<Library Management System>

Architecture Notebook

There is guidance within this template that appears in a style named InfoBlue. This style has a hidden font attribute that allows you to toggle whether it is visible or hidden in this template. Use the Microsoft® Word® menu **Tools > Options > View > Hidden Text** check box to toggle this setting. There is also an option for printing: **Tools > Options > Print**.

# Purpose

The library management system has been shaped by system requirements, constraints and design decisions. This document explains how the system is organized and how it works with its components.

# Architectural goals and philosophy

The Library Management System will be written in java. In this case it can be run by any system that supports the JVM. If the operating system is new or old, it will not affect the performance of the system or if the operating system is Windows or Linux, it will not affect the performance of the system. But the Library Management System will only be run on the computer. It is not compatible with the phone or tablet.

The Library Management System is designed to be robust for long term maintenance.

The Library Management System is designed to run independently from the internet. Also it does not need any hardware dependencies.

# Assumptions and dependencies

* Java has its own working environment that’s name is JVM. So, the Library Management System is designed to be developed in Java to run any environment.
* The Library Management System is intended to be used by wide users. So, the interfaces of system are designed simple.
* The Library Management System is expected to adapt to new technologies. So, it is designed to be developed.

# Architecturally significant requirements

The following technologies are used and are necessary when the system is developed:

* <https://github.com/Angelcrest/bbm487s2017g3>
* <https://marketplace.eclipse.org/>
* <http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-jvm-419420.html>

# Decisions, constraints, and justifications

* The Library Management System does not use internet. The system needs to be installed on the computer.
* Members, librarians and books are registered in the array list temporarily.
* Admin must be registered in the LMS.
* Admin can add librarians to the LMS or delete them from the LMS or search them in the system.
* Admin cannot add member to the LMS and delete them from the LMS.
* Admin cannot update member or librarians in the system.
* Users must be a member of the system.
* Librarians cannot add themselves to the LMS or delete themselves from the LMS.
* Librarians can add books to the system, delete them from the system or update them in the system.
* Librarians can borrow books from system or return them back to the system.
* Librarian can search books or members.
* Librarians can add members to the system delete them from the system or update them in the system.
* Members cannot add themselves to the system or delete themselves from the system or update themselves in the system.
* Members cannot add books to the system, delete them from the system or update them in the system.
* Members can search books from system.
* Members can borrow books from system or return them back to the system.
* Members can reserve books in the system.
* Members can check books out with barcode from system or check books in with barcode to the system.

# Architectural Mechanisms

* ***Archiving***: provides a means to move data from active storage when user is librarian or admin.
* ***Auditing***: provides audit trails of system execution. Checks whether the user is registered in the system or the record of the book in the system.
* ***Debugging***: Provides items that support debugging while the application is being developed.
* ***Disaster Recovery***: Provides facilities to recover systems, application and data. This mechanism will be added when the database is included in the system.
* ***Error Management***: Allows errors to be detected and notification to the screen.
* ***Event Management:*** A user cannot perform multiple synchronous events.
* ***Graphics:*** The system provides interface support for the user
* ***Information Exchange***: Supports changing and updating information through the system interface
* ***Memory Management:*** Support adding data to the memory and data deletion from the memory
* ***Persistence:*** It provides services related to reading, searching and writing stored data.
* ***Printing:*** Supports the display of searched books and members in the interface
* ***Resource Management:*** Provides support for the management of database connections. This mechanism will be added when the database is included in the system.
* ***Transaction Management:*** Support handling ACID transactions. This mechanism will be added when the database is included in the system.

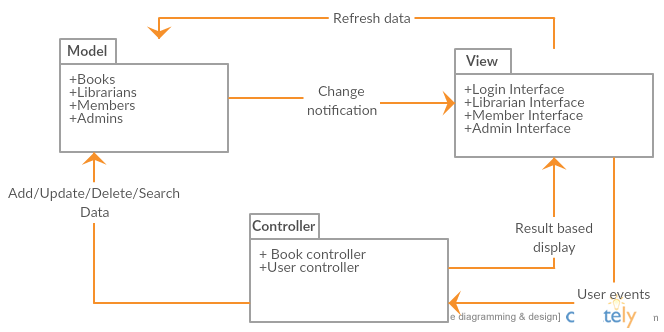
# Key abstractions

* ***Member and their own account:*** Each member has their own id, name, surname, e-mail, username and password. Their accounts include these attributes.
* ***Librarian and their own account***: Each librarian has their own id, name, surname, e-mail, phone number, username and password. Their accounts include these attributes.
* ***Books and their own record:*** Each book has their own id, name, author and year. Their records include these attributes.

# Layers or architectural framework

The system's layers are represented by the model-view-controller pattern.

* ***Model***: show stored data. Books, librarians, members and admin are stored like data in the library management system. This data is accessed and can be added, deleted, changed, etc.
* ***View***: generates an interface to the user based on changes in the model. Firstly the users have to login in the system. Then the user will be redirected to own interfaces.
* ***Controller***: connection between view and model. Controller can be user controller or book controller.



# Architectural views

[Describe the architectural views that you will use to describe the software architecture. This illustrates the different perspectives that you will make available to review and to document architectural decisions.]

## Recommended views

* **Logical:** Describes the structure and behavior of architecturally significant portions of the system. This might include the package structure, critical interfaces, important classes and subsystems, and the relationships between these elements. It also includes physical and logical views of persistent data, if persistence will be built into the system. This is a documented subset of the design.
* **Operational:** Describes the physical nodes of the system and the processes, threads, and components that run on those physical nodes. This view isn’t necessary if the system runs in a single process and thread.
* **Use case:** A list or diagram of the use cases that contain architecturally significant requirements.