**Java Fresher Program**

**Project Assignment**

# Table of Contents

**Prepared by DXC Technology Vietnam**

[Table of Contents i](#_Toc504143805)

[1 Project 1: Library Management System (LMS) 1](#_Toc504143806)

[1.1 Description 1](#_Toc504143807)

[1.2 Functional Specification 1](#_Toc504143808)

[2 Project 2: Patients Profile Management (PPM) 2](#_Toc504143809)

[2.1 Description 2](#_Toc504143810)

[2.2 Functional Specification 2](#_Toc504143811)

[3 Project 3: E-Invoice System 3](#_Toc504143812)

[3.1 Description 3](#_Toc504143813)

[3.2 Functional Specification 3](#_Toc504143814)

[4 Project 4: Mini File Sharing (MFS) 4](#_Toc504143815)

[4.1 Description 4](#_Toc504143816)

[4.2 Functional Specification 4](#_Toc504143817)

[5 Technical Requirements 5](#_Toc504143818)

[6 Amendment Record 6](#_Toc504143819)

# Project 1: Library Management System (LMS)

## Description

We will propose a web-based system to help a library in their management operations. In general, each book borrower will have to register his/her account in order to borrow book from the system. LMS will manage all books through their ISBN. There’s a limitation number of books which are allowed a user to borrow at the same time, this number could be changed by admin.

## Functional Specification

* Borrower functions:
  + Register account
  + Browse/Search list of books with information:
    - Title of book
    - Author
    - Publishing Year
    - ISBN number
    - Short description
    - Total number of borrowing tickets
    - Status: Available or not?
    - Open ticket to borrow books
* Admin functions:
  + o Add a book (with ISBN) to the store
  + o Update/Remove books
  + o Browse/Search book list (student should think about the details)
  + o Change the “limitation number of books” in each ticket.
  + o Reports:
    - Statistics for number of borrowing tickets weekly/monthly/yearly
    - Statistics of weekly/monthly/yearly income
* System constraints:
  + Validate against the limitation number of books when user open a ticket.
  + Check availability of books (based on time of borrowing in a ticket) to ensure borrowing time of each book is not overlapped with other tickets
  + ISBN validation

# Project 2: Patients Profile Management (PPM)

## Description

We will build a system which administrates patients profile in a hospital. Each profile will be divided into 2 parts:

* Personal Info: Full name, Address, DOB, Sex, Place of Birth …
* Medical treatment profile: History of diseases, prescriptions, medicines being used or recently used, medical test results...

## Functional Specification

* Nurse functions:
  + Initial input patient profile.
  + Only able to update personal info in a patient profile.
  + Search patient profile (student should think in detail).
* Doctor functions:
  + Search and pickup patient profile: Filter by name, diseases, medicine.
  + Create/Update a medical treatment profile in a patient profile: Basic information, uploading medical test results (scan or copy of prescriptions, X-Ray pictures…)
  + Check whether a patient be allergic to a specific medicine.
* Admin functions:
  + CRUD accounts for nurse and doctor.
  + Medicine list management (CRUD only)
  + CRUD on patient profiles.
* System constrains:
  + Each patient profile could have one or many medical treatment profiles.
  + Medical test result files size is limited to 10 MB.
  + Tracking changes history in medical treatment profile: Which doctor makes change, Date and time of each change.

# Project 3: E-Invoice System

## Description

E-Invoice is built for the purpose of monthly invoice management in a family. It helps us to have an overall report of monthly expenses so that user could make plan to save money. User could set an alert for a limited amount of money consumed every month. Each invoice will have following information:

* Type of invoice: Electric, internet, telephone, water, ...
* Amount of money.
* VAT
* Charged period (monthly).
* Invoice No.
* Customer code (optional).

## Functional Specification

* Users:
  + Register account.
  + CRUD services.
  + CRUD invoice consumed by a service.
  + View expenses report: Monthly, Yearly, or a period of time.
  + View charts: Bar chart, Pie chart (students should think about the details).
  + Set the monthly limited expenses for notification (via email).
* Admin:
  + Activate/Deactivate a user account.
  + Configure the trigger point of time for running scheduled job to send email for monthly notification to each user if their monthly expenses exceed the pre-set limited amount of money.
* System constrains:
  + Check duplication of monthly invoice when adding an invoice.
  + Able to send email for monthly notification to every user at trigger time (using POP3).

# Project 4: Mini File Sharing (MFS)

## Description

MFS is a web application which allows user to share their file to the other over the internet. Users are required to register and login if they want to upload or download any file from MFS. Guest user could browse file by category, size or uploader but cannot download any file from MFS.

## Functional Specification

* Guest (not login):
  + Search file by name, category, uploader, size
  + View information of an uploaded file (with comment).
  + Register new account.
* Registered user:
  + Search file by name, category, uploader, size
  + View information of an uploaded file (with comment).
  + Upload file with a limitation of file size according to account level
  + Download a file from MFS
  + CRUD files which have been uploaded by the his/her self
  + Comment on an uploaded file.
* Admin:
  + CRUD users
  + View information of each user from user dashboard
  + Upgrade/downgrade level of a user
* System constrains
  + Checking uploading every level of user
    - Bronze: maximum 5MB each upload
    - Silver: maximum 10MB each upload
    - Gold: maximum 20MB each upload
  + Auto upgrade user level following total size of uploaded files:
    - Bronze: 20MB
    - Silver: 50MB
    - Gold: above 100MB
  + Limit on daily downloading quota:
    - Bronze: Below 50 MB
    - Silver: Below 70MB
    - Gold: Unlimited

# Technical Requirements

* Framework: Spring MVC, Hibernate. (Bonus if: Spring Security, Applying Spring boot)
* Client side: AngularJS, Boostrap, HTML5/CSS3
* Database system: MySQL or MongoDB
* All CRUD operation should have JUnit tests.
* Sample data should be prepared before presentation
* Pagination on returned result in search function.
* Source control: GitHub

# Amendment Record

**Versions are incremented to next whole number when there is a major change. Minor changes are incremented to the next .x of the Version (i.e., 1.1 vs 2.0)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Version** | **Nature of Change** | **Changed by** | **Reviewed by** | **Approved by** |
| mm/dd/yy | 1.0 | Initial Release |  |  |  |
| 01/18/18 | 1.1 | Update with DXC Template | Anh Mai | Anh Mai | Quang Tran |
|  | 2.0 |  |  |  |  |