Assignment 1

Analysis and Design Document

Digital Library System

(using GAIA methodology)

Project Group 2

Abdelrahman Elnaggar

Reet Ghosh

Sufiyan Bukhari

Sujesh Padhi

Table of Contents

[**1** **System Specifications** 3](#_Toc116665412)

[1.1 Problem Statement 3](#_Toc116665413)

[1.2 System Description 3](#_Toc116665414)

[1.3 Assumptions 4](#_Toc116665415)

[1.4 Requirements 4](#_Toc116665416)

[1.5 Wishlist (Not Implemented) 4](#_Toc116665417)

[**2** **System Analysis** 5](#_Toc116665418)

[2.1 Role Model 5](#_Toc116665419)

[2.2 Role Schema 5](#_Toc116665420)

[2.3 Interaction Model 7](#_Toc116665421)

[**3** **System Design** 8](#_Toc116665422)

[3.1 Agent Model 8](#_Toc116665423)

[3.2 Services Model 9](#_Toc116665424)

[3.3 Acquaintance Model 10](#_Toc116665425)

[**4** **Multi-Agent System Architecture** 11](#_Toc116665426)

[**5** **Agent Description** 12](#_Toc116665427)

[5.1 Admin 12](#_Toc116665428)

[5.2 Librarian 12](#_Toc116665429)

[5.3 Stationer 12](#_Toc116665430)

[5.4 Print 12](#_Toc116665431)

[5.5 Data Manager 12](#_Toc116665432)

# **System Specifications**

## Problem Statement

* The pandemic has boosted digitization of technologies which has led to conversion of physical resources into digital technology that could be accessed using a computer
* Computer can be used to access resources available on the internet which in turn benefits a variety of users including students
* Students have been comfortable taking virtual classes which brings the need to provide the students with materials and books as required for academics
* Unlike pre-pandemic times the students cannot access the library at ease due to social distancing and so the optimum solution could be to bring the library to them over a computer
* A digital library system for the students to extending the library functionality and accessibility would be a convenient solution beyond papers and in-person arrangements

## System Description

* The proposed Digital Library System (DLS) is a multi-agent system designed to provide access to books and various functionalities of a library
* The system also helps to register a new student and de-register an existing student
* The system provides access to millions of books, various stationary items, and access the printer features

INTERNET

Portal/Interface/GUI

Student

Library

Data

Storage

*Print Manager Web Services*

*Stationer*

*Web Services*

*Librarian*

*Web Services*

## Assumptions

* The system maintains the student data, stationary item data and books data
* A Student has two options –
  + A register student can enter their details such as student name, the email-id to access the system
  + An unregistered student can register by entering the student details
* The student can access the admin portal to
  + lend and/or return books
  + purchase stationary items
  + take printouts
* The Librarian takes the book name from the student and
  + checks for the availability in the Data Storage
  + lends the book to the student if available
  + collects the book from the student and updates the Data Manager
* The Stationer takes the stationary item name from the student and
  + checks for the availability in the Data Storage
  + gives the item to the requesting student
* The Print Manager takes the request from the student and
  + checks for the allowed no of free pages for the student in the Data Storage
  + prints the page/s based on the above criterion

## Requirements

* The DLS shall provide access to the list of books in the library
* The DLS shall provide access to the stationary items
* The DLS shall provide the student document printouts
* The DLS shall register, de-register and update the student data
* The DLS shall keep the count of the books and the stationary items updated

## Wishlist (Not Implemented)

* The DLS shall allow the student to login using an SSO
* The DLS shall not allow multiple logins of the same student to avoid the conflict of multiple update requests on the books, the stationary items, and the student data
* The DLS shall have a payments-based system to allow students to take printouts once the free pages count is exhausted
* The DLS should have alert mechanisms to send remainder to the students of the return dates of the borrowed books

# **System Analysis**

## Role Model

We have identified the below roles for our multi-agent Digital Library System –

* Verification
* Registration
* Book Handling
* Stationer
* Printer

## Role Schema

|  |  |  |
| --- | --- | --- |
| Role Schema | | **Verification** |
| Description | | verifies student identity |
| Protocols and Activities | | AuthenticateStudent |
| Permissions | | read studentdata  write studentdata |
| Responsibilities | Liveness | authenticate = (Authenticate, Student) |
| Safety | successful authentication of a student |

|  |  |  |
| --- | --- | --- |
| Role Schema | | **Registration** |
| Description | | Registers/Deregisters a student |
| Protocols and Activities | | RegisterStudent  DeregisterStudent  UpdateStudentData |
| Permissions | | read studentdata |
| Responsibilities | Liveness | register = (Register, Student)  deregister = (De-register, Student)  update = (Update.StudentData, Student) |
| Safety | successful register/de-register a student  keep the data storage updated |

|  |  |  |
| --- | --- | --- |
| Role Schema | | **Book Handling** |
| Description | | handles lending and collecting books with students |
| Protocols and Activities | | LendBook, CollectBook  UpdateBookData |
| Permissions | | read studentdata  read bookdata  write bookdata |
| Responsibilities | Liveness | lending = (Lending.Books, Books)  collecting = (Collecting.Books, Books)  update = (Update.BookData, Book) |
| Safety | lending books based on availability  collecting books and updating the data storage  keep the data storage updated |

|  |  |  |
| --- | --- | --- |
| Role Schema | | **Stationer** |
| Description | | give stationary items |
| Protocols and Activities | | GiveStationary |
| Permissions | | read itemdata  write itemdata  UpdateItemData |
| Responsibilities | Liveness | giving = (Giving.Items, Items)  update = (Update.ItemData, Item) |
| Safety | giving items based on availability  keep the data storage updated |

|  |  |  |
| --- | --- | --- |
| Role Schema | | **Printer** |
| Description | | prints pages for the students |
| Protocols and Activities | | PrintPages |
| Permissions | | read studentdata |
| Responsibilities | Liveness | print = (Print.Pages, Pages) |
| Safety | printing pages based on availability  keep the data storage updated |

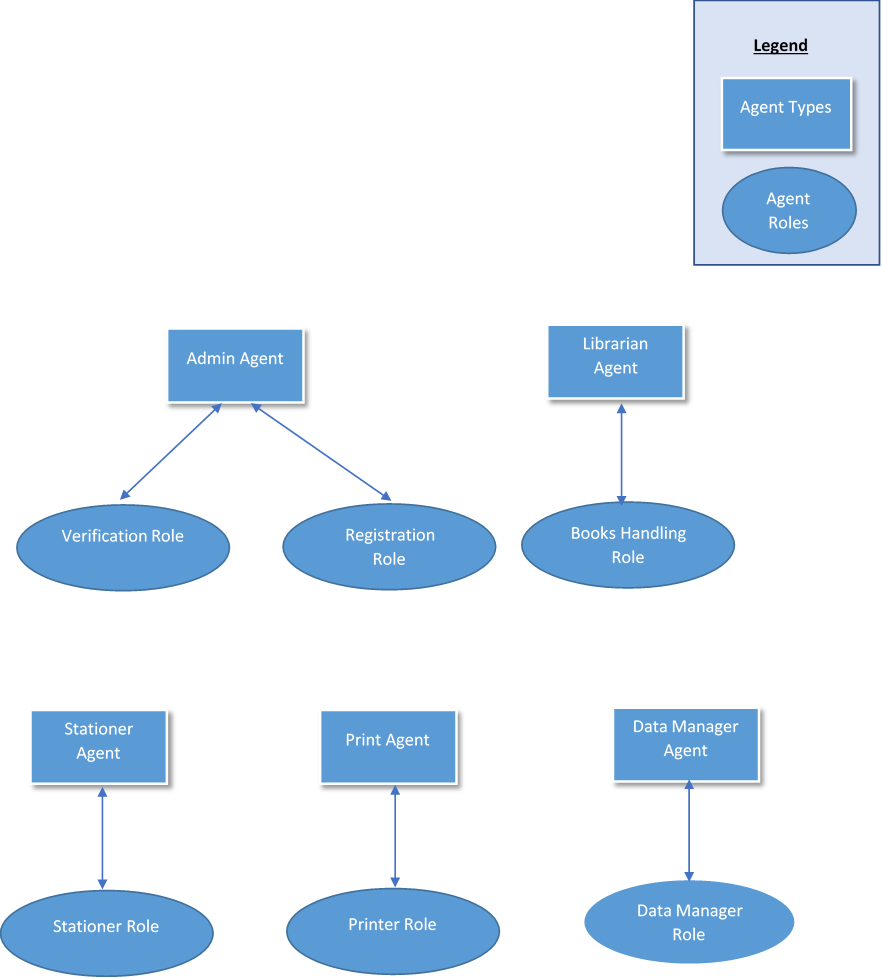
## Interaction Model

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Protocol** | **RegisterStudent** | **DeregisterStudent** | **AuthenticateStudent** | **LendBook** | **CollectBook** |
| Purpose/  Parameters | Register a student | Deregister a student | Authenticate a student | Lending a book | Collecting a returned book |
| Initiator | Student | Student | Student | Book Handling | Book Handling |
| Receiver | Registration | Registration | Verification | Data Manager | Data Manager |
| Responding Protocol | Student registration successful | Student de-registration successful | Student authenticated successfully | Book lent successfully + name of the book | Book returned successfully + name of the book |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Protocol** | **GiveStationary** | **PrintDocuments** | **UpdateStudentData** | **UpdateBookData** | **UpdateItemData** |
| Purpose/  Parameters | Giving stationary items | Print documents requested by the student | Makes required changes to student data (number of books, printouts, stationary) | Makes required changes to book data (updating details of books lent and returned) | Makes required changes to stationery items (updating details of stationery items given and added to data storage) |
| Initiator | Stationer | Printer | Admin | Book Handling | Stationer |
| Receiver | Data Manager | Data Manager | Data Storage | Data Storage | Data Storage |
| Responding Protocol | (Name of item given) + “given successfully!” | Printing successful | Changes updated in Data Storage | Changes updated in Data Storage | Changes updated in Data Storage |

# **System Design**

## Agent Model



Diagram

Description automatically generatedBackground pattern

Description automatically generated with low confidence

+

+

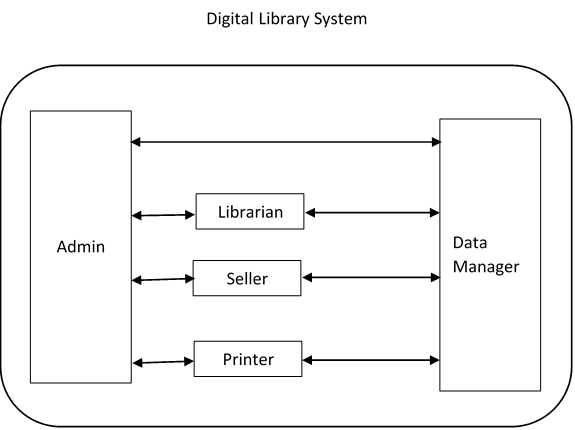
+

+

## Services Model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service** | **Inputs** | **Outputs** | **Pre-conditions** | **Post-conditions** |
| Admin | Student Name | verification, registration, deregistration,  update  notification | Launch the Admin GUI | Send a request to the data storage to verify or update the student data or remove the student data |
| Student ID |
| Librarian | Book name | book lent,  book is not available | Go to the Librarian GUI | Send a request to the data storage to check the book name and its availability |
| book collected |
| update book count |
| Stationer | Item name | item given,  item is not available | Go to the Stationer GUI | Send a request to the data storage to check the stationary item and its availability |
| update item count |
| Print | Number of pages | printout successful,  printout unavailable | Go to the Print manager GUI | Send a request to the data storage to check the allowed number of pages to print for the student |
| update print count |

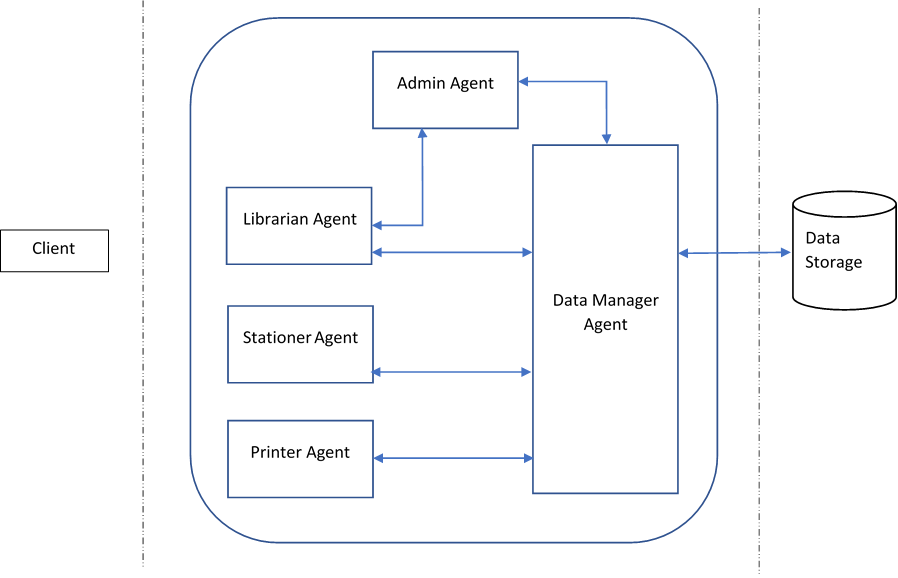
## Acquaintance Model



Stationer

Print

# **Multi-Agent System Architecture**



Print Agent

* The multi-agent system based Digital Library System is envisioned to work between the client and the already existing web-services that are playing distinctive roles
* The Student would be requested to enter specific information on the GUI to allow the agents process their requests
* Agents would pick up the information provided by the student and process the request in the backend to provide a positive or negative response/result
* The Agents communicate internally to keep all the necessary data updated in the storage to offer non-conflicting and seamless service to each student

# **Agent Description**

## Admin

* The Admin agent talks to the client through the portal or the web interface. It is the primary point of correspondence with the client (the student) and handles all the tasks that require interaction with the client
* The Admin agent talks to the Librarian, the Stationer, and the Print agents to lead the processes of lending/collection library books, giving stationary items and adding items to the data storage, and fulfilling the process of printing documents requested by the client
* Finally, the Admin agent can communicate directly with the Data Manager. This direct communication with the Data Manager is made possible to have a single point of control for all the data handled by the system

## Librarian

* As the name suggests, the Librarian agent takes care of all the duties a librarian in a real-world library would do; this agent handles the tasks of lending books to students, as well as collecting book returned by students
* The Librarian agent talks to the Admin agent to receive details about the books (to be lent or received)
* The Librarian agent in turn talks to the Data Manager, to search for books and keep track of what books were lent and if a book requested by a client is available or not

## Stationer

* The Stationer agent is responsible for giving items from the available stationery items data storage. This is made possible by having the Stationer agent talk to the Data Manager agent, so that the remaining items in the Data Storage can be searched through. Once the search is completed, the items can be given as per the request made by the client.
* The request to the Stationer agent is made through the Admin agent. This means that the Admin agent first receives the request, and then this request is relayed to the Stationer agent.

## Print

* The Print agent takes care of all requests pertaining to printing pages. These requests are also relayed to the Print agent by the Admin agent (who initially receives them from the clients).
* The Print agent then communicates with the Data Manager agent to check the status of the client’s print quota. Once this verification is complete, the printing task is completed, and the print status is shown to the client.

## Data Manager

* The Data Manager agent talks to the Data Storage and updates the Data Storage as requested by the other agents. The Data Manager agent is constantly in touch with the Librarian, the Stationer, the Print, and the Admin agents. The single point of contact with the Data Storage is made to ensure security and prevent concurrent Data Storage accesses.
* The Data Manager agent is the main point of contact with the data storage used by the system. Due to this reason, the Data Manager agent’s tasks include the following:
  + Adding and removing from the count of a particular book, once it has been lent or returned
  + Adding and removing student names and IDs from the Data Storage once the student has been registered or deregistered
  + Subtracting items from the storage once they have been given