**Medical Journal Publication System**

**Design document**

The purpose of this document is to outline the high-level design specification of the Medical Journal system in which it presents the application objective, high-level requirement analysis and assumption, technology of choice, data model, architecture diagrams describing process, control, and data flows, and implementation overview of sub-system.

The digital Medical Journal Publication system allows publishers to share medical related journals to any public user (subscriber) who is interested to specific subject by subscribed and read it, electronically.

**Functional Requirements**

The system allows publishing and subscribing to medical journals in a secure way. The system **must** implement the following specification:

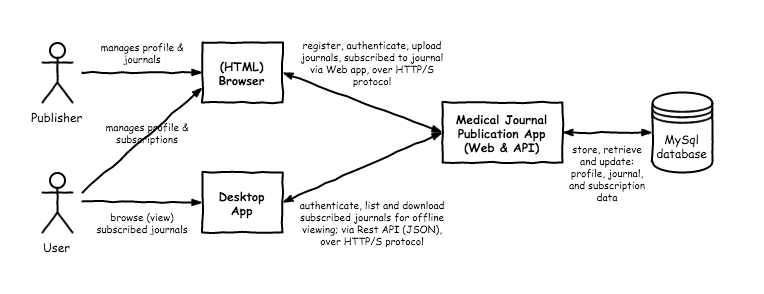
1. For publishers
   1. A web portal to upload and manage a list of medical journals
   2. To upload journals in PDF format
2. For public users
   1. A web portal to find and subscribe to journals of their interest
   2. A desktop client;
      1. To list and read the subscribed journals
      2. That does not allow copying the journal’s content in any common way
         1. Not from web requests/response
         2. Not from their file system
         3. Not from the client use interface (copy to clipboard, screenshot, etc.)

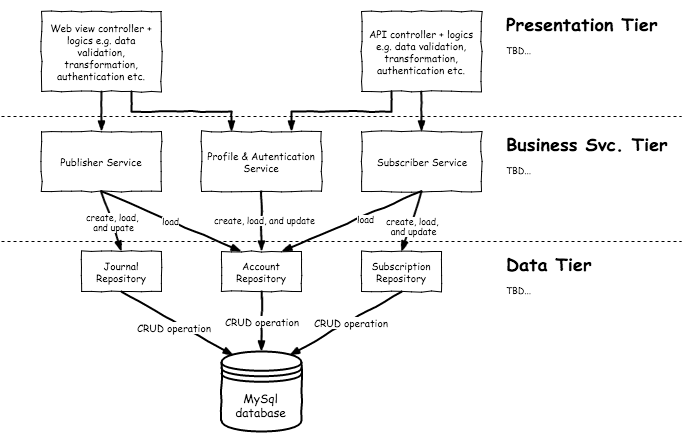
Assumptions

The following assumptions is established to allow a simple, efficient, and secure implementation of the system:

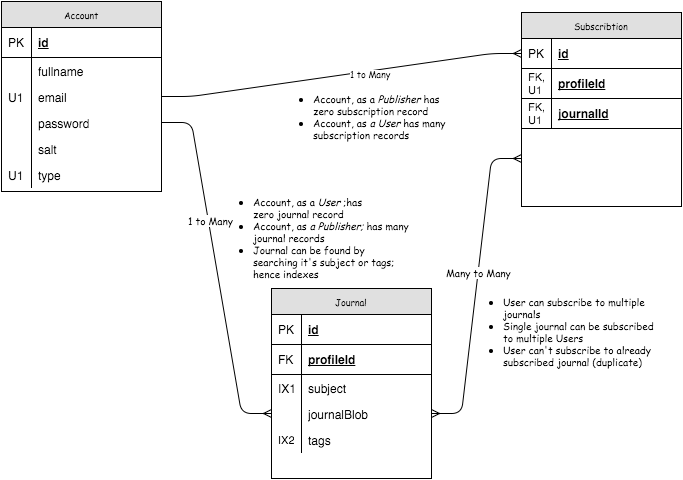
1. User can register and/or access the system as either a *Publisher* and *Public user* role or ***both***, therefore: a user must specify their role, during register and login to Web portal
2. A journal, in addition to pdf document, should consists short notes describing context & topic and tags for easier search
3. Both *Publishers* and *Public Users* must be authenticated before using Web portal & Desktop client:
   1. To identify users in managing medical journals and subscription
   2. Securing access to web portal and protected data
   3. Fetch correct list of subscribed journal
4. Only author (publisher) of a journal can manage (list, read, update, delete) thru Web portal, therefore: copy protection feature does not apply to Web portal
5. Desktop client **allows** *Public user* read subscribed journal *offline*, and to avoid user from **copying the journal’s content** it must in encrypted form both in transit and at rest
6. User can’t have duplicate journal

**Architecture**





**Data Model**



**Implementation**

Security

TBD… explain how high-level security

Use-Cases

|  |  |  |  |
| --- | --- | --- | --- |
| Actor | Operation | Test Steps | Unit Tested |
| Publisher |  |  |  |
|  | Registration | Happy path:   1. Navigate to [host]/index 2. Registration form is rendered 3. Enter: full name (optional); email, password, and type (all mandatory) 4. New publisher is created, user redirect to [host]/journals | Tested |
|  |  | Non-happy path   1. ~~Invalid email and/or password format~~ 2. Duplicate email | Tested |
|  | Login | Happy path:   1. Navigate to [host]/index 2. Login form is rendered 3. Enter: email, password, and type; all mandatory 4. Upon successful login, user redirect to [host]/dashboard/journals | Tested |
|  |  | Non-happy path:   * 1. Invalid email and password |  |
|  | Reset Password |  |  |
|  |  |  |  |
|  | List Journals | Happy path:   * 1. Navigate to [host]/journals   2. Dashboard page is rendered, displaying list of journal   3. Journal’s item: Subject, file name, and link to preview journal |  |
|  | Upload Journals | Happy path:   * 1. Navigate to [host]/dashboard/publisher   2. Dashboard path is rendered, displaying list of journal   3. User click “Upload” button, user redirect to [host]/journal/upload   4. Enter: subject, tags and select local file (pdf type only) – all mandatory   5. User click “Submit   6. New Journal is added, user redirect back to [host]/dashboard/publisher |  |
|  |  | Non-happy path:   * 1. Missing mandatory fields   2. File is not in pdf format   3. Upload incomplete (connection severe or timeout) |  |
|  |  |  |  |
| Public User (Subscriber) |  |  |  |
|  | Registration | Happy path:   1. Navigate to [host]/register 2. Registration form is rendered 3. Enter: full name (optional); email, password, and type (all mandatory) 4. New publisher is created, user redirect to [host]/subscriber |  |
|  |  | Non-happy path   1. Invalid email and/or password format 2. Duplicate email |  |
|  | Login | Happy path:   1. Navigate to [host]/login 2. Login form is rendered 3. Enter: email, password, and type; all mandatory 4. Upon successful login, user redirect to [host]/publisher |  |
|  |  | Non-happy path:   1. Invalid email and password |  |
|  | Reset Password |  |  |
|  |  |  |  |
|  | List of subscribed journals | Happy path:   1. Navigate to [host]/subscriber 2. Dashboard page is rendered, displaying list of subscribed journals 3. subscribed journal’s item: Subject |  |
|  | Find Journal, by subject ***~~or tags~~*** | Happy path:   1. Navigate to [host]/subscriber 2. Dashboard page is rendered, displaying list of subscribed journals 3. User enter “Search topic” (subject and/or tags), click search 4. Redirect to [host]/journal?search= 5. Journal page is rendered with list of related journals, per search text |  |
|  |  |  |  |
|  | Add Journal | Happy path:   1. Perform “Find journal” 2. User checked one or many journal item, click “Subscribe” 3. User is now subscribed to multiple journals 4. Redirect to [host]/subscriber |  |
|  |  | Non-happy path:   1. No journal item is selected, when click “subscribe” |  |
|  |  |  |  |
|  | Logout |  |  |
|  |  |  |  |
|  | | | |
| Desktop User (Subscriber) |  |  |  |
|  | Login | Happy path:   * 1. Launch Desktop app   2. Login form is rendered   3. Enter: email, password; both mandatory   4. User navigate to dashboard | Tested |
|  |  | Non-happy path:   1. Invalid email and password 2. Connection is unavailable (offline) or timeout |  |
|  | Reset Password |  |  |
|  |  |  |  |
|  | List of subscribed journals | Happy path:   1. Navigate Dashboard 2. App fetches list of journal via API 3. Upon completion, Dashboard page renders list of subscribed journals 4. subscribed journal’s item: Subject and file name |  |
|  |  | Non-happy path:   1. Session is invalidated 2. Connection timeout |  |
|  | Read journal | Happy path:   1. Navigate Dashboard 2. App fetches list of journal via API 3. Upon completion, Dashboard page renders list of subscribed journals 4. subscribed journal’s item: Subject and file name 5. User click “Read” button 6. App download selected journal and rendered pdf file. |  |
|  |  | Non-happy path:   1. Session is invalidated 2. Connection timeout |  |
|  | Copy protect |  |  |

**Technology of Choice**

The application, both Web portal and Desktop client will be build using Java technology along with proven open source technology.

* Spring Boot 1.3.3. Web, Security, JPA, MySQL, Thymeleaf & H2
* JavaFX (Desktop app)
* ~~Java JCE,~~ *~~(encryption)~~*
* Java 1.8
* Maven 3.0+
* Tomcat 8.0
* ~~MySQL~~
* Eclipse IDE