System Description

For

Ebooks

Document Version: [0.1]

Date: [11/04/14]

TABLE OF CONTENTS

1. OVERVIEW

1.1. Service description

Ebooks is a web application for storing, viewing and editing books in FictionBook format and contains elements of social interaction.

Service provides the following functionality

For unauthorized users:

- Viewing page "Recent books».
- Full-text search and search by metadata.
- Online reading books.
- Free registration and authorization in the system.

For unauthorized users:

- All described possibilities of unauthorized user.
- Uploading book files in fb2 format (also mobi and epub with conversion).
- Viewing page "My books" that displays all uploaded books.
- Editing access type for a book (private, public).
- Sharing books to another user.
- Editing metadata for uploaded books.
- Deleting book.
- Editing profile.

1.2. Data Model

Figure 1—Data Model

1.3. Technology

Basic information

Software platform	Java platform
Programming language	Java SE 7
Application server	JBoss 7.1
Database management system	MongoDB 2.4.9

Back-end frameworks and libraries

Context framework	Spring Dependency Injection 4.0.1
Server-Side UI framework	Spring MVC 4.0.1 & Apache Tiles 3.0.3
Persistence framework	Spring Data 4.0.1
Security framework	Spring Security 3.2.0
Information retrieval library	Apache Lucene 4.6.0
Testing	JUnit 4.8.2 + Mockito 1.9.5
Logging	Log4j 1.2.17

Front-end frameworks and libraries

Web design	HTML5 + CSS3

JavaScript library	JQuery
Fundamental framework	Twitter Bootstrap 3.1.1

1.1. Development Tools

IDE	IntelliJ IDEA 13
Build automation tool	Apache Maven 3.2.1
Revision control system	Git 1.9.0

1.2. Source Control

Ebooks uses Git as a revision control system and GitHub as a hosting. The full code can be seen at https://github.com/riaval/EBooks/tree/mongodb.

1.3. User Documentation

2. TECHNICAL ARCHITECTURE

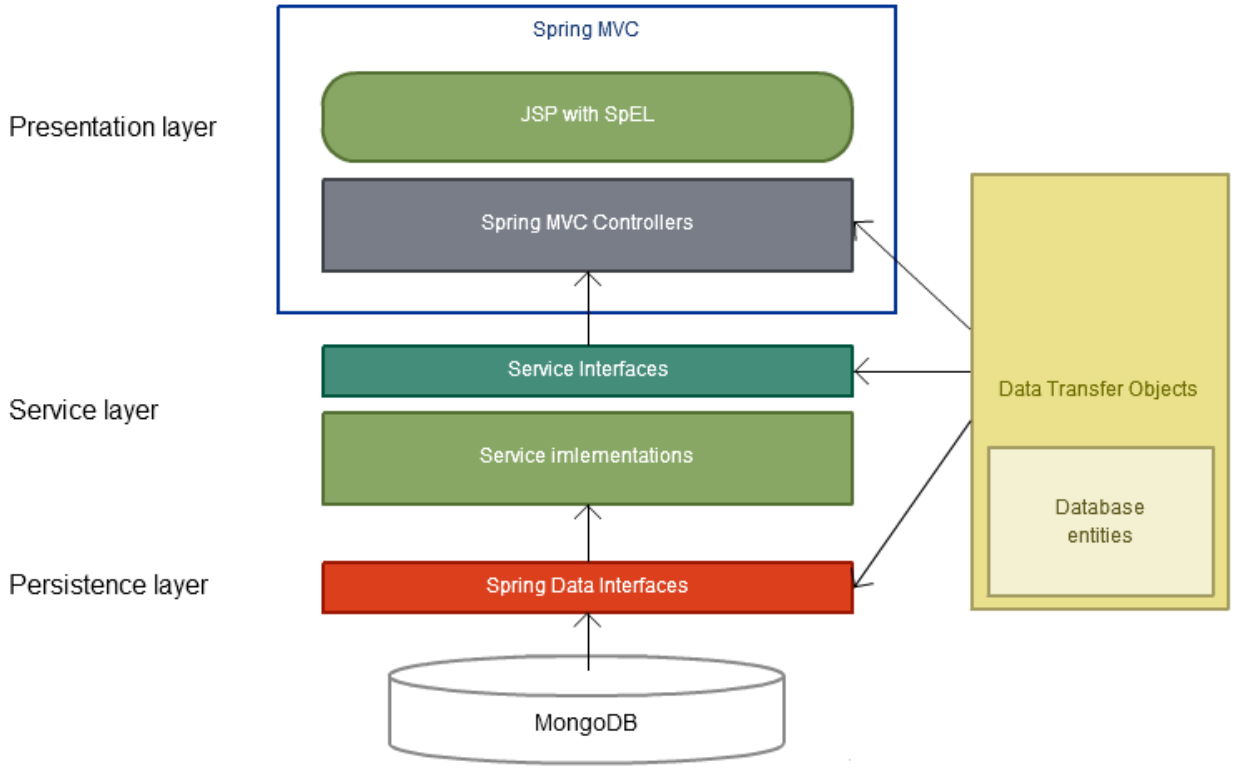

Figure 2—Basic system architecture

Information system is based on pretty standard architecture. Basic components structure divided into three layers, which is independent and replaceable.

System layers encapsulate operations (methods) which operate exclusively with the following:

- Primitive types;
- Implementation of classes that defined in JavaSE;
- Data Transfer Objects.

2.1. **Persistence layer** (Data access layer).

Part of the System that provides connection between DBMS and software.

It is written with Spring Data framework and theoretically able to support any DBMS including RDBMS.

Located in `ua.miratch.zhukov.repository` package and contains the following interfaces:

- `BookRepository.java`
- `UserRepository.java`

2.2. Service layer.

The largest part of the System that contains:

- All business logic;
- Exception handling;
- Logging in basic appenders;

Located in `ua.miratch.zhukov.service` package and contains the following interfaces:

- `BookIndexerService.java` — for indexing and searching.
- `BookService.java` — provides operations with Book entities.
- `ConverterService.java` — provides conversion from .mobi or .epub into FictionBook (.fb2) format. Uses free REST web-service on http://online-convert.com
- `FileService.java` — allows to save, read and delete files in File Storage.
- `UserService.java` — conducts CRUD operations with users in System.
- In addition to this, service layer include some technical components:
- `EbookStorage.java` — component that determines absolute path to File Storage, based on value in system path variable.
- `MongoUserDetailService.java` — component with configuration for Spring Security.

2.3. Presentation layer.

Part of system that that is responsible for User interaction. Based on Spring MVC framework and include two big part — Controllers and JSP pages.

2.3.1. Controllers

Located in `ua.miratch.zhukov.controller` package and contains the following classes:

- `BookController.java`
- `EditBookController.java`
- `ExceptionController.java`
- `MainController.java`
- `MyBooksController.java`
- `SearchController.java`
- `SecurityController.java`
- `UploadController.java`
- `UserController.java`

2.3.2. JSP pages

Located in `webapp/WEB-INF/pages` directory and contains the following pages:

- `editBook.jsp` – `/book/{bookID}/edit`. Page for editing books.
- `index.jsp` – `/`. Main page.
- `login.jsp` – `/login`. Login page.
- `myBooks.jsp` – `/mybooks`. Page with the books of the current user.
- `profile.jsp` – `/profile`. Profile page.
- `readBook.jsp` – `/book/{bookID}`. Page for reading.
- `search.jsp` – `/search`. Search page.
- `signup.jsp` – `/sugnup`. Sign Up page.
- `upload.jsp` – `/upload`. Uploading page.

The project uses Apache Tiles framework as a template engine. Settings file (tiles-templates.xml) can be found in `webapp/WEB-INF/ templates` directory. In the same directory, you may found templates and pages for import:

- `template.jsp` — basic template (header, footer, etc.).
- `book-include.jsp` — book template.
- `pagination-include.jsp` — pagination template.
- `users-modal-include.jsp` — template for modal window.

3. HOW-TO GUIDE