****

**BN002/013/104 Year 2 Software Engineering and Testing**

**Assessment 2: Requirements Document**

**Submitted by:**

**Nikolaos Vourdanos (B00093003)**

**Joseph Tierney (B00092923)**

**Eoghan de Bhal (B00092942)**

**Submission date**

**24/02/2017**

**Declaration**

I herby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ordinary Degree in Computing in the Institute of Technology Blanchardstown, is entirely my own work except where otherwise stated.

Author: Nikolaos Vourdanos Dated: 24/02/17

Author: Joseph Tierney Dated: 24/02/17

Author: Eoghan de Bhal Dated: 24/02/17

**Table of Contents**

[Title: Primary School Library 4](#_Toc475728591)

[Client: Our Lady of Victories 4](#_Toc475728592)

[1. Project Overview 4](#_Toc475728593)

[2. Document Revision 4](#_Toc475728594)

[3. Scope 5](#_Toc475728595)

[5. Software Requirements Analysis: 6](#_Toc475728596)

[5.1 Non-functional Requirements: 7](#_Toc475728597)

[5.2 Use Case Diagram 8](#_Toc475728598)

[5.3 Class Diagram 9](#_Toc475728599)

[5.4 Admin Sequence Diagram 9](#_Toc475728601)

[5.5 User Sequence Diagram 10](#_Toc475728602)

[6. Graphical User Interface Design 11](#_Toc475728603)

[7. Technical Requirements and Feasibility: 12](#_Toc475728604)

[8. Conclusion 12](#_Toc475728605)

[Appendix A - Project Diary 13](#_Toc475728606)

# Title: Primary School Library

# Client: Our Lady of Victories

# Project Overview

Our Lady of Victories (OLV) Primary School holds a “Books for Tots” book lending scheme which is run by a local parent of a child from the school. The parent currently uses a paper based recording system for the books, which OLV roughly receives 40 books. The parent keeps track of which student has taken a book out on loan, when the book was returned, if the student has taken out another book and the current condition. This project proposes that a Graphical User Interface (GUI) be used in replacement of this time consuming and redundant recording system. This will allow the parent to use a database system to keep track of everything that they need to record for the books, and will ultimately improve the overall quality, efficiency and effectiveness of the “Books for Tots” scheme.

# Document Revision

Rev. 1.0 9th of February 2017 – initial version

Rev. 1.1 16th of February 2017 – Made changes to the walkthrough and updated system requirements

Rev. 1.2 20th of February 2017 – Adjusted system requirements to feature admin

Rev. 1.3 23rd of February 2017 – Added Use Case, Use Case Specification and Activity Diagram. Added requirements for the system

Rev. 1.4 24th of February 2017 – Added non-functional requirements, a draft of the Graphical User Interface design and a conclusion.

# Scope

The purpose of this project is to create a user friendly system for the management of a library within a primary. The primary user will be a parent from the school with a student count of 20.

This project will be designed with the focus of making the user experience as intuitive as possible. The user interface will consist of individual buttons that will correspond with each function that will interact with the book database. These functions will consist of a login function which will log the user in so that they can interact with the interface in order to manipulate the database.

The update functions will consist of buttons that will correspond with the various update functionalities. These will include the loan date, return date, condition of the book on return and update the student details who has taken the book out on loan.

When the program is launched the login screen will appear to the user and the correct username and password must be entered.

1. **Walkthrough Scenarios**

User:

User should be able to login, providing the username and password that meet the credentials provided to the user on the initial setup. This will display the Graphical User Interface (GUI) that will enable the user to interact with the database. Through this system the user will be able to add books, edit book details (book condition, author names etc.), loan dates and return dates.

Admin:

The admin will be able to change the username and password in the event that the user misplaces their login details. The admin will accomplish this by entering their own username and login details. Once these details are entered correctly the admin will be presented with a GUI that will allow them to enter the new username and password of the user into the fields provided. These details will be saved once the admin completes this form.

# Software Requirements Analysis:

***Functional Requirements:***

User:

* Login: The system will allow the user to log into the system using the details provided. This will display the Graphical User Interface (GUI).
* Add book: The system should allow the user to add a new book to the library system. This will display blank fields for them to enter the title, the author, the publisher, the condition of the book and the release date.
* Update book: The system should allow the user to update a books condition on return. If the book is in the same condition, then no action is taken.
* Add loan date: The system should allow the user to add a loan date if a book is currently out on loan, if it is not, the system takes no action.
* Add return date: The system should allow the user to add a return date to a book if it is out on loan.

Admin:

* Login: The system should allow the admin to log into the system using their details.
* Change passwords: The system should allow the admin to change the password of the user in the event that the user forgets their details.
* Change username: The system should allow the admin to change the username of the user in the event that the user forgets their details.

System:

* The system will check if the username and password match the credentials, if they do not, the user will be notified by an error message and prompted to login again, and if the text fields are blank, a message will appear asking them to enter a value. When the user logs in successfully, the system will display the GUI and make a connection to the database.
* When a user selects to add a book, they must be logged in to the system before displaying the blank text fields. If the user is successfully logged in, the GUI will display blank text fields for the user to enter new book details. The user cannot add a new book if the text fields are empty, if they are empty, the user will be notified. The title can consist of numbers and characters, author must only consist of characters, if the user attempts to add an author with numbers the system will notify the user that this is not allowed and that an author can only consist of characters. These rules apply to publishers and condition of the book. Release date must be in the format of dd/mm/yyyy, if any other format is attempted to be used the user will be notified by an error message telling them the correct format. The system should make a connection to the database.
* When a user selects to update the condition of the book on return date, the system will only allow the user to edit the condition and the system will block the other text fields from being edited.
* If a book is out on loan, the user will be required by the system to add a loan date. Loan dates are required to be in the format of dd/mm/yyyy, any other format will result in the user being notified by an error message.
* When a book is returned, the user will be required by the system to add a return date and update the condition of the book on return. Return date is required to be in the format of dd/mm/yyyy, and condition of the book cannot be an empty text field, if the user leaves the text field blank, the system should notify them with an error message and require input.

# 5.1 Non-functional Requirements:

* The system should log the user in instantly if they have entered the right details.
* When the user adds a book, the system should add the book in approximately 10 ms.
* Updating book details should be done instantly when the user clicks update.

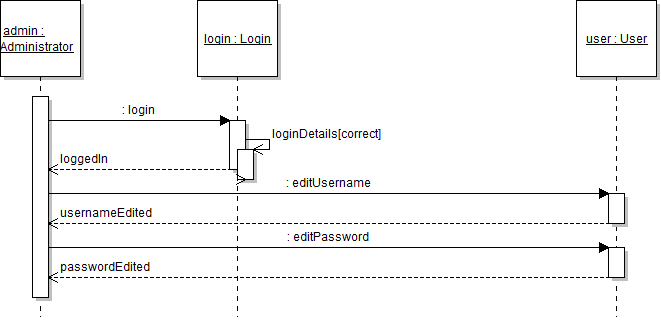
# 5.2 Use Case Diagram



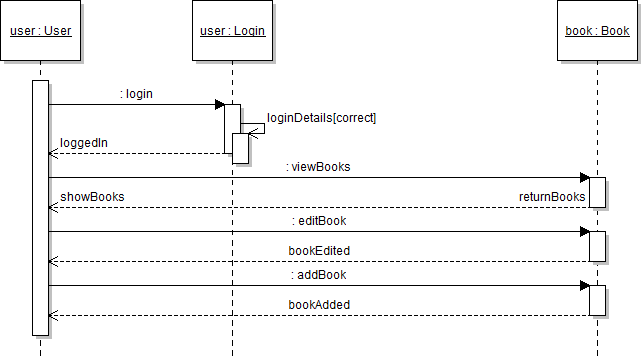
# 5.3 Class Diagram

# 

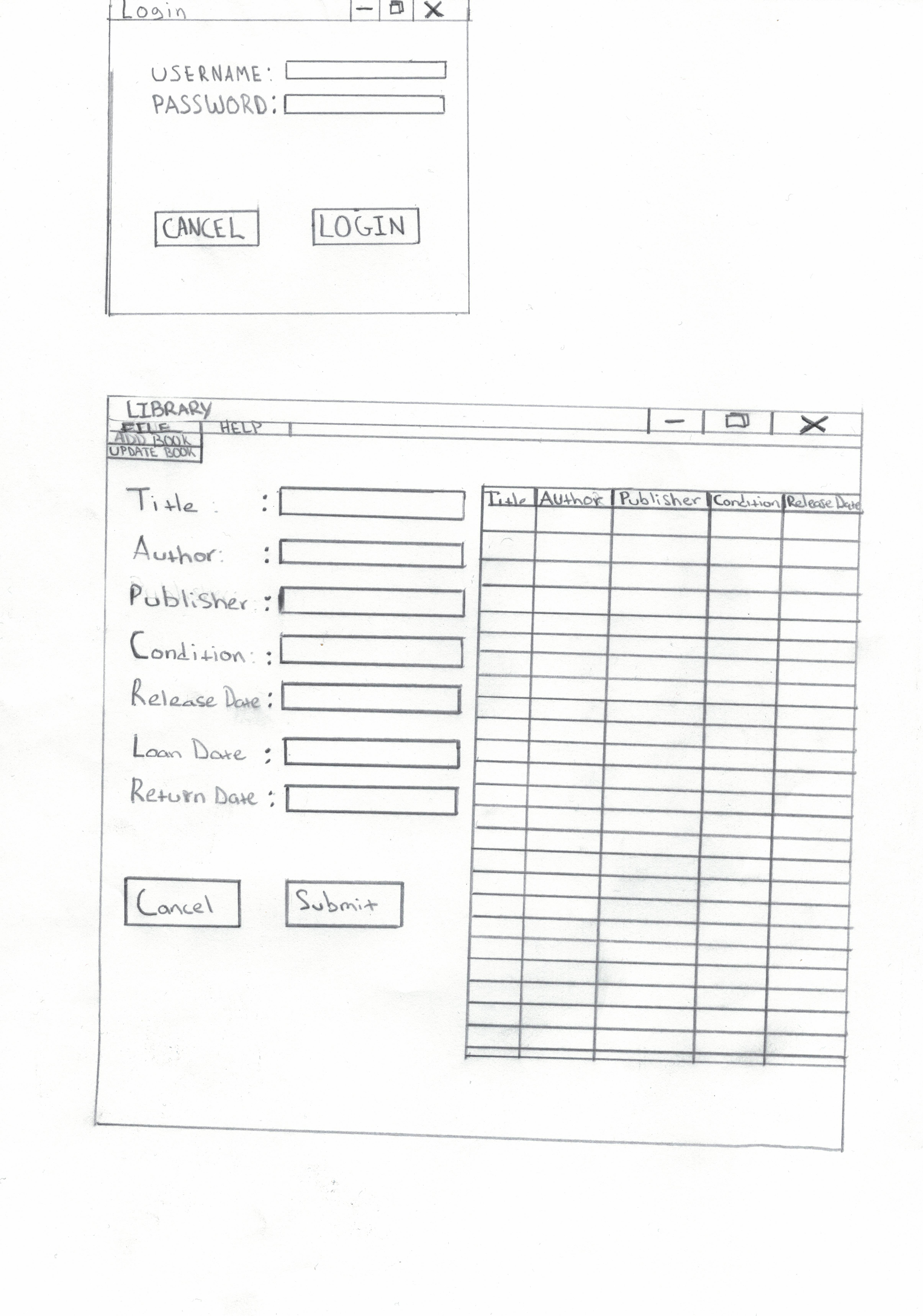
# 5.4 Admin Sequence Diagram



# 5.5 User Sequence Diagram



# Graphical User Interface Design



# Technical Requirements and Feasibility:

System models: This will be developed using the UML and documented through the rational rose environment.

Development language: The system software will be implemented in Java SE.

Persistent storage: A database will be used to maintain the book details as well as the user’s login details. This will be developed in MySQL.

Interface: The interface will be developed in Java using the Swing/AWT APIs.

# Conclusion

Based on this document and the customer’s needs, the group has agreed that the project is doable and will go ahead according to the project plan. There may be some minor changes to the system and the overall design when it comes to the implementation of the project, when necessary and with confirmation from the customer.

# Appendix A - Project Diary

***9th of February 2017***

The group was given the initial version of the requirements document. Joseph and Eoghan reviewed the sample requirements document that was uploaded by the lecturer on moodle. As a team, we collaborated on the layout of the requirements document and we assigned a different task to each individual based on their strengths.

***16th of February 2017***

On that date, the team gathered to discuss about the scope of the project and started working on it. There were few difficulties at the beginning on what was required for the scope. Nikos on his part researched online for the scope of a library system. (<http://www.academia.edu/10360207/The_Scope_Due_to_library_management_system>)

As a team, we began to work on the scope of the requirements document, which was then completed by Eoghan.

We then continued with the walkthrough scenarios and as a team, we wrote the user part of the document. Joseph volunteered to write the admin part of the walkthrough scenarios. In addition to this, we started as a team working on the system requirements.

***20th of February 2017***

Joseph and Eoghan worked on the user functional requirements of our library system. Josep was assigned with the task of finishing writing the user functional requirements and then write the admin part of the functional requirements.

Eoghan created the two sequence diagrams (User & Admin), a class diagram and a use case diagram. He used Violet UML Editor to carry out the task that he was assigned to create the diagrams.

***23rd of February 2017***

Nikos added the diagrams created by Eoghan to the requirements document. Joseph and Nikos began to work on the system functional requirements. The user graphic interface design was done by Niko and the Technical Requirements and Feasibility was done by Joseph.

***24th of February 2017***

Joseph added to the system functional requirements, wrote the conclusion and the non-functional requirements. Nikos did the table of contents for the requirements document. Eoghan and Niko were assigned to keep a diary of the meetings that the team had for the requirements document.