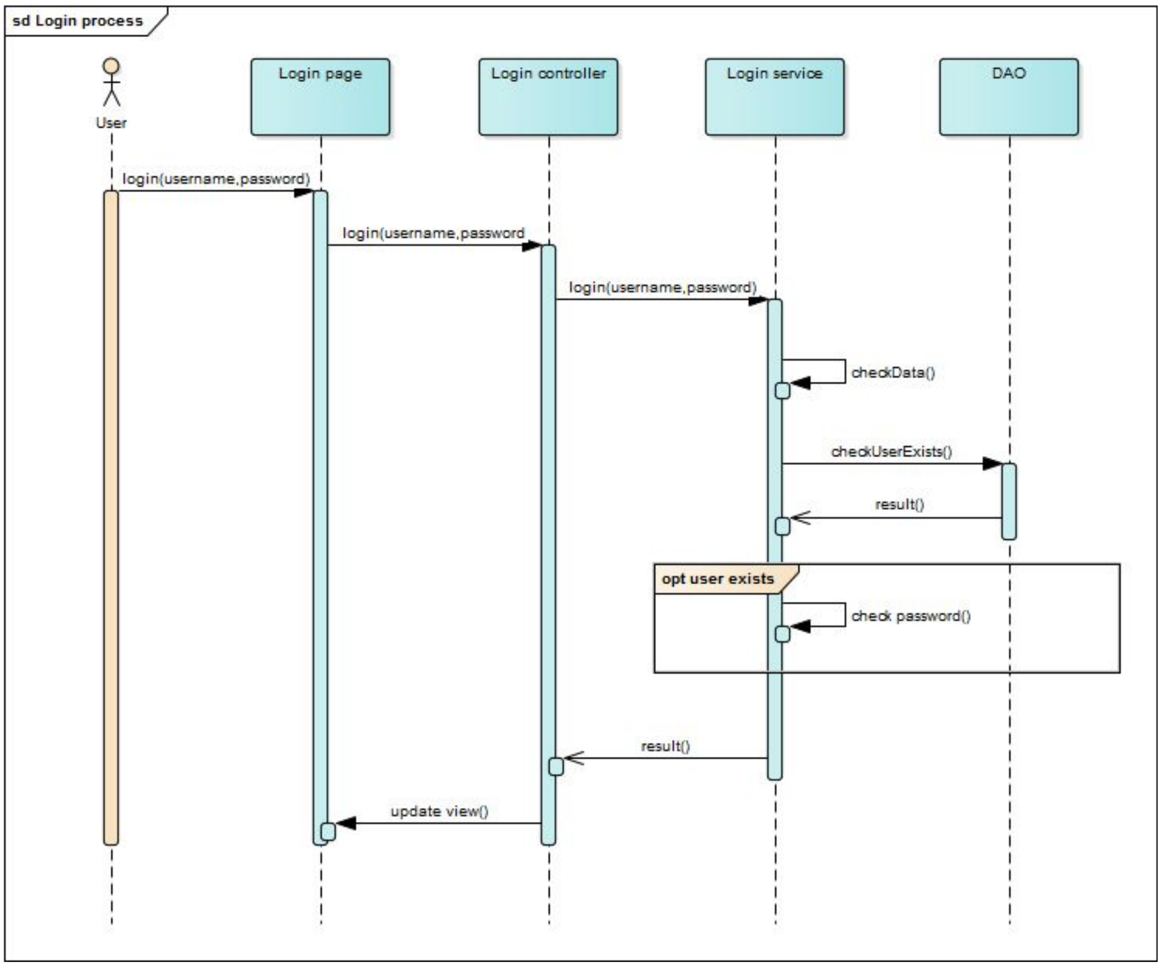


Figure 3.2: Login process sequence diagram

3.4. Use cases

This is the main overview of the project defined use cases, which are based on the functional requirements.

Business actors are User, Voice actor and Project manager, all are playing integral roles in relation to Voicer application.

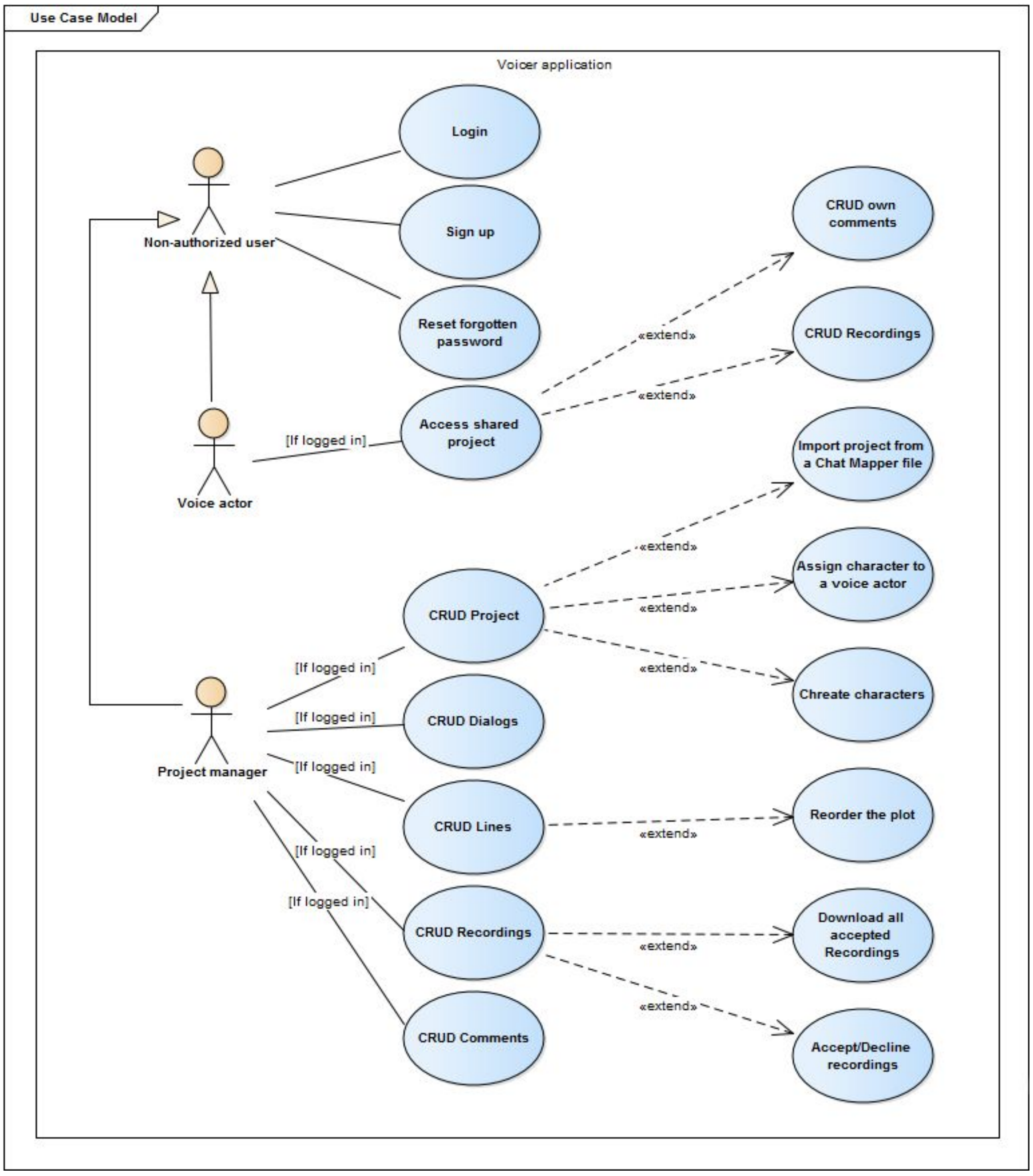


Figure 3.3: Use cases overview

3.5. Entities

Entities are created based on the use-cases and represent the objects that need to be persistent in the system.

User

User has a unique id, password, username, email and enable flag. Enable flag serves as a switch between blocked(banned)/unblocked states. Also, user entity can have different roles (Project manager/Voice actor).

Project

Different projects can be accessed by multiple users with different roles. Depending on the user role, one can access different actions in the project. For example, Voice actor can record audio in the project and project manager can leave comments and edit the plot. Project has unique id, name and date when it was created.

Dialog

Each project can have multiple dialogs. Dialog has a name and dates when it was created and last changed.

Line

This entity is needed because every dialog can contain multiple lines, on which the dialog is based. Line has its name and text.

Comment

This entity is required for the project manager to be able to comment a recording, so that voice actors knows what needs to be improved.

Recordings

Each line can have up to 3 recordings, which represent 3 takes for the voice actor to record a line. A recording can have one of 3 possible states (Recorder, Accepted, Declined).

Character

A character entity is assigned to a voice actor, so the voice actor can access the project and record required lines.

Following diagram provides detailed overview of the database architecture.

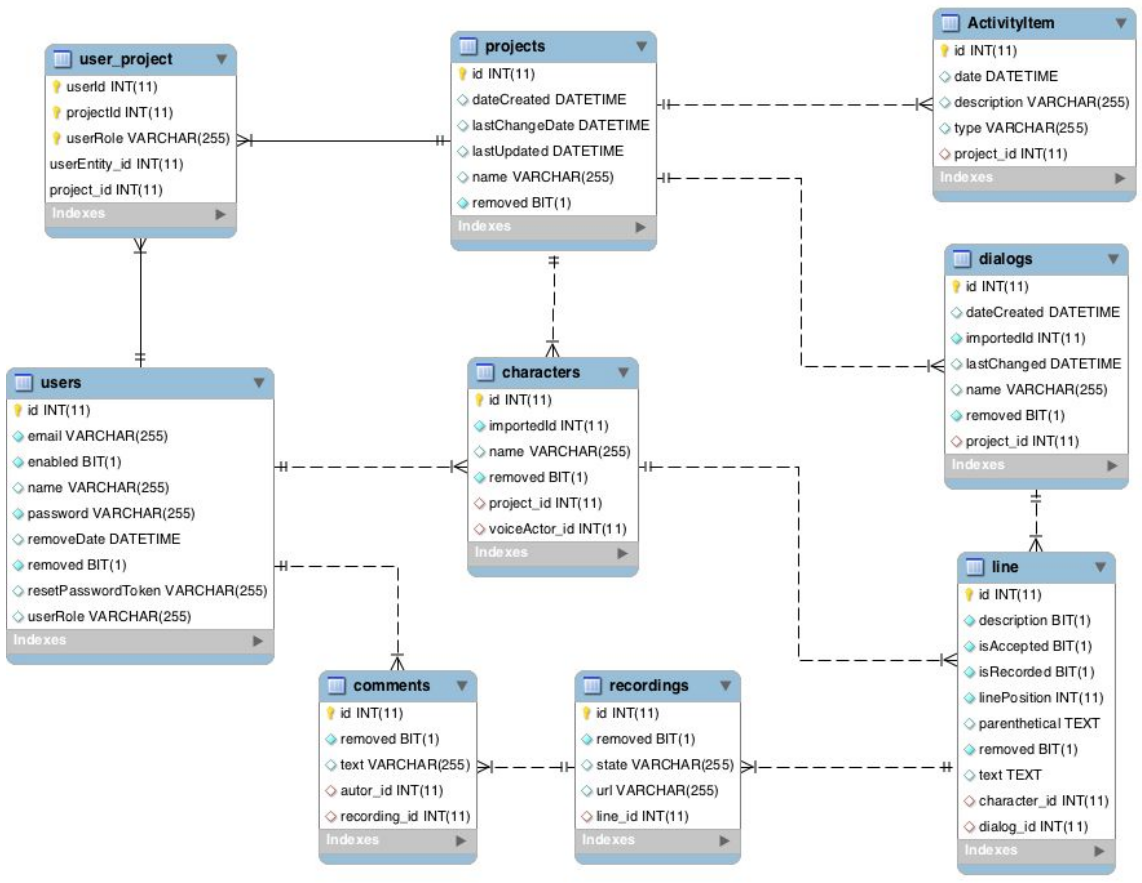


Figure 3.4: Entity relationship diagram describing the logical model.

3.7. Used technologies

There are several technologies that are used in the project.

Programming Language

As a programming language for this project I've chosen Java because it has proven itself over the years as a one of the best options for building scalable and reliable web applications. It is a platform independent language, thus, the final applications can be easily deployed to any application server. The application then will be running inside an application container on the server.

Hibernate ORM
For persistence implementation I used Hibernate ORM framework [4]. It makes all the work of creating database tables and database access. Using such technology makes it possible for us to access database in form of Java objects without manual object conversion.

Web Framework
The application is based on the Spring MVC framework [5]. It separates business logic from data processing and keeps the complexity of the project much easier to maintain. It is also highly compatible with Spring Security framework, which is the security core of the application [6]. It helps to restrict access for different user roles and data encryption. All passwords are encrypted with Bcrypt algorithm, which uses salt [7].

Apache Maven
The application uses Apache Maven framework to simplify the build process and to make it automatic. All configurations are located in the POM file. This file contains dependencies and build steps, that are needed to build the application. Maven can automatically download dependencies from central repositories and simplify libraries management. Resulting artifact is stored in the form of a WAR[6] file.

Web Server
I've chosen Apache Tomcat , as it is one of the most popular application servers to execute Java servlets [9]. It also supports Java Server pages, that was chosen for presentation layer technology. It is lightweight, open-sourced and highly flexible. It has a lot of built-in customization options and can be easily tweaked to adjust application needs. It has also proven itself as an extremely stable platform to build on.

Persistence
As a database management system, I chose MySQL [10]. It is a free, open-sourced database management system. It is quite flexible and scalable. Hibernate ORM framework is connected to MySQL. This allows me to work with the database in terms of objects implementing Data Access Objects pattern.

JavaServer Pages (JSP)
This technology provides ability to create dynamically generated web pages, extending HTML and XML. Java server pages are easy to read and the technology is quite flexible.

Security
The application uses Spring Security framework [6]. It provides tools for secure authorization for Java applications. It is highly extensible framework, so it can easily support custom requirements. It also has deep Servlet API and Spring Web MVC framework integration [5].

[6] Web archive file

Authentication is done by using Bcrypt password hashing function [7]. It incorporates salt to provide a one-way hashing function. The salt is contained in the password, so even if the encrypted string is intercepted, it cannot be decoded without it.

Repository
Currently, audio files that are uploaded to the tool are located on the server's hard drive. The path to the file is persistent in the database, so the file can be later retrieved.

3.8. Configurations

Spring applications are configured by xml config files. This is very convenient, as the configuration is clear and easy to find. Currently, there are 4 configurations files in the application.

spring-config.xml

This configuration file defines persistence-needed resources for the database connectivity. There are two beans: dataSource and sessionFactory.

Data source bean provides hibernate-needed properties, such as database driver, database URL, username and password. Those properties are essential for the database connection.

Session factory bean has injected data source bean and hibernate properties, such as hibernate dialect and hbm2ddl.auto.

Moreover, session factory bean has a list of annotated classes with values. It represents all entities, that are included in the project and going to be persistent through hibernate framework. This list is required to delegate all specific model-related logic to hibernate. For example, if we change the structure of a model and it is recognized as an entity, hibernate framework will automatically change database structure to add those updates.

security-context.xml

This configuration file is needed by Spring security framework and defines security rules for the application. It has an authentication manager, where authentication process specifications are provided. For this project BCrypt password encoder is used.

In this configuration file we also provide accessibility restrictions of different URL routes. Each rule defines a specific URL pattern and its access parameter.

It is a good design pattern to restrict the access to all URL routes, and to allow the access to specific routes only. It helps to keep the application secure.

There are also login and logout redirect pages defined.

dispatcher-servlet.xml

This configuration file is the main access point for the whole web application. It defines rules for the JSP files locations (prefix and suffix), and resource mappings. It translates the requests and maps them to the real files.

web.xml
This is a web application configuration that each java based web application has. It defines a set of rules to bootstrap the application. There are defined error pages for each http code.

3.9. Deployment

Deployment requires an application server which supports Java 8. Currently it is used with Apache Tomcat server. Also it requires MySQL RDBMS to store the data. Apache Maven plugin generates a web archive file, which is then deployed and run on the application server.

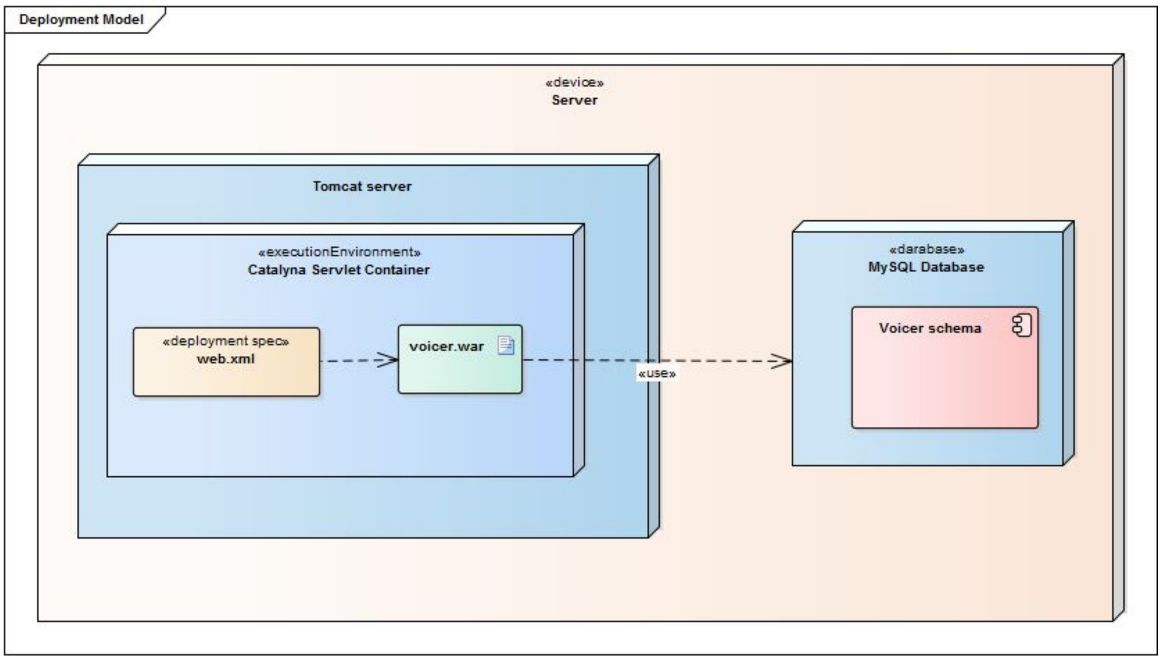


Figure 3.5: Deployment diagram

3.10. Final application description

This is a description of the final result. The UI was designed by Sergey Krasotin and is described in his part of the bachelor thesis [11]. This part describes how the back-end is communicating with front end.

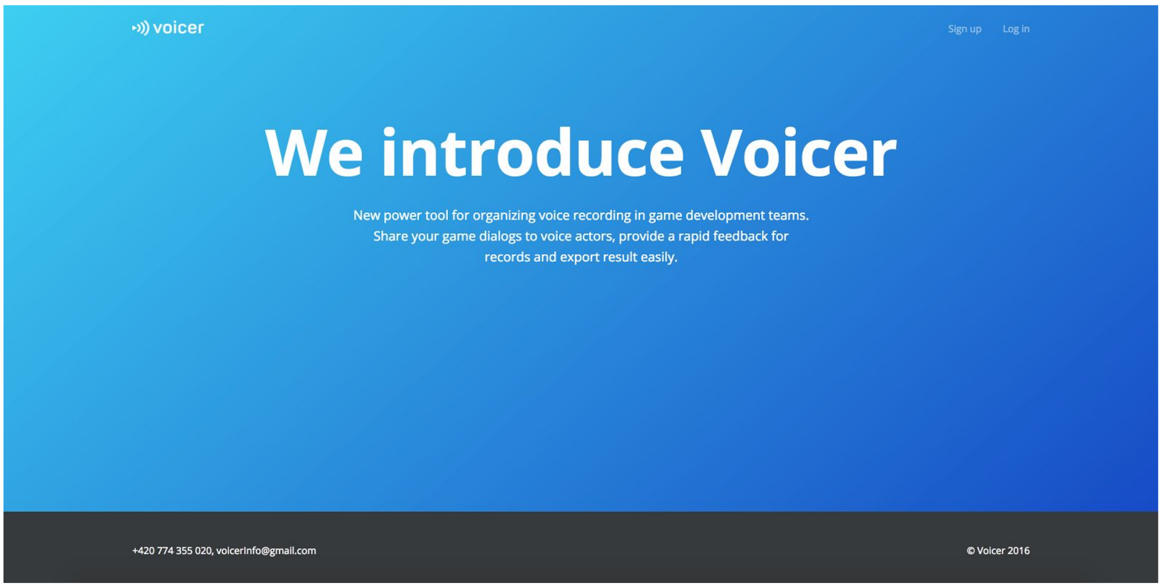
Main page

This is the first page that the users see when they open the application in their browser. From this page on, it is required to be logged in to create and open projects. It is going to be extended by marketing content in the future, which is going to describe the product’s benefits and attract the customers to try the product.

If the user is already logged in, the system will redirect the user to the “My projects page”. If the user does not remember the password, there is an option to click on “Forgot password” link and restore the it. In case of forgotten password, a unique URL link will be generated and sent to the user’s email. Afterwards, if the user clicks on the link in the email, they will be redirected to the page where they can enter a new password.

Use case	**Status**
Login	Implemented
Logout	Implemented
Register	Implemented
Password recovery	Implemented

Table 3.0: *Uses cases completion overview of the Main page*



We introduce Voicer

New power tool for organizing voice recording in game development teams.
Share your game dialogs to voice actors, provide a rapid feedback for
records and export result easily.

+420 774 355 020, voicerinfo@gmail.com

© Voicer 2016

Figure 3.6: Main page

My projects page

On this page there are all the projects that belong to the user, or the user has access to. There is an option to create a new project or to import a project from a Chat Mapper[7] project file.

There are also options to edit the project's name, download all finalized records and archive the project. All those options can be accessed from a contextual menu, which is located on the project card and is represented by 3 vertical dots icon.

Each project contains information about unassigned actors, records that are needed to be evaluated and dialog progress (how many dialogs are already finished).

After clicking on the project card, the user will be redirected to the project detail page.

[7] http://www.chatmapper.com/

Use case	**Status**
CRUD projects	Implemented
Import ChatMapper project	Implemented
Download all finalized records in the whole project	To be done
See status of the project	Implemented

Table 3.1: *Uses cases completion overview of the My projects page*

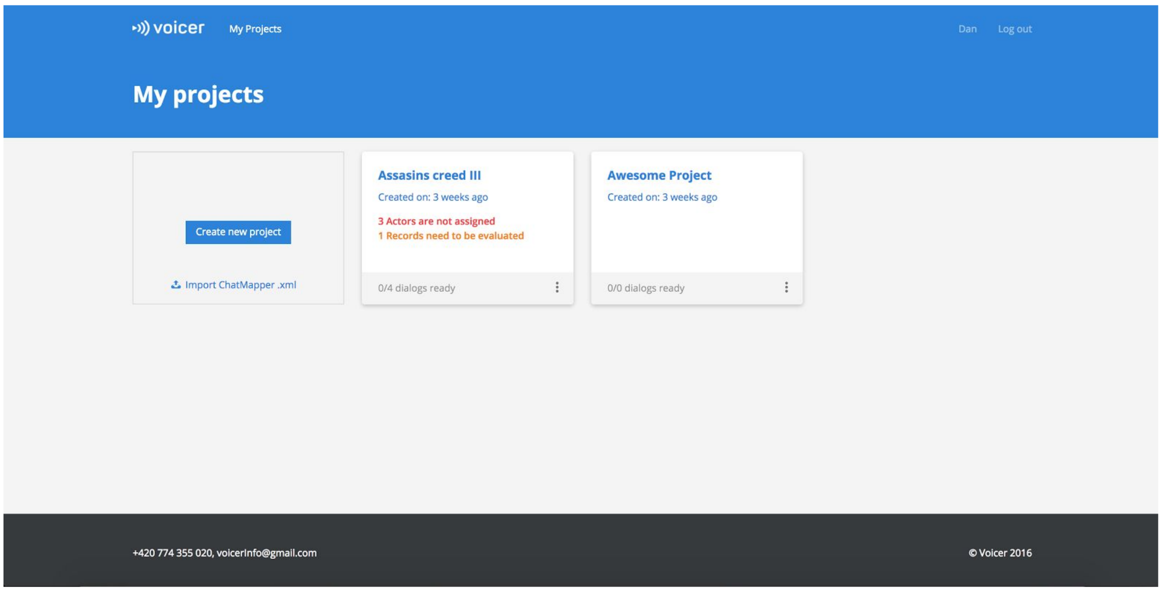


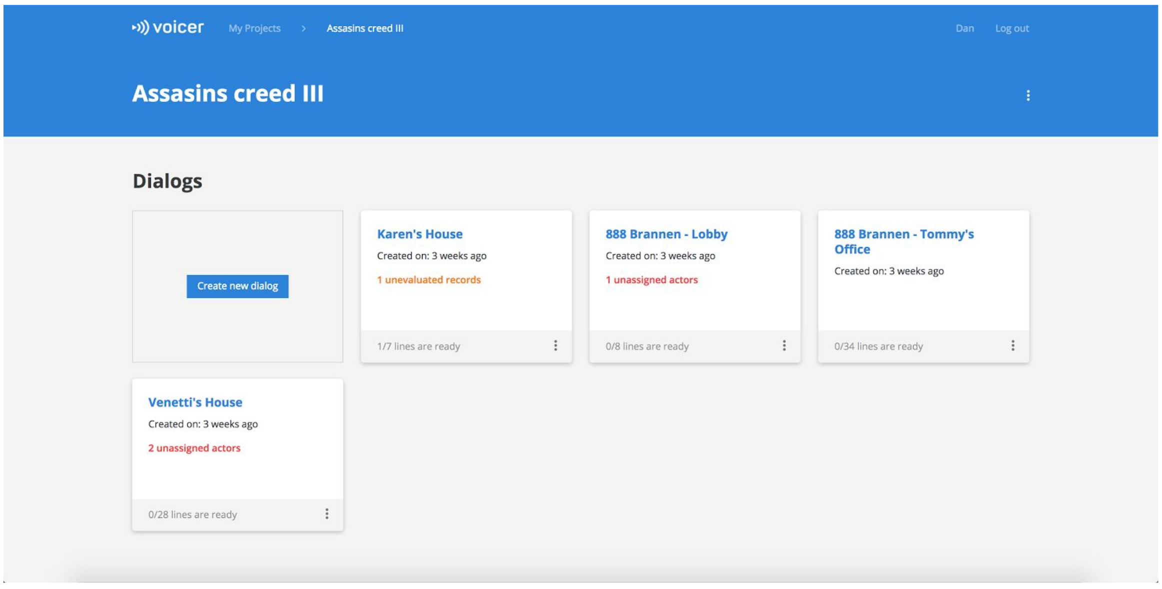
Figure 3.8: My projects page

Projects detail page

This page is shown after the user clicks on a project. It contains the list of all dialogs, that are in the project, as well as information about unevaluated records and unassigned actors. Contextual menu allows to edit the dialog's name, and to delete the currently opened dialog.

Use case	Status
CRUD dialogs	Implemented
Download accepted records	Implemented
See status of the dialog	Implemented

Table 3.2: *Uses cases completion overview of the Projects detail page*



Assasins creed III

Dialogs

Karen's House
Created on: 3 weeks ago
1 unevaluated records
1/7 lines are ready

888 Brannen - Lobby
Created on: 3 weeks ago
1 unassigned actors
0/8 lines are ready

888 Brannen - Tommy's Office
Created on: 3 weeks ago
0/34 lines are ready

Venetti's House
Created on: 3 weeks ago
2 unassigned actors
0/28 lines are ready

Figure 3.9: Project detail page

Dialog page

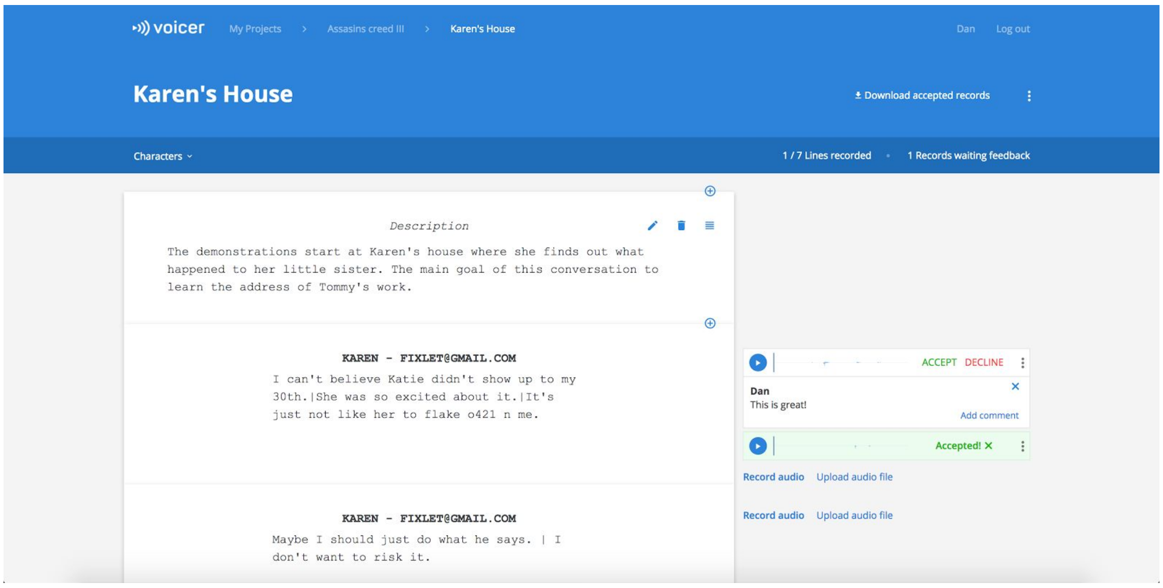
On this page the user is presented with the dialog script. Each line can have an actor assigned, apart from description. Lines can be dragged to rearrange their position. There is also an option to edit, remove and add new line.

Recordings are displayed in the right part of the document. Each recordings can be accepted or declined. There is an option to add a comment to a recording. Lines can be recorded by using a microphone.

Clicking on the characters bar will allow the user to see the overview of the characters and to assign the character a voice actor by entering their email address.

Use case	Status
CRUD lines	Implemented
Assign characters	Implemented
Reorder the plot	Implemented
CRUD recordings	Implemented
CRUD comments	Implemented
Accept/Decline lines	Implemented

Table 3.3: *Uses cases completion overview of the Dialog page*



Karen's House

Description

The demonstrations start at Karen's house where she finds out what happened to her little sister. The main goal of this conversation to learn the address of Tommy's work.

KAREN - FIXLET@GMAIL.COM

I can't believe Katie didn't show up to my 30th.|She was so excited about it.|It's just not like her to flake o421 n me.

Dan
This is great!

KAREN - FIXLET@GMAIL.COM

Maybe I should just do what he says. | I don't want to risk it.

Figure 3.10: Dialog page

4. Testing

This is an overview of the tests, that have been executed to find errors and validate the application.

Static code analysis

This analysis is done without running the application code. It aims to discover bad design patterns and possible bugs. In order to do this type of testing I used FindBugs™ software[8]. After the analysis of the project files, analyser has not found any critical bugs. Other minor bugs have been reviewed and the majority of them was fixed.

Unit tests

Unit testing is a software testing method, which tests individual parts of the application. Individual part of the application is a standalone functional component. It is a White-box testing method, where the internal structure of the component is understood by the testers.

DAO pattern was implemented using generics, thus only one class is responsible for all database-related work. It helps greatly to reduce test cases count and improve test coverage.

```java
public interface DAO<T> {
    public T createObject(T t);

    public T readObjectById(int id, Class<T> tClass);

    public T readObjectByUsername(String username, Class<T> tClass);

    public boolean updateObject(T t);

    public boolean deleteObject(int id, Class<T> tClass);

    public Collection<T> getAllObjects(Class<T> tClass);

    public boolean isObjectInDatabase(int id, Class<T> tClass);
}
```

Figure 4.1: Using of Java Generics in DAO class

[8] http://findbugs.sourceforge.net/

Unit	Coverage
DAO	98%
Models	100%
Services	30%

Table 4.1: Unit test coverage

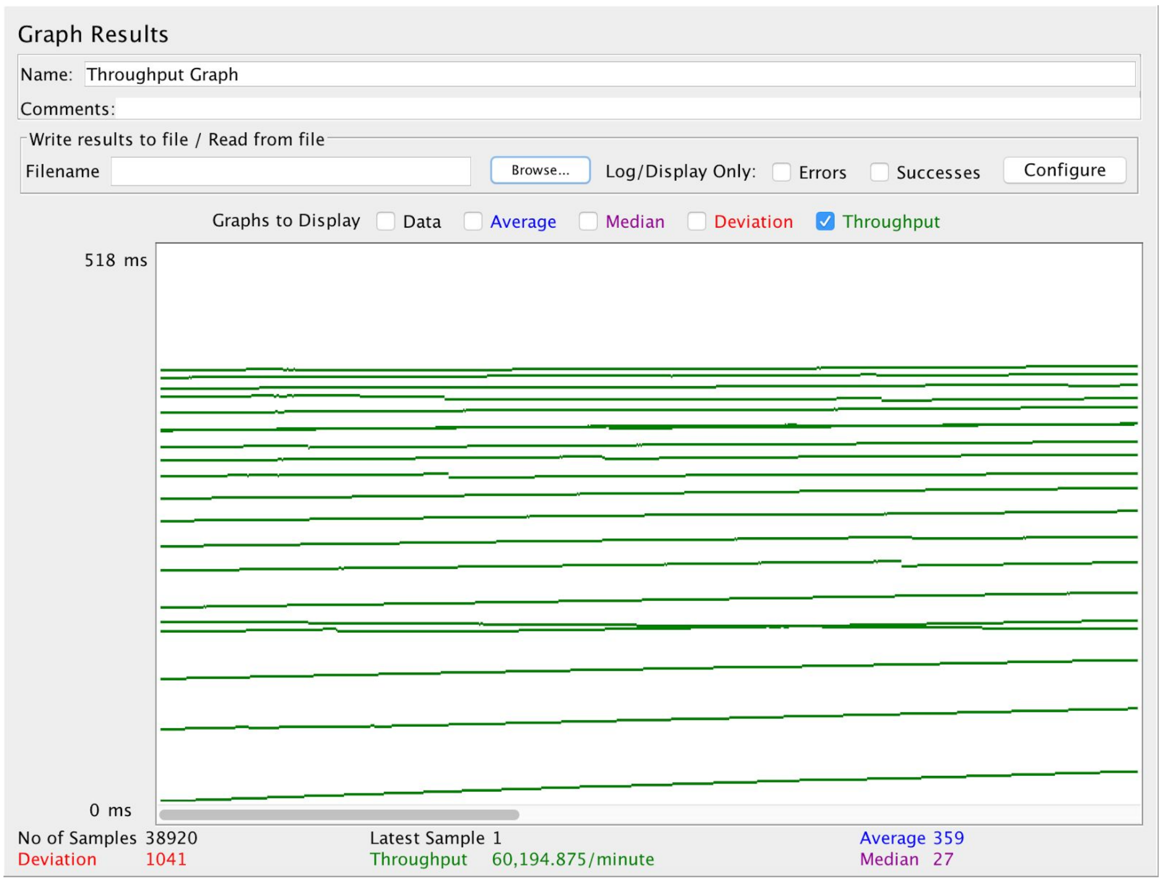
Services should be tested in more detail, as it requires more time to cover all the use cases with tests and to keep the tests updated with the project development pace.

Stress Testing

The purpose of stress testing is to get the insights about the speed of the application when many users are performing their tasks simultaneously. It is required to calculate hardware expenses and see the overall application performance using the specific hardware. It also can provide insights of improvement areas, which are required to be solved in order to optimize the speed and application performance.

Stress tests are yet to be performed on the deployed hardware, as the tests are dependent on the server throughput and repository location. During the local tests, there were not discovered any critical issues and in all cases application has shown performing in constant time. The performance graph is described on the figure 4.1.

More advanced tests are to be done after the application is deployed to a server. Possible problem areas would be use cases related to the downloading audio files. Before the download, the files are compressed to an archive and then can be retrieved by the user request. This can result in slow performance when many users attempt to download the files simultaneously. In this cases a load balancer strategy should be considered, to balance the requests from the users.


Graph Results

Name: Throughput Graph

Comments:

Write results to file / Read from file

Filename

Log/Display Only: Errors Successes

Graphs to Display: Data, Average, Median, Deviation, [✓ Throughput]

No of Samples 38920
Deviation 1041
Latest Sample 1
Throughput 60,194.875/minute
Average 359
Median 27

Figure 4.1: Throughput graph of 100 simultaneous connections using JMeter[9]

[9] http://jmeter.apache.org/

5. Conclusion

5.1. Project goal completion

Based on the results of the developed application, I consider our collaborative project a success.

Goals accomplishment:

- Simplifying audio content management has been accomplished because the developed application provides the tools to manage audio content in one place, where it is easy to see the project's progress and track the changes.
- Providing a project collaboration platform has been accomplished because the application synchronizes the work of the team, and provides an online place for the team to work on the project.
- Creating a single storage for the whole project goal has been accomplished because all the data is securely stored on the server side at one place, and it is easy to retrieve and backup it.

My contribution to the project was the design and implementation of the back-end of the application, in particular:

- Translating the requirements into a feasible solution.
- Choosing the right technology for the project’s implementation.
- Building the application’s architecture.
- Capturing real-world entities with the database physical model.
- Testing and preparing for the application deployment.

5.2. Further work

Current state of the application is prepared for the deployment to a server with limited public access. It is ready for the internal alpha release. As the application grows, it gets harder to test everything, thus, more testing should be done to validate that all the use cases are covered and the application works stable all the time.

Marketing materials should be prepared to describe the benefits of the tool and to attract new customers. After that, it is needed to contact users from the target audience and present the tool. It would be great to attend a gaming conference and have a speech about the application there.

In addition, monetization plan should be prepared. If the application gets the users, that will increase the costs connected with server maintenance and storage expansion. Different monetization approaches should be analyzed and decided on the strategy.

Finally, more usability testings should be done with the users. We would like to gather all the responses and improve the tool based on the users’ feedback.

6. References and literature

[1] "List of video game developers - Wikipedia." Cited on May 6, 2017. https://en.wikipedia.org/wiki/List_of_video_game_developers.

[2] "List of animation studios - Wikipedia." Cited on May 6, 2017. https://en.wikipedia.org/wiki/List_of_animation_studios.

[3] "Kanban The Agile Coach - Atlassian." https://www.atlassian.com/agile/kanban. Cited on 7 May. 2017.

[4] "Hibernate ORM - Hibernate ORM." 2013. Cited on 4 Jan. 2016 <http://hibernate.org/orm/>

[5] "22. Web MVC framework - Spring.". Cited on May 6, 2017. https://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html.

[6] "Spring Security - Projects.". Cited on May 6, 2017. https://projects.spring.io/spring-security/.

[7] Provos, Niels; Mazières, David; Talan Jason Sutton 2012 (1999). "http://www.usenix.org/events/usenix99/provos/provos_html/node1.html". Proceedings of 1999 USENIX Annual Technical Conference: 81–92.

[8] "Maven – POM Reference." 2010. Cited on 20 Jan. 2016 https://maven.apache.org/pom.html

[9] "Apache Tomcat® - Welcome!.". Cited on May 6, 2017. http://tomcat.apache.org/download-80.cgi.

[10] "MySQL.". Cited on May 6, 2017. https://www.mysql.com/.

[11] Sergey Krasotin — Bachelor thesis. Internal archive of diploma works, CTU in Prague, 2016

7. Appendix A. Abbreviations

DAO — Data access object design pattern

MVC — Model View Controller design pattern

WAR — Web archive

JSP — Java Server Pages

URL — Unified resource locator

API — Application programming Interface

ORM — Object-relational mapping

POM — Project object model

RDBMS — Relational database management system

CRUD — Create, Read, Update, Delete

8. Appendix B. CD content

CD contains:

- Thesis document in .doc format
- Thesis document in .pdf format
- War artifact of the application
- Zip archive with sources