**NextGen Library System**

**Introduction**

NextGen is a Standalone windows application using the C# language. This application allows a user to search for and check out books.

The application features an interactive graphical UI, an efficient search engine and uses authorization for logins. There are 2 types of users: regular and admin. A regular user can search and check-out books. An admin will be able to manage the books and the checkout system.

**Architecture**

Model View Controller (MVC) is used to ensure that the subsystems can be tested in an individual manner. This will also help in code reviews, readability and simplicity of design.

1. Model : Utilities, Services, LibraryItems, Users, Managers
2. View: Home page, signup page, login page, contact page, admin tools.
3. Controller: UserManager, FeeManager, BookManager

**Requirements** (see deliverable1.2\_Requirements\_and\_Qualiy\_Plan)

1. User Registration: Users and Admin (aka librarian) user types are implemented and stored in a backend DB.
2. Book Search: enables users to search for titles, authors, and categories.
3. Placing Hold on Book: A hold is placed on a book
4. Checking out Book: A book is available if it is not checked out or on hold.
5. Renewing a Book: A re-issue process will enable this process by first checking for fines.
6. Adding New Book to Collection: Librarians can add new books to the library’s collection. The data of author, category, title, and ISBN can be added.
7. User Browse Book Collection: A user can browse a list of books. This can be done by finding a category to search for from a drop down list.
8. Inactivating User: A user can be deactivated by the librarian and cannot borrow any books.

**Design Goals**

The design goals are to use flexible, reusable and modular subsystems that reduce defects. These are achieved by implementing several non-functional requirements.

1. Reliability: a major goal of the system to provide 99% uptime for the user. To achieve this, the system provides proper error handling and logging for maintenance activities. This enables the application to prevent and log errors before the user experiences them.
2. Performance: the goal is to expect a high rate of responsiveness.
3. Usability:The system uses the think-out-loud method. This method offers an efficient, flexible and robust set of operations to get instant feedback on the performance. Users would comment in areas such as problems, improvements and good UI experiences.
4. Efficiency: Timing will be used to analyze the search functions which should be completed in < 5 seconds. Also, it will return fewer than 5% inaccurate results.
5. Portability: the application will use standard libraries when possible.

**Testability**

The application contains unit tests for the authentication service and the search engine. Unit tests were tested to ensure that the subsystems as well as their sub components were working properly. With tests we can properly see if the functions are working or not.

**Program source code organization**

The code is organized by placing code in logical containers

1. DataSource
2. Images
3. Interfaces
4. Models
5. NextGenManager
6. Pages
7. Resources
8. Services

**Logging**

The application uses logging targets of console, file and email to share information about the state of the application

**Security**

The system uses authentication (login, password) to allow only registered users to use the application. It keeps track of how many failed attempts by a specific user exist.

**Quality Goals**

1. The system user interface should be user-friendly and a new user should be able to understand how to use it the first time.

2. System should protect the integrity of the data.

3. The system should efficiently and effectively display a search result after the user submits a search request.

4. The system should provide accurate search results

5. Only authorized users (librarians) have access to modify catalog and patron data.

6. The system must be very reliable.

7. The system must maintain fault tolerance

**Quality Metrics**

1. A typical user should be able to know how to use the system in 30 minutes of introduction.

2. There should be no duplicate users. User data should be stored in a format to keep it safe from alteration and arbitrary reading.

3. The system should be able to return a search result in less than 5 seconds.

4. The system should display the research result with 98% accuracy with valid search input.

5. The system should authenticate 2 types of users and should be able to differentiate between the user types utilizing the system.

6. Users must be able to access the system and perform common tasks 98% of the time without running into any issues. The system must track the overall error count to make sure it is reliable.

7.Errors should be handled properly without forcing the application to terminate.

**Installation**

1. Clone or download main repository

2. Navigate the folder for the NextGenLibraryProgram zipped folder and extract it to your location of choice

3. Run the Executable named “NextGenLibrary”

**Configuration**

No user level configuration. However, a developer with access to the JSON file of “UserAccounts” and “books” may edit books or user accounts programmatically.

**Estimated cost of targeted quality -** The project usesproactive planning and managing as compared to a reactive approach. This will help to identify issues more effectively.