

Filippo Lobisch
Dragan Postolovski
Mike Schmitz

Issuing University of Luxembourg

Blockchain Application (v1.0)

Messip User Manual
- v 1.0.0 -

Based on IEEE Std 1063-2001 [\[1\]](#)

Tuesday 7th November, 2017 - 17:03

Contents

1	Product information	5
1.1	Identification	5
1.2	Copyright	5
1.3	Trademark notices	5
1.4	Restrictions	5
1.5	Warranties	5
1.6	Contractual obligations	5
1.7	Disclaimers	5
1.8	Contact	5
2	Introduction	7
2.1	Scope	7
2.2	Purpose	7
2.3	Intended audience	7
2.4	<i>Blockchain Application (v1.0)</i>	8
2.4.1	Actors & Functionalities	8
2.4.2	Operating environment	8
2.5	Document structure	8
3	Usage Guide	9
3.1	Multi-procedures	10
3.1.1	MyMultiProcedure1	10
3.1.2	MyMultiProcedure2	10
3.1.3	MyMultiProcedure3	10
3.2	Mono-procedures	10
3.2.1	MyActor1	10
3.2.2	My-Actor2	10
3.3	Librarians	11
3.3.1	Multi-procedures	11
3.3.2	Mono-procedures	13
3.4	Clients	16
3.4.1	Multi-procedures	16
3.4.2	Mono-procedures	17
4	Software operations	21
4.1	System Administrator	21
4.1.1	Add Librarian	21
4.1.2	Edit Librarian	21
4.1.3	Delete Librarian	22
4.1.4	Messages	22
4.2	Librarian	23
4.2.1	Access Customer Database	23
4.2.2	Search Customer	23
4.2.3	Inspect Customer	23
4.2.4	Add Loan	24
4.2.5	Return Book	24

4.2.6	Suspend Customer	24
4.2.7	Reinstate Customer	25
4.2.8	Request to ban customer	25
4.2.9	Request to unban customer	25
4.2.10	Notes	26
4.2.11	Access Book Database	26
4.2.12	Search Book	26
4.2.13	Add Book	27
4.2.14	Remove Book	27
4.3	Clients	27
4.3.1	Request book loan	27
4.3.2	Request book reservation	28
4.3.3	Cancel book reservation	28
5	Error messages and problem resolutions	29
5.1	Error message 1	29
5.1.1	Problem identification	29
5.1.2	Probable cause	29
5.1.3	Corrective actions	29
A	Title of the appendix 1	31
A.1	My Section	31
A.1.1	My subSection	31
	References	33

List of Figures

Chapter 1

Product information

1.1 Identification

This Blockchain Library System is software developed for usage by libraries, which is composed of three main parts. Those are system administrators' software with desktop client and web application, librarians' software with desktop and web application and a mobile application for clients' portable devices.

The system implements a blockchain data structure that is used to store the clients' book loans, in order to prevent security breaches and improve the stability of the system.

1.2 Copyright

1.3 Trademark notices

1.4 Restrictions

Restrictions on copying or distributing the software and its associated documentation.

1.5 Warranties

1.6 Contractual obligations

1.7 Disclaimers

1.8 Contact

Information for contacting the issuing organization.

Chapter 2

Introduction

2.1 Scope

This document provides information regarding the usage of the *Blockchain Application (v1.0)* software.

This document is not intended to provide information about how to connect, deploy, configure, or use any external device or third-party software system that is required for the correct functioning of *Blockchain Application (v1.0)*.

This document may be used with other documents provided by third-party companies to have an overall view and correct understanding of the environment and procedures where the software system *Blockchain Application (v1.0)* is aimed to be deployed and run.

2.2 Purpose

This user manual is aimed towards three user groups the system administrator, librarians and clients. It is aimed at the System Administrator, who will be using the desktop app of the sysadmin to access all its related functionality.

This document's purpose is to provide a way to help the system administrators in understanding, navigating the different panels and explaining how the different functionalities work and are used.

This document is intended to be read by system administrators making it easier for them to adapt and understand the application.

This user manual is also aimed towards the library's clients, who will be using the mobile app in order to gain access to their library account.

The purpose of this document is to help the users in navigating the different panels, as well as teach them the functionality that the application provides.

This document is meant to be read by the clients in order to make it more easy for them to adapt to using the application.

2.3 Intended audience

This document is meant to be used by all users (System Administrator, Librarians and Clients) of the *Blockchain Application (v1.0)* software. The users must be in a library environment (School library or public library) in order for these applications to work. The intended System Administrator is someone with a good tech background who can easily solve technological issues for the librarians after using the messages system.

2.4 *Blockchain Application (v1.0)*

This software is designed to be used in accordance with a library system. The users are able to browse through list of loan-available books, send loan request for a certain book(s) and send reservation requests for temporarily unavailable books.

The system administrators are able to browse through the list of librarians, add, edit and delete librarians and send and receive messages to librarians. The system administrator is also able to view the librarians history through a log.

2.4.1 *Actors & Functionalities*

Overview of all the *actors* interacting with the software being them either humans (called end-users in the standard [1]) or not. For each actor, describe the main software functions that are offered to him. Structure of this sub-section MUST be by actor/functionalities.

2.4.2 *Operating environment*

The software will be deployed on both desktops and mobile platforms. The desktop version will be supported in the three most popular systems which are Windows, macOS and Linux. As for the mobile platforms there will be both an Android version as well as iOS version.

2.5 Document structure

The document is structured where in each chapter you will see information about the system administrators before librarians and then clients.

Chapter 3

Usage Guide

This section is aimed at describing the general use of the software, since it is **deployed, configured and run**.

This software is used by actors. These actors rely on the software to perform a set of business activities (called here procedures) aimed at reaching a particular goal.

These procedures are split in two groups:

- **Multi-procedures:** which are procedures at **summary** or **user-goal** level involving several active or pro-active actors. Each of these procedures aims at illustrating intertwined business activities required to be performed by the involved actors to reach the expected goal. Each business activity between the system and an actor must correspond to a **system operation** instance given with actual parameter values.
- **Mono-procedures:** which are procedures at **summary** or **user-goal** level involving only one active or pro-active actor. Each of these procedures aims at illustrating the required business activities an actor has to perform to reach the expected goal. Each business activity between the system and the actor must correspond to a **system operation** instance given with actual parameter values.

Each process has to be documented using the following textual description template [2] **BUT its content must be as low level as possible with actual values:**

Procedure: ProcessMissionOne

Scope: Crisis Management System (*CMS*)

Primary Actor: Coordinator John

Secondary Actor(s): FirstAidWorker Bob,
ExternalResourceSystem ERS

Goal: The intention of the Coordinator is to process mission with ID equal to 1.

Level: User-goal level

Main Success Scenario :

1. *John* instructs the *CMS* to process the mission with ID equal to 12.031005
2. *CMS* selects the internal worker *Bob* to execute the mission 12.031005
3. *CMS* instructs *Bob* to behave as *First Aid Worker (FAW)*
4. *Bob* informs the *CMS* of his arrival
5. *Bob* informs the *CMS* that he starts to execute the mission 12.031005
6. *Bob* informs the *CMS* that the mission 12.031005 outcome is "Mission completed"

Extensions :

- 2.a None internal worker can execute the mission
 - 2.a.1 *CMS* sends a request for an external resource to the *ERS* actor instance
 - 2.a.2 *ERS* informs *CMS* that the request can be processed
 - 2.a.3 *ERS* informs *CMS* that *Bob* can now be selected as first aid worker
- procedure continues at step 3**
-

Remark-Processes presentation: processes should be introduced to the reader in a pedagogical manner. Thus, simple and common processes should be presented before than more complex and less utilised ones.

Remark-Graphical User Interfaces (GUIs): include GUIs screenshots to show the different stages of the process while its is performed by the actor(s).

3.1 Multi-procedures

3.1.1 *MyMultiProcedure1*

...

3.1.2 *MyMultiProcedure2*

...

3.1.3 *MyMultiProcedure3*

...

3.2 Mono-procedures

Mono-procedures must be grouped by actors.

3.2.1 *MyActor1*

3.2.1.1 MyProcedure1MyActor1

...

3.2.1.2 MyProcedure2MyActor1

...

3.2.2 *My-Actor2*

3.2.2.1 MyProcedure1MyActor2

...

3.2.2.2 MyProcedure2MyActor2

...

3.3 Librarians

3.3.1 Multi-procedures

Procedure: Librarian login

Scope: LibrarianSystem (*LibSys*)

Primary Actor: Librarian Smith

Secondary Actor(s): LoginSystem LS

Goal: The intention of the User is to login into the system in order to use the services of the application.

Level: User-goal level

Main Success Scenario :

1. *Smith* enters the user name and password in the respective fields
2. *Smith* presses on the login button which sends a login request message to *LS*
3. *LS* logs *Smith* into the application

Extensions :

- 3.a Username of *Smith* not found
 - 3.a.1 *LS* returns a message stating that the user name or password is not correct
procedure returns to step 1
 - 3.b Password of *Smith* not correct
 - 3.b.1 *LS* returns a message stating that the user name or password is not correct
procedure returns to step 1
-

Procedure: Grant book loan

Scope: LibrarianSystem (*LibSys*)

Primary Actor: Librarian Smith

Secondary Actor(s): Customer John

Goal: The intention of the User is to grant a loan for a book requested by a customer.

Level: User-goal level

Main Success Scenario :

1. *John* is approaching the librarian with a book he wants to loan.
2. *John* gives the librarian his ID so the librarian can look up his account
3. *Smith* opens up *John*'s account and clicks on "Add loan".
4. *Smith* enters the book ID in the popup window and clicks on "Add Loan".

Extensions :

- 2.a The librarian cannot verify the customer account
 - 3.a *Smith* decides not to allow the loan, either due to too many active loans, or because a suspended or banned account.
 - 3.a.1 *John* gets told by *Smith* that he cannot currently loan a book and why.
procedure ends
-

Procedure: Close book loan

Scope: LibrarianSystem (*LibSys*)

Primary Actor: Librarian Smith

Secondary Actor(s): Customer John

Goal: The intention of the User is to close a loan for a book returned by a customer.

Level: User-goal level

Main Success Scenario :

1. *John* is approaching the librarian with a book he wants to return.
2. *John* gives the librarian his ID so the librarian can look up his account
3. *Smith* opens up *John*'s account and selects the book *John* wants to return.
4. *Smith* then clicks on "Return Book".

Extensions :

- 2.a The librarian cannot verify the customer account
 - 3.a *Smith* sees that either the book *John* wants to return is overdue and *John* needs to pay a fine, or due to too many overdues his account has been suspended/banned.
 - 3.a.1 *John* gets told by *Smith* that he needs to pay fine since the books is overdue or his account is suspended/banned.
procedure ends
-

Procedure: Suspend Customer

Scope: LibrarianSystem (*LibSys*)

Primary Actor: Librarian Smith

Secondary Actor(s): Customer John

Goal: The intention of the User is to suspend the customer account until a certain fine is paid.

Level: User-goal level

Main Success Scenario :

1. *Smith* has been made aware of a customer with multiple overdue books.
 2. *Smith* opens this customers account and clicks on "Suspend Customer"
 3. *Smith* enters a small descriptive note in the text field on the customer page.
 4. *John* now can't loan any more books until the fine is paid off and every librarian accessign his account sees what the fine is.
-
-

Procedure: Reinstate Customer

Scope: LibrarianSystem (*LibSys*)

Primary Actor: Librarian Smith

Secondary Actor(s): Customer John

Goal: The intention of the User is to reinstate a suspended customer account.

Level: User-goal level

Main Success Scenario :

1. *John* has been made aware that his customer account got suspended and he needs to pay a fine to reinstate it.
2. *John* goes to the library and gives *Smith* his customer ID so he can pull up his account.
3. *Smith* opens this customers account and checks the fine *John* needs to pay.
4. *Jonh* pays *Smith* the fine.
5. *Smith* receives the payment and clicks on "Reinstate Account".

Extensions :

- 4.a *John* doesn't have enough money on him to pay the fine and thus cannot complete the reinstation of his customer account.
-
-

Procedure: Ban Customer

Scope: LibrarianSystem (*LibSys*)

Primary Actor: Librarian Smith

Secondary Actor(s): LibrarianSupervisor Jane, SysAdmin Fil

Goal: The intention of the User is to put in a request to ban a customer.

Level: User-goal level

Main Success Scenario :

1. *Smith* has been made aware of a deviant customer with multiple infractions.
2. *Smith* opens this customers account and click on "Ban Customer"
3. *Smith* enters a small descriptive text and clicks "Send".
4. *Jane* receives this request and approves this.
5. *Fil* sees the request has been approved and closes the customer account.

Extensions :

- 4.a *Jane* denies the request to close the account and thus *Fil* never gets involved.
-
-

Procedure: Unban Customer

Scope: LibrarianSystem (*LibSys*)

Primary Actor: Librarian Smith

Secondary Actor(s): LibrarianSupervisor Jane, SysAdmin Fil, Customer John

Goal: The intention of the User is to put in a request to unban a customer.

Level: User-goal level

Main Success Scenario :

1. *John* has paid off all his fines at the library and his ban has been in effect for quite some time and he wants to be unbanned and promises to follow the rules.
2. *Smith* opens this customers account and click on "Unban Customer"
3. *Smith* enters a small descriptive text and clicks "Send".
4. *Jane* receives this request and approves this.
5. *Fil* sees the request has been approved and reopens the customer account.

Extensions :

- 1.a *Smith* sees that *John* actually hasn't paid of all his fines or for whatever reason doesn't think *John* shouldn't be allowed to put in a request to reopen his account.
 - 4.a *Jane* denies the request to reopen the account and thus *Fil* never gets involved.
-

3.3.2 Mono-procedures

Procedure: Search books list**Scope:** LibrarianSystem (*LibSys*)**Primary Actor:** Librarian Smith**Secondary Actor(s):****Goal:** The intention of the User is to search the book database for a specific book.**Level:** User-goal level**Main Success Scenario :**

1. *Smith* is at the home panel of the application after logging in.
2. *Smith* now clicks on the button "Books" to get to the book database and search for books.
3. In the search field on top *Smith* enters either the book name, ISBN or author of the book he wants to find.
4. In the dropdown menu next to the search field *Smith* chooses which property he just entered.
5. *Smith* presses "Search" to get a list of all the books matching his search query.

Extensions :

- 5.a *Smith* left the search field blank, which then prints him a list of all the books in the book database.
-

Procedure: Search customer list**Scope:** LibrarianSystem (*LibSys*)**Primary Actor:** Librarian Smith**Secondary Actor(s):****Goal:** The intention of the User is to search the customer database for a specific customer.**Level:** User-goal level**Main Success Scenario :**

1. *Smith* is at the home panel of the application after logging in.
2. *Smith* now clicks on the button "Customers" to get to the customer database and search for customers.
3. In the search field on top *Smith* enters either the first name, last name, customer ID or account status of the customer he wants to find.
4. In the dropdown menu next to the search field *Smith* chooses which property he just entered.
5. *Smith* presses "Search" to get a list of all the customers matching his search query.

Extensions :

- 5.a *Smith* left the search field blank, which then prints him a list of all the customers in the customer database.
-

Procedure: Inspect customer**Scope:** LibrarianSystem (*LibSys*)**Primary Actor:** Librarian Smith**Secondary Actor(s):****Goal:** The intention of the User is to inspect a customers profile to get further information about current or past loans, comments from other librarians, or take further actions with his account.**Level:** User-goal level**Main Success Scenario :**

1. *Smith* successfully launched a search query on the customer database.
2. *Smith* now selects the customer he wishes to inspect from the list displayed in the application matching his search query.
3. A new window will popup displaying per default current loans, with the possibility to show past loans instead, as well as giving the librarian the possibility to grant a loan, make a book return, suspend or reinstate the account, request a ban or unban and add notes to his page.

Extensions :

Procedure: Show loan history

Scope: LibrarianSystem (*LibSys*)

Primary Actor: Librarian Smith

Secondary Actor(s):

Goal: The intention of the User is to check a customers loan history

Level: User-goal level

Main Success Scenario :

1. *Smith* is at the account page of a customer.
2. *Smith* clicks the button on the bottom right titled "History" which replaces the current table of current loans which a table of past loans as well as replacing the button with another one to get back to current loans.

Extensions :

Procedure: Edit customer notes.

Scope: LibrarianSystem (*LibSys*)

Primary Actor: Librarian Smith

Secondary Actor(s):

Goal: The intention of the User is to edit the notes and comments on a customers account page.

Level: User-goal level

Main Success Scenario :

1. *Smith* is at the account page of a customer.
2. *Smith* clicks in the text box on the bottom of the window and adds his notes and comments on the customers in this field like fines to be paid or suspension issued at date DD/MM/YYYY.

Extensions :

Procedure: Add book.

Scope: LibrarianSystem (*LibSys*)

Primary Actor: Librarian Smith

Secondary Actor(s):

Goal: The intention of the User is to add a book to the book database.

Level: User-goal level

Main Success Scenario :

1. *Smith* is at the book database menu.
2. *Smith* clicks on the button labeled "Add Book".
3. *Smith* enters the books ISBN into the popup window and clicks on "Add Book".
4. The system now automatically adds a new entry for this book, looking up the author and title associated with the ISBN and generating a new book ID for this book.

Extensions :

- 3.a *Smith* entered a wrong ISBN prompting him to either cancel the procedure or manually adding book name and author name.
-

Procedure: Remove book.

Scope: LibrarianSystem (*LibSys*)

Primary Actor: Librarian Smith

Secondary Actor(s):

Goal: The intention of the User is to remove a book from the book database.

Level: User-goal level

Main Success Scenario :

1. *Smith* is at the book database menu.
2. *Smith* selects the book he wishes to remove from the book database.
3. *Smith* clicks on the button labeled "Remove Book".
4. The system now removes this particular copy of a book from the database.

Extensions :

- 3.a *Smith* chose a book currently being loaned which will bring up a prompt telling him he can't remove this copy from the database.
- 3.b *Smith* chose a book currently being reserved by a customer which will bring up a prompt if he really wants to remove

this book from the database as this will cancel the reservation.

3.4 Clients

3.4.1 Multi-procedures

Procedure: User login

Scope: Library mobile application (*LMA*)

Primary Actor: User John

Secondary Actor(s): LoginSystem LS

Goal: The intention of the User is to login into the system in order to use the services of the application.

Level: User-goal level

Main Success Scenario :

1. *John* enters the user name and password in the respective fields
2. *John* presses on the login button which sends a login request message to *LS*
3. *LS* logs *John* into the application

Extensions :

- 3.a Username of *John* not found
 - 3.a.1 *LS* returns a message stating that the user name or password is not correct
procedure returns to step 1
- 3.b Password of *John* not correct
 - 3.b.1 *LS* returns a message stating that the user name or password is not correct
procedure returns to step 1

Procedure: Request book loan

Scope: Library mobile application (*LMA*)

Primary Actor: User John

Secondary Actor(s): LibrarianSystem LibSys, Librarian Smith

Goal: The intention of the User is to request a loan for a book he is interested in.

Level: User-goal level

Main Success Scenario :

1. *John* is browsing the library's list of books.
2. *John* clicks on a book he wants to request a loan for.
3. A new panel is shown where *John* can choose whether he wants to request a loan or not.
4. *John* presses on the 'Send loan request' button.
5. A request is sent to *LibSys* and it is stored in the database.
6. *Smith* processes the request and allows the loan to *John*.
7. *LibSys* sends a notification to *John* saying that his loan is approved and he can pick up his book at the library .

Extensions :

- 3.a The book selected has no physical copy available at the moment
 - 3.a.1 The application shows a panel that gives *John* the choice to send a reservation request or not for the selected book
for continuation , see procedure : Request book reservation
- 6.a *Smith* decides not to allow the loan
 - 6.a.1 *John* receives a notification saying that his loan request has been denied.
procedure ends

Procedure: Request book reservation

Scope: Library mobile application (*LMA*)

Primary Actor: User John

Secondary Actor(s): LibrarianSystem LibSys, Librarian Smith

Goal: The intention of the User is to request a reservation for a book he is interested in.

Level: User-goal level

Main Success Scenario :

1. *John* is browsing the library's list of books.
2. *John* clicks on a book he wants to request a loan for, but the book has zero(0) physical copies available.
3. A new panel is shown where *John* can choose whether he wants to request a reservation or not.

4. *John* presses on the 'Send reservation request' button.
5. A request is sent to *LibSys* and it is stored in the database.
6. *LibSys* puts *John's* request in a queue for the selected book.
7. When *John* is the first in the queue for the reserved book, a notification is sent, asking whether *John* wants to confirm and get the loan, or cancel it.

Procedure: Edit phone number

Scope: Library mobile application (*LMA*)

Primary Actor: User John

Secondary Actor(s): PhoneVerificationSystem VerSys

Goal: The intention of the User is to change his phone number that is connected to the library account.

Level: User-goal level

Main Success Scenario :

1. *John* is at the home panel of the application after logging in.
2. *John* presses either his profile picture icon, or his name on the top of the screen.
3. *John's* personal information, as well as the buttons to change it are now visible.
4. *John* clicks on the 'Change phone number' button.
5. *John* enters his new phone number in the first input field marked as 'New phone number'.
6. *John* presses the 'Send SMS' button which makes VerSys send a code to the number that has been input in the first input field.
7. *John* receives a 4-pin code on the phone that has the new number.
8. *John* inputs the received pin code in the second input field marked as 'Code received'.
9. *John* presses the 'Confirm number change' button.
10. *John* is sent to the confirmation screen, after which he is redirected to the main personal information screen.

Extensions :

- 7.a *John* doesn't receive the code.
 - 7.a.1 *John* entered the wrong number in the first input field.
 - 7.a.2 If *John* doesn't enter the sent code in 60 seconds, the request will time out and he will be redirected to the main personal settings panel.

3.4.2 Mono-procedures

Procedure: Browse books list

Scope: Library mobile application (*LMA*)

Primary Actor: User John

Secondary Actor(s):

Goal: The intention of the User is to get the full list of books in the library and scroll through them.

Level: User-goal level

Main Success Scenario :

1. *John* is in at the home panel of the application after logging in.
2. The library's book list is available for scrolling through.

Extensions :

- 1.a Current panel is 'Loans' panel
 - 1.a.1 *John* clicks on the 'Browse books' button
 - 1.a.2 The visible panel is the home panel

procedure continues to step 2
- 1.b *John* is in 'Reservations' panel
 - 1.b.1 *John* clicks on the 'Browse books' button
 - 1.b.2 The visible panel is the home panel

procedure continues to step 2
- 1.c *John* is in 'Personal information' panel
 - 1.c.1 *John* clicks on the 'Back' button

procedure returns to step 1

Procedure: Browse loans list

Scope: Library mobile application (*LMA*)

Primary Actor: User John

Secondary Actor(s):

Goal: The intention of the User is to get the full list of loans to his name and scroll through them.

Level: User-goal level

Main Success Scenario :

1. *John* is in at the home panel of the application after logging in.
2. *John* presses the 'Browse loans' button on the bottom of the screen.
3. *John's* loan list is available for scrolling through.

Extensions :

- 1.a *John* is in 'Personal information' panel
 - 1.a.1 *John* clicks on the 'Back' button near the top of the screen**procedure returns to step 1**
- 2.a *John* is in 'Reservations' panel
 - 2.a.1 *John* clicks on the 'Loans' button under his profile picture near the top of the screen
 - 2.a.2 The visible panel is the loans panel**procedure continues to step 3**

Procedure: Browse reservation list

Scope: Library mobile application (*LMA*)

Primary Actor: User John

Secondary Actor(s):

Goal: The intention of the User is to get the full list of reservations to his name and scroll through them.

Level: User-goal level

Main Success Scenario :

1. *John* is at the home panel of the application after logging in.
2. *John* presses the 'Browse loans' button on the bottom of the screen.
3. *John* presses the 'Reservation' button under his profile picture, near the top of the screen
4. *John's* reservation list is available for scrolling through.

Extensions :

- 1.a *John* is in 'Personal information' panel
 - 1.a.1 *John* clicks on the 'Back' button near the top of the screen**procedure returns to step 1**

Procedure: Edit e-mail address

Scope: Library mobile application (*LMA*)

Primary Actor: User John

Secondary Actor(s):

Goal: The intention of the User is to change his e-mail address that is connected to the library account.

Level: User-goal level

Main Success Scenario :

1. *John* is at the home panel of the application after logging in.
2. *John* presses the either his profile picture icon, or his name on the top of the screen.
3. *John's* personal information, as well as the buttons to change it are now visible.
4. *John* clicks on the 'Change email address' button.
5. *John* enters his new e-mail address in the first input field marked as 'New email address'.
6. *John* enters his new e-mail address in the second input field, marked as 'Confirm new email address'.
7. *John* presses the 'Submit' button, which confirms the change of e-mail address.
8. *John* is redirected to the main personal information panel.

Procedure: Edit password

Scope: Library mobile application (*LMA*)

Primary Actor: User John

Goal: The intention of the User is to change his account's password.

Level: User-goal level

Main Success Scenario :

1. *John* is at the home panel of the application after logging in.
2. *John* presses either his profile picture icon, or his name on the top of the screen.
3. *John's* personal information, as well as the buttons to change it are now visible.

4. *John* clicks on the 'Change password' button.
5. *John* enters his new old password in the first input field marked as 'Old password'.
6. *John* enters his new password in the second input field marked as 'New password'.
7. *John* enters his new password in the third input field marked as 'Confirm new password'.
8. *John* presses the 'Submit' button which completes the request.
9. *John* is sent to the confirmation screen, after which he is redirected to the main personal information screen.

Procedure: Edit address

Scope: Library mobile application (*LMA*)

Primary Actor: User John

Goal: The intention of the User is to change his current living address.

Level: User-goal level

Main Success Scenario :

1. *John* is at the home panel of the application after logging in.
2. *John* presses either his profile picture icon, or his name on the top of the screen.
3. *John's* personal information, as well as the buttons to change it are now visible.
4. *John* clicks on the 'Change address' button.
5. *John* enters his new address in the three available input lines in the panel that is shown.
6. *John* presses the 'Submit' button which completes the request.
7. *John* is sent to the confirmation screen, after which he is redirected to the main personal information screen.

Procedure: Add Visa card as a payment option

Scope: Library mobile application (*LMA*)

Primary Actor: User John

Goal: The intention of the User is to add a Visa card to his account as an option of payment

Level: User-goal level

Main Success Scenario :

1. *John* is at the home panel of the application after logging in.
2. *John* presses either his profile picture icon, or his name on the top of the screen.
3. *John's* personal information, as well as the buttons to change it are now visible.
4. *John* clicks on the 'Edit payment options' button.
5. *John* presses on the 'Add' button that is to the right of the 'Visa card' text.
6. *John* enters the card holder name and surname in the first input line in the newly opened panel.
7. *John* enters the card's number in the second input line.
8. *John* enters the card's security number in the third input line.
9. *John* presses on the 'Add new card' button in the lower right corner of the screen.
10. The card whose information has just been input is added to *John's* Visa payment option.

Procedure: Add Mastercard as a payment option

Scope: Library mobile application (*LMA*)

Primary Actor: User John

Goal: The intention of the User is to add a Mastercard card to his account as an option of payment

Level: User-goal level

Main Success Scenario :

1. *John* is at the home panel of the application after logging in.
2. *John* presses either his profile picture icon, or his name on the top of the screen.
3. *John's* personal information, as well as the buttons to change it are now visible.
4. *John* clicks on the 'Edit payment options' button.
5. *John* presses on the 'Add' button that is to the right of the 'Mastercard' text.
6. *John* enters the card holder name and surname in the first input line in the newly opened panel.
7. *John* enters the card's number in the second input line.
8. *John* enters the card's security number in the third input line.
9. *John* presses on the 'Add new card' button in the lower right corner of the screen.
10. The card whose information has just been input is added to *John's* Mastercard payment option.

Procedure: Add PayPal account as a payment option

Scope: Library mobile application (*LMA*)

Primary Actor: User John

Goal: The intention of the User is to add a PayPal account to his account as an option of payment

Level: User-goal level

Main Success Scenario :

1. *John* is at the home panel of the application after logging in.
 2. *John* presses either his profile picture icon, or his name on the top of the screen.
 3. *John's* personal information, as well as the buttons to change it are now visible.
 4. *John* clicks on the 'Edit payment options' button.
 5. *John* presses on the 'Add' button that is to the right of the 'Paypal' text.
 6. *John* enters the PayPal account's email address in the first input line in the newly opened panel.
 7. *John* presses on the 'Add account' button in the lower right corner of the screen.
 8. The PayPal account whose e-mail address has just been input is added to *John's* PayPal payment option.
-

Procedure: Add AmericanExpress card as a payment option

Scope: Library mobile application (*LMA*)

Primary Actor: User John

Goal: The intention of the User is to add an AmericanExpress card to his account as an option of payment

Level: User-goal level

Main Success Scenario :

1. *John* is at the home panel of the application after logging in.
 2. *John* presses either his profile picture icon, or his name on the top of the screen.
 3. *John's* personal information, as well as the buttons to change it are now visible.
 4. *John* clicks on the 'Edit payment options' button.
 5. *John* presses on the 'Add' button that is to the right of the 'American Express' text.
 6. *John* enters the card holder name and surname in the first input line in the newly opened panel.
 7. *John* enters the card's number in the second input line.
 8. *John* enters the card's security number in the third input line.
 9. *John* presses on the 'Add new card' button in the lower right corner of the screen.
 10. The card whose information has just been input is added to *John's* AmericanExpress payment option.
-

Procedure: Remove a payment option

Scope: Library mobile application (*LMA*)

Primary Actor: User John

Goal: The intention of the User is to remove a certain payment option that is active on his account

Level: User-goal level

Main Success Scenario :

1. *John* is at the home panel of the application after logging in.
 2. *John* presses either his profile picture icon, or his name on the top of the screen.
 3. *John's* personal information, as well as the buttons to change it are now visible.
 4. *John* clicks on the 'Edit payment options' button.
 5. *John* presses on the 'Edit' button that is to the right of the active option that he wants to remove.
 6. *John* presses the 'Remove' button that is on the right side of the newly opened panel.
 7. The option is now successfully removed from *John's* account.
-

Chapter 4

Software operations

This section explains the different operations and functionalities of the system and how to attain them.

4.1 System Administrator

This subsection provides a detailed description of system administrator specific functionalities.

4.1.1 Add Librarian

The add librarian functionality is only available for the system administrator. As the title suggests, the system administrator creates and adds a new librarian account being informed from organization/library.

Parameters: ID (Randomly generated), First Name, Last Name, Date of Birth

Precondition: The system administrator must be logged into the system and must have received the information about the librarian too add.

Post-condition: The new librarian account been added to the database system and the librarian and the organization have received an automated notification so that the account can be finalized.

Output messages: The librarian and the organization will receive an automated notification informing that the account can now be finalized.

Triggering:

1. From the users main window, click on the Add Librarian button to open the add librarian page.
2. From the users main window, click on the Add Librarians button to open the add librarian page.
3. Once the add librarian page opens, fill out all the required parameters with information related to the new librarian that is being created.
4. Click on the Submit button to add the new librarians information to a database.

4.1.2 Edit Librarian

The edit librarian functionality is used when editing user information. This includes but is not limited to the first name, last name and date of birth as well as password eventually. The goal of this feature is to allow the system administrator to edit a librarians information at the request of the librarian.

Parameters: ID, First Name, Last Name

Precondition: The system administrator must be logged in, at the users page and must have the different information of the librarian that needs to be edited.

Post-condition: The edits for the librarian have been saved and a system popup appears stating that the librarian information has successfully been edited and saved.

Output messages: The system administrator will see a system popup informing the system administrator that the librarian information has successfully been edited and saved.

Triggering:

1. Click on the "Edit Librarian" button that is found in the users main page.
2. Once the edit librarian page has opened, either find the user manually or search the user via the search field at the top.
3. Click on the Librarian ID to open up the edit details page.
4. Edit the necessary details of the user that need to be edited. Click submit when complete. You will be automatically be taken back to the users page.

4.1.3 Delete Librarian

The delete librarian is once again only available for the system administrator. The delete librarian functionality is used when deleting an existing librarian account that will no longer be used due to different reasons such as; the librarian no longer works, an unfortunate event has happened to the librarian.

Parameters: ID (Randomly generated), First Name, Last Name

Precondition: The system administrator must be logged into the system and must have received the information about the librarian too remove.

Post-condition: The librarian account has been successfully deleted from the database system and the sysadmin will receive an onscreen notification informing that the user has been successfully deleted.

Output messages: The librarian and the organization will receive an automated notification informing that the account can now be finalized.

Triggering:

1. Click on the Delete Librarian button at the bottom of the users main
2. Click on the Delete Librarians button at the bottom of the users main window to open the users deletion page.
3. Once in the users deletion page, find the user that needs to be deleted manually or via the search field.
4. Once the user has been found, click the red x to remove the user permanently from the database system.
5. Click on the "back/done" button to return to users page.

4.1.4 Messages

The messages functionality is a system which allows Librarians to contact System Administrators and vice versa. Its main purpose is to allow an easy communication system to help System Administrators and Librarians communicate issues and changes more easily.

4.1.4.1 New Message

The new message functionality like the title suggests allows a system administrator to create a new message to send to a librarian.

Parameters: Librarian ID, First Name, Last Name, Message

Precondition: The system administrator must be logged in and must have received the information (ID, First Name, Last Name) of the librarian to whom the message has to be sent.

Post-condition: The message is sent to the Librarian and the system administrator will see the message as a rectangle in the chat.

Output messages: None.

Triggering:

1. Click on the new message button at the bottom left of the messages main panel.

2. Select the Librarian that needs to receive the message from the drop down menu.
3. Type the message you want to send in the text box at the bottom.
4. Click the send button at the top left when complete.

4.2 Librarian

This subsection provides a detailed description of librarian specific functionalities.

4.2.1 Access Customer Database

The "Access Customer Database" functionality is only available for librarians. As the title suggests, the librarian is able to access the database with all the customers in it and search for a specific customer to perform certain actions

Parameters: None

Precondition: The librarian must be logged into the system.

Post-condition: The librarian is now able to look for a specific customer using either their first name, last name, customer ID or account status (active, overdue, suspended, banned) and inspect them to perform certain actions on their account.

Output messages: None.

Triggering: Press the "Customers" button after logging into your librarian account.

4.2.2 Search Customer

The "Search Customer" functionality is only available for librarians. As the title suggests, the librarian is able to look for customers using either their first name, last name, customer ID or account status (active, overdue, suspended, banned).

Parameters: First name, last name, customer ID, account status or nothing if he intends to see the full database.

Precondition: The librarian must be logged into the system and chosen to access the customer database.

Post-condition: The librarian is now able to further inspect the customer and take further actions.

Output messages: Prints the results in a table containing First name, Last name, customer ID as well as account status (active, overdue, suspended, banned).

Triggering:

1. Enter the first name, last name, customer ID or account status into the search bar.
2. Select the relevant property from the adjoining dropdown menu.
3. Hit the "Search" button and a list of all customers matching the query will be displayed

4.2.3 Inspect Customer

The "Inspect" button becomes available as soon as the librarian has searched for a customer, so he can check the info available on a customer

Parameters: None

Precondition: The librarian must be logged into the system, received the necessary information to look up the customer and started a query for said customer.

Post-condition: The librarian is now able to grant loans to the inspected customers, initiate a book return, suspend and reinstate his account, add notes and put in a request to either ban or unban his account.

Output messages: None.

Triggering: Press the "Inspect" button after searching for customers and selecting the one you want to inspect.

4.2.4 Add Loan

The "Add Loan" functionality is available for the librarian. As the title suggest the librarian is able to add a loan to a customers account after inspecting the latter.

Parameters: Book ID

Precondition: The librarian must be in the inspect window for a customer and press the "Add loan" button.

Post-condition: The librarian has now added a loan to the customers account with the loan starting date automatically being set and the standard loan due date being 2 days from start.

Output messages: None.

Triggering:

1. Open the inspect window for the customer who wants to loan a book.
2. Select "Add loan" button from the bottom of the window.
3. Enter the book ID of the book the customer wants to loan.

4.2.5 Return Book

The "Return Book" functionality is available for the librarian. As the title suggest the librarian is able to let a customer return a book he loaned and add it back to the pool of available books.

Parameters: None

Precondition: The librarian must be in the inspect window for a customer and have the currents loans being displayed, then choose the book to return and click the "Return book" button.

Post-condition: The librarian has now successfully returned a book from the customer to the book database and that book loan is added to the loan history.

Output messages: None.

Triggering:

1. Open the inspect window for the customer who wants to loan a book.
2. Confirm you are shown the current loan and not the loan history.
3. Select the book to be returned.
4. Select "Return book" button from the bottom of the window.

4.2.6 Suspend Customer

The "Suspend Customer" functionality is available for the librarian. As the title suggest the librarian is able to suspend a customer disallowing him to make further loans until his account is reinstated at a librarians discretion (i.e. by paying a fine).

Parameters: None

Precondition: The librarian must be in the inspect window for a customer click the "Suspend Customer" button. Furthermore, this button won't be displayed if the customer is already suspended, instead a "Reinstate Customer" button will be shown.

Post-condition: The librarian has now suspended a customer and he is unable to make further loans until he is reinstated.

Output messages: None.

Triggering:

1. Open the inspect window for the customer to be suspended.
2. Select "Suspend Customer" button from the bottom of the window.

4.2.7 Reinstate Customer

The "Reinstate Customer" functionality is available for the librarian. As the title suggest the librarian is able to reinstate a customer reallowing him to make further loans.

Parameters: None

Precondition: The librarian must be in the inspect window for a customer click the "Reinstate Customer" button.

Post-condition: The librarian has now reinstated a customer and he can once again request loans.

Output messages: None.

Triggering:

1. Open the inspect window for the customer to be reinstated.
2. Select "Reinstate Customer" button from the bottom of the window.

4.2.8 Request to ban customer

The "Ban Customer" functionality is available for the librarian. With this functionality, the librarian is able to put in a request to ban the currently inspected customer by providing a small paragraph as to why the customer should be banned.

Parameters: None

Precondition: The librarian must be in the inspect window for a customer click the "Ban Customer" button.

Post-condition: The librarian has now issued a request to ban the customer in question.

Output messages: None.

Triggering:

1. Open the inspect window for the customer to be banned.
2. Select "Ban Customer" button from the bottom of the window.
3. A new window opens where the librarian can enter a reason as to why the customer should be banned.
4. Finally the librarian clicks the "Send" which sends his request off for review by a higher ranked person.

4.2.9 Request to unban customer

The "Unban Customer" functionality is available for the librarian. With this functionality, the librarian is able to put in a request to unban the currently inspected customer by providing a small paragraph as to why the customer should be unbanned.

Parameters: None

Precondition: The librarian must be in the inspect window of a currently banned customer and click the "Unban Customer" button.

Post-condition: The librarian has now issued a request to ban the customer in question.

Output messages: None.

Triggering:

1. Open the inspect window for the customer to be unbanned.
2. Select "Unban Customer" button from the bottom of the window.
3. A new window opens where the librarian can enter a reason as to why the customer should be unbanned.
4. Finally the librarian clicks the "Send" which sends his request off for review by a higher ranked person.

4.2.10 Notes

The "Notes" functionality is available for the librarian. With this functionality, the librarian is able to add notes to a customer account like why he has been suspended and what he has to do to be reinstated so the librarian doing his reinstation knows why he was suspended in this example.

Parameters: None

Precondition: The librarian must be in the inspect window of a customer and click into the input field at the bottom to allow notes to be added to his account.

Post-condition: The librarian has now added a note to the customers account.

Output messages: None.

Triggering:

1. Open the inspect window for a customer.
2. Click into the input field at the bottom of the window.
3. The librarian can now add comments to the customers account.
4. When the librarian is done adding comments and notes to the customers account, no further actions need to be taken.

4.2.11 Access Book Database

The "Access Books Database" functionality is only available for librarians. As the title suggests, the librarian is able to access the database with all the customers in it and search for a specific customer to perform certain actions

Parameters: None

Precondition: The librarian must be logged into the system.

Post-condition: The librarian is now able to look for a specific book using either the ISBN, the author or the title. Furthermore he is able to add books to the database and remove some.

Output messages: None.

Triggering: Press the "Books" button after logging into your librarian account.

4.2.12 Search Book

The "Search Book" functionality is available for the librarian. As the title suggest the librarian is able to view the books in the database as well as perform other book related tasks.

Parameters: Book name, ISBN, author, or nothing if he intends to see the full database.

Precondition: The librarian must be logged into the system, chosen to access the book database and received the necessary information to look up the book (if applicable).

Post-condition: The librarian is now able to check the status of all the books matching the search term, as well as add books to the database, or remove one from it.

Output messages: Prints the results in a table containing ISBN, Author, full name, internal ID, as well as loan status (available, loaned, overdue).

Triggering:

1. Enter the book name, ISBN, author into the search bar.
2. Select the relevant property from the adjoining dropdown menu.
3. Hit the "Search" button and a list of all books matching the query will be displayed

4.2.13 Add Book

The "Add Book" functionality is available for the librarian. As the title suggest the librarian is able to add books to the database after opening the latter through the "Search Book" functionality.

Parameters: ISBN

Precondition: The librarian must be logged into the system, opened the book database and pressed "Add book".

Post-condition: The librarian is now able to add a book to the database by entering the books ISBN number into the popup windows and confirming with the "Add book" button again. The system will automatically gather the author and book name from a server and assign it a random internal ID.

Output messages: None.

Triggering:

1. Open the library database with the "Search book" functionality by either searching for a specific book or viewing the whole database by leaving the search field blank.
2. Select "Add book" from under the table.
3. Enter the ISBN of the new book into the input field and confirm with "Add book" to add it to the database.

4.2.14 Remove Book

The "Remove Book" functionality is available for the librarian. As the title suggest the librarian is able to remove books from the database after opening the latter through the "Search Book" functionality.

Parameters: None

Precondition: The librarian must be logged into the system, opened the book database.

Post-condition: The librarian is now able to remove a book from the database by selecting it in the database and pressing "Remove book" under the database display.

Output messages: None.

Triggering:

1. Open the library database with the "Search book" functionality by either searching for a specific book or viewing the whole database by leaving the search field blank.
2. Select a book from the database.
3. Select "Remove book" from under the table.

4.3 Clients

This subsection provides a detailed description of the client mobile application's specific functionalities.

4.3.1 Request book loan

When browsing books, the user has the option to request a book loan. The request is sent to the main server, where it will wait for a librarian's input, whether the loan is allowed or not.

Parameters: ISBN number, Book name

Precondition: The library has at least one physical copy available of the required book in store

Post-condition:

Output messages: The client receives a confirmation of his request being sent successfully

Triggering:

1. User opens the list of available books via the interface button on the bottom of the screen.
2. He then proceeds to click on a book of his choice.
3. In the newly opened screen, the user clicks on the Request loan button.
4. When the user clicks Request loan, the request is sent to the server and it then awaits for further confirmation.

4.3.2 Request book reservation

When browsing books, the user has the option to request a reservation for a book that is currently unavailable. The request is sent to the main server, where it will wait for a copy of the book to be returned back, and then the loan can be confirmed.

Parameters: ISBN number, Book name, Date of reservation

Precondition: The library has no physical copies available for the said book.

Post-condition:

Output messages: The client receives a confirmation of his reservation request being sent successfully.

Triggering:

1. User opens the list of available books via the interface button on the bottom of the screen.
2. He then proceeds to click on a book of his choice.
3. The user can clicks on the Request reservation button on the newly opened screen.
4. When the user clicks Request loan, the request is put in a queue, awaiting for his turn to receive a copy of the book.

4.3.3 Cancel book reservation

Apart from being able to send a book reservation request, the user also has the option to cancel one. The cancellation is effective immediately.

Parameters: Book name

Precondition: The user has sent at least one request for a book reservation.

Post-condition: The reservation request is removed from the queue in the server.

Output messages: The user receives a message saying that his cancelation has been successful and is returned to the reservation-browsing screen.

Triggering:

1. Click on the Browse loans button at the bottom of the main screen.
2. Click on the Reservation button in the top half of the screen.
3. In the list of reservations, click on the book's reservation that is to be cancelled.
4. In the newly opened screen, click on the Cancel reservation button.
5. The user is returned back to the reservation list screen.

Chapter 5

Error messages and problem resolutions

All known problems in using the software should be listed and explained in details using the structure presented below.

Contact information for reporting any problems (either with the software or this document) should be clearly indicated

5.1 Error message 1

5.1.1 Problem identification

A description explaining the meaning of the faced problem.

5.1.2 Probable cause

A description explaining the reasons why such a problem has been raised.

5.1.3 Corrective actions

Describe the required steps the actor should take to recover from such situation.

Appendix A

Title of the appendix 1

Here you write the context of the appendix, structuring such content in sections, sub-sections and sub-sub-sections, if needed.

An example of appendix is the flat presentation of all the graphical user interface screens. Each screen can be presented (identification symbol and description) and screens transition graph can be given.

A.1 My Section

Description of the section.

A.1.1 My subSection

A.1.1.1 My subSubSection

References

1. IEEE: IEEE Standard for Software User Documentation. IEEE Std 1063-2001 (Dec 2001) 1–24
2. Armour, F., Miller, G.: Advanced Use Case Modeling: Software Systems. Addison-Wesley (2001)