***SYSTEMS ANALYSIS AND DESIGN***

# GUIDE TEMPLATE

***Systems Analysis & Design Guide***

***System Analysis & Design Report***

## *1 SYSTEM ANALYSIS AND DESIGN REPORT*

1. *The System Analysis and Design (“SA&D”) phase signifies the commencement of system implementation. The objectives of this phase are:*

*i) to investigate and understand the user and technical requirements; ii) to specify and design the new system; and iii) to detail the implementation requirements in terms of cost, effort and time.*

1. *The SA&D Report will be produced to document the findings and recommendations of this phase.*
2. *A sample template of the SA&D Report with sample content is provided in the following pages.*
3. *Notes for using the template are written in “*italic*” text enclosed in pointed brackets “< >”, while sample contents are written in “***bold italic***” and can be replaced by project-specific information or removed to suit specific project needs. After all changes are made, all notes should be removed and font of all “*italic*” text should be changed to black.*

***SYSTEM ANALYSIS AND DESIGN REPORT***

# FOR

**MelodyMaker**

# OF

## Team 3

### *2018 05 08*

### Table of Contents

1. ***MANAGEMENT SUMMARY ............................................................................. 1*** *1.1 APPROVAL SOUGHT .................................................................................... 1* 
   1. *SYSTEM OBJECTIVES .................................................................................. 1*
   2. *BACKGROUND .............................................................................................. 1* 
      1. *Current System Description .................................................................... 1*
   3. *PROPOSED SYSTEM ..................................................................................... 2* 
      1. *System Overview .................................................................................... 2*
      2. *System Functions .................................................................................... 2*
      3. *Technical System Architecture ................................................................ 3*
   4. *RESOURCE IMPLICATIONS ........................................................................ 3*
   5. *COSTS AND BENEFITS ................................................................................ 4* 
      1. *Costs ........................................................................................................ 4*
      2. *Benefits .................................................................................................... 4*
   6. *IMPLEMENTATION PLAN ............................................................................ 5*
2. ***CURRENT ENVIRONMENT DESCRIPTION ................................................. 6*** *2.1 CURRENT SYSTEM DESCRIPTION ............................................................ 6* 
   * 1. *Current System Overview ....................................................................... 6*
     2. *Current System Hardware, Software and Network ................................. 6*
     3. *Volumes and Frequencies ........................................................................ 7*
     4. *Interface with Other Systems .................................................................. 7*
   1. *CURRENT BUSINESS PROCESS ................................................................. 8*
   2. *CURRENT PROBLEMS AND ISSUES ......................................................... 9*

#### 3REQUIREMENTS SPECIFICATION .............................................................. 10 **3.1 USER REQUIREMENTS DOCUMENT ...................................................... 10**

*3.1.1 Proposed System Overview ................................................................... 10*

*3.1.1.1 Description of Proposed* Library Management System *............. 10*

*3.1.1.2 System User Profile ................................................................... 10*

*3.1.2 Future Business Process .........................................................................11*

*3.1.2.1 List of Future Business Processes ..............................................11 3.1.2.2 BP-001 Borrow book .................................................................11*

*3.1.2.3 BP-002 Reserve Book ............................................................... 13*

*3.1.3 Functional Requirements ....................................................................... 13*

*3.1.3.1 List of Functional Requirements ............................................... 13 3.1.3.2 REQ-SYS-001 Login ................................................................ 13 3.1.3.3 REQ-BOR-001 Borrow book .................................................... 14*

*3.1.3.4 REQ-BOR-002 Check availability ............................................ 14*

*3.1.4 Non-functional Requirements ............................................................... 15*

*3.1.4.1 List of Non-functional Requirements ........................................ 15*

*3.1.4.2 REQ-ACS1 System Audit ......................................................... 15*

*3.1.4.3 REQ-ACS2 System Control ...................................................... 15*

*3.2 TECHNICAL REQUIREMENTS .................................................................. 16*

*3.2.1 List of Technical Requirements ............................................................. 16 3.2.2 TR-SBR-001 Server House Keeping .................................................... 17*

*3.2.3 TR-SBR-002 Backup, Recovery and System Archive .......................... 17*

#### 4SYSTEM SPECIFICATION .............................................................................. 18 **4.1 FUNCTIONAL SPECIFICATION ................................................................ 18**

*4.1.1 Required System Overview ................................................................... 18*

*4.1.2 Function Definition ............................................................................... 18*

*4.1.2.1 UF-0001 Borrow book .............................................................. 19*

*4.1.2.2 UF-0002 Check availability ...................................................... 20*

*4.2 ARCHITECTURE DESIGN .......................................................................... 21*

*4.2.1 Application Architecture ....................................................................... 21*

*4.2.1.1 Application ................................................................................ 21*

*4.2.1.2 Security ...................................................................................... 23*

*4.2.1.3 Integration ................................................................................. 24*

*4.2.2 Data Architecture ................................................................................... 25*

*4.2.2.1 Data Source ............................................................................... 25*

*4.2.2.2 Data Store Architecture ............................................................. 25*

*4.2.2.3 Data Retention and Archive ...................................................... 26*

*4.2.2.4 Data Conversion Architecture ................................................... 26*

*4.3 SYSTEM DESIGN ......................................................................................... 28*

*4.3.1 Application ............................................................................................ 28*

*4.3.1.1 Design Application .................................................................... 29*

*4.3.1.2 Design Integration ..................................................................... 35*

*4.3.1.3 Design Data Conversion ............................................................ 35*

*4.3.1.4 User Experience Design ............................................................ 36*

*4.3.2 Data Model ............................................................................................ 39*

*4.3.2.1 Logical Data Model ................................................................... 39*

*4.3.2.2 Physical Data Model ................................................................. 40*

#### 5TECHNICAL SYSTEM OPTIONS .............................................................. 42 **5.1 TECHNICAL SYSTEM ARCHITECTURE .................................................... 42**

*5.1.1 Network Architecture ............................................................................ 42*

*5.1.2 Storage Architecture .............................................................................. 43*

*5.1.3 Platform Architecture ............................................................................ 43*

*5.2 SIZING MODEL ............................................................................................ 44*

*5.3 COST / BENEFIT EVALUATION ................................................................ 48*

*5.4 IMPACT ANALYSIS ..................................................................................... 49*

*5.5 IMPLEMENTATION PLAN .......................................................................... 52*

# 1 MANAGEMENT SUMMARY (NEWLY ADDED SECTION)

# 1.1 APPROVAL SOUGHT

This SAD document aims to seek approval from Belgium Campus to proceed to the next stage of system implementation and Integration. We as a team believe that our project has massive potential value for the field of music.

##### 1.2 SYSTEM OBJECTIVES

Our main objectives are:

* Create a test and training dataset to use as input for the teaching of our artificial neural network.
* Create a database that stores paths/storage locations of different music genres of music in .midi format.
* Create a melody focused artificial neural network that uses the database as input for its unsupervised learning.
* Create a new way of producing musical melodies.

##### 1.3 BACKGROUND

Reasons for Project

* This project does not serve any business-related reason for being, it is purely an attempt to create new material for the musical field and spur on the imagination of composers and musicians worldwide. We might find a new way of looking at the creation of a melody, as our project aims to develop melodies that are pleasing to the ear. E.g. to focus on using strong semitones between notes of a melody.

1.3.1 Current System Description

*Currently the only way of composing music is by imagination.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User** |  | **Channel** |  | **Imagination** |  | **External System** |

**Using imagination to create music**



Find a combination of notes that sound pleasing



Add bassline + drumbeat



Record song or compose song in a music library like Cubase

Music Composer

Record Label

Give song to public

##### 1.4 PROPOSED SYSTEM

<Describe the basic information of proposed system functions which can deliver value to the business needs according to the collected system requirements.>

###### 1.4.1 System Overview

**Sample:**

**The proposed IT system will enable Users to sign into the library system using their mobile application, and where they can search for their desired book and immediately see if there is any stock available. If the book is available and not reserved, and the User has no outstanding fines, the librarian will issue the book to the User…**

*Login*

*Search book*

*Add book*

*Reserve book*

*Delete book*

*Return book*

*Borrow book*

*Update book*

*Check Account*

*Issue book*



*User*



*Librarian*

###### 1.4.2 System Functions

**Sample:**

**The proposed system will cover the following major function areas:**

|  |  |
| --- | --- |
| ***Item*** | ***System Functions*** |
| **1** | **Common requirements for Library Management System** |
| **1.1** | **The general functional requirements for features in the Library System which includes...** |
| **2** | **Book Management System** |
| **2.1** | **Book record maintenance** |
| **2.2** | **House Keeping** |
| **3** | **Borrow Book System** |
| **3.1** | **…** |

###### 1.4.3 Technical System Architecture

**Sample:**

**The system will reside at a virtual environment under the Central Computer Centre Virtualized**

**Infrastructure (CCCVI) which is supported by the Office of the Government Chief Information**

**Officer (OGCIO) operation team, and will make use of the existing components of…**

##### 1.5 RESOURCE IMPLICATIONS

<Describe the resource required including staff effort in term of man-days or cost.> **Sample:**

**The estimated staff resources required in man-days to develop the system and to provide ongoing system maintenance are summarized below:**

**Detailed breakdown for System Implementation & Integration (SI&I) services in man-days:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Stage*** | ***Program***  ***Manager*** | ***Business***  ***Analyst*** | ***System***  ***Analyst*** | ***Analyst***  ***Programmer*** | ***UX***  ***Designer*** |
| **Stage 2 - Physical**  **Design and Program Development** | **30** | **60** | **80** | **120** | **30** |
| **Stage 3 -**  **Acceptance Testing** | **10** | **30** | **60** | **30** | **10** |
| **Stage 4 - Data Conversion** | **10** | **10** | **30** | **60** | **0** |
| **Stage 5 -**  **Documentation and Rollout** | **20** | **20** | **80** | **80** | **0** |
| **Stage 6 - Nursing** | **5** | **0** | **10** | **10** | **0** |
| **Stage 7 –**  **…** | **5** | **...** | **…** | **…** | **…** |
| **Total** | **80** | **…** | **…** | **…** | **…** |

##### 1.6 COSTS AND BENEFITS

|  |
| --- |
| ***Benefits***   * No real cost except for the Udemy course fee * No Annual Cost |
| ***Development Cost: R2400***   * Udemy Courses: R2400.00 |
| ***Operational Cost: R0.00***   * Hardware: R0.00 * Software: R0.00 * Operational: R0.0 |
| ***Total Cost: R2400.00*** |

###### 1.6.1 Costs

There is no cost in creating a neural network nor storing it on a Github repository, thus after development no more capital in needed.

|  |  |
| --- | --- |
| ***Non-Recurrent Cost*** | ***Cost (R)*** |
| **Hardware & Software** | **2400.00** |
| **Service implementation** | **0 .00** |
| **IT-Staff** | **0 .00** |
| **Total** | **2400 .00** |

###### 1.6.2 Benefits

Our project holds value for the field of music in a way that is unmeasurable. We seek not profit, but new ways of looking at music and a better understanding of why certain combinations of notes sounds more pleasing to the ear than others.

##### 1.7 5.5 IMPLEMENTATION PLAN SUMMARY

** Implementation Strategy

1. **Prepare the infrastructure.**
2. **Coordinate with the organizations involved in implementation.**
3. **Install the production solution.**
4. **Convert the data**
5. **Perform final verification in production**
6. **Implement new processes and procedures.**
7. **Monitor the solution.**

** Implementation Schedule

|  |  |  |
| --- | --- | --- |
| **Deliverable** | **Description** | **Planned Date** |
| Milestone 1  Final project proposal document submission | This includes the final proposal that will be assessed before commencement of the project. It is the first item of the development process that will be used to guide all other items on a basic level. | 2018-03-21 |
| Milestone 2  Planning document submission | Much like the proposal, the planning document will be used to guide the development of all other items, however, it will be used on a more in-depth scale. It provides guides and steps in a predefined manner that allows the configuration of all items to be executed as harmoniously as possible. | 2018-04-13 |
| Milestone 3  System Analysis and Design document submission | Here we identify, break down and assess all major and minor aspects of the development of the system, the environment, effects and the functionality of the system itself. | 2018-05-04 |
| Milestone 4  Technical 1: Database Design | This is technical aspect of the system. The database is one of the most important physical components of the system. Once this has been correctly developed, it can also be used for Class creation which is the next item. | 2018-05-25 |
| Milestone 5  Technical 2: Class Design | This item can be based on the elements found in the database design. It is a technical item that defines the basis of code development of the system. | 2018-06-14 |
| Milestone 6  Technical 3 : Object Behaviour Model | This technical item works with thee classes and database as a baseline. However, it is also the baseline for coding the system. It will be assessed and defined before coding commences for the purpose of simplifying the intense complicated nature of coding. | 2018-07-04 |
| Milestone 7  Technical 4 : Coding | This is the item that configures all previous items. It is the physical compilation of all planning up until this point. The system is physically created here. | 2018-08-04 |
| Test plan document submission | Here we will identify our approach to testing the application. | 2018-09-14 |
| Milestone 8  Testing | This is where we will implement our testing strategy that we have identified in the test plan document for our application. | 2018-10-22 |
| Milestone 9  Implementation | This is where the implementation process of our application will be executed. | 2018-10-29 |
| Project Submission | The project is in final form and must be ready to be summited on the given date. | 2018-11-01 |

*Current Environment Description*

*Current system description*

*Currently the only way of composing music is by imagination.*

*Current System Overview*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User** |  | **Channel** |  | **Imagination** |  | **External System** |

**Using imagination to compose music**



Find a combination of notes that sound pleasing



Add bassline + drumbeat



Record song or compose song in a music library like Cubase

Music Composer

Record label

Give song to public

*High level system overview*

|  |  |
| --- | --- |
| **Subsystem** | **Function** |
| **<List the subsystems>** | **<List the functions>** |
| **Using imagination to compose music** | * Find a pleasing combination of notes. * Add bassline + drumbeat * Record or compose song in a music library such as Cubase |

*Current System Hardware, Software and Network*

*Minimum Cubase requirements*

|  |
| --- |
| Windows 7, 8, 10 |
| 64-Bit Windows 7 / 8.x / 10 | |
| 64-bit Intel or AMD multi-core processor (Intel i5 or faster recommended) | |
| 4 GB RAM (8 GB or more recommended) | |
| 18 GB free HD space | |
| 1366 x 768 display resolution (1920 x 1080 recommended) | |
| Graphics card with DirectX 10 and WDDM 1.1 support (Windows only) | |
| USB port for USB-eLicenser (license management) | |
| OS-compatible audio hardware\* | |
| An internet connection is required for activation, account setup and personal/product registration. Additional downloads may be required for the installation. | |

*No other software is required to compose music. An instrument or two might help.*

*CURRENT PROBLEMS AND ISSUES IN COMPOSING MUSIC*

* *Composing good melodic music takes time and experience.*
* *Even composers can run out of ideas for new melodies.*

*CURRENT PROBLEMS AND ISSUES IN DEVELOPMENT*

|  |  |
| --- | --- |
| No. | Description |
| <Unique number> | <Description> |
| 1 | Sorting our .csv files into a database. |
| 2 | Converting .midi to .csv without doing it manually. |
| 3 | Visualizing the optimal output via MathLib – a python addon |

#### 3 REQUIREMENTS SPECIFICATION

##### 3.1 USER REQUIREMENTS DOCUMENT

###### 3.1.1 Proposed System Overview

<This section provides a brief description about the proposed IT system to be developed by presenting a high-level conceptual model of the system and showing a system user profile about the users of the proposed IT system that will be referred to in following sections.> **Sample:**

**The proposed IT system will enable Users to sign into the library system using their mobile application, and where they can search for their desired book and immediately see if there is any stock available. If the book is available and not reserved, and the User has no outstanding fines, the librarian will issue the book to the User…**

##### Description of Proposed Library Management System

<Include an overview of the whole business function.> <Diagram for overview of the whole business function.> **Sample:**

*Login*

*Search book*

*Add book*

*Reserve book*

*Delete book*

*Return book*

*Borrow book*

*Update book*

*Check Account*

*Issue book*



*User*



*Librarian*

##### System User Profile

<The following provides a table of external and internal users of the proposed IT system. Each user will have a role in the proposed IT system as shown in the circles in the above context diagram, and mapped to a user type in the table below.> System User Profile:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***No.*** | ***User Role*** | ***Responsibilities*** | ***Branch/***  ***Division/***  ***Section/ Unit*** | ***Staff***  ***Post/Rank*** | ***Stakeholder Group*** |
| **1** | **Librarian** | **Responsible for overseeing the entire** | **Administration** | **Librarian** | **Library stakeholders** |
| ***No.*** | ***User Role*** | ***Responsibilities*** | ***Branch/***  ***Division/***  ***Section/ Unit*** | ***Staff***  ***Post/Rank*** | ***Stakeholder Group*** |
|  |  | **system’s operation and use** |  |  |  |
| **2** | **XX** | **Responsible for xx** | **XX Section** | **XX or equivalent ranks** | **Library stakeholders** |
| **3** | **XX** | **Responsible for xx** | **XX Section** | **XX or equivalent ranks** | **Library stakeholders** |

###### 3.1.2 Future Business Process

<For each future business process, include a separate business process diagram. Provide an overview of functional requirements. For complex business process, breakdown the process into smaller processes, each with a separate diagram.>

##### List of Future Business Processes

<The following table provides a list of future business process flows for the system.> List of Future Business Processes:

|  |  |
| --- | --- |
| ***Process ID*** | ***Business Process Title*** |
| **BP-001** | **Borrow book** |
| **BP-002** | **Reserve book** |
| **BP-003** | **Return book** |
| **BP-004** | **Search book** |

##### BP-001 Borrow book

<Diagram(s) for future business processes.> **Sample:**

***Borrow book***

*User*

*Librarian*

*Library System*

*Start*

*Sign into*

*Library*

*system*

*Issue*

*book*

*Check*

*availability*

*(*

*Not Available*

*)*

*(*

*Available*

*)*

*End*

*Notify*

*User*

*Borrow*

*book*

*E*

*-*

*Book*

*(*

*Is not e*

*-*

*book*

*)*

*(*

*Is e*

*-*

*book*

*)*

*Scan QR*

*code*

<The major elements that need to be captured in the Process Diagrams are:

* Event
  1. Denote something that happens
* Activity
  1. Describe the kind of work that needs to be performed ** Gateway

*○* Describe the decision factors and determines where the next step should go ** Connections

*○* Describe how the process flow objects are connected >

Narratives:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Task No.*** | ***Actor*** | ***Task Name and Description*** | ***Input*** | ***Output*** |
| <Input the task no.> | <Input actor> | <Input task name and description> | <Input> | <Output> |
| **1** | **User** | **Sign into Library system – User signs into Library system without need to display library card to Librarian** | **Task step: Provide library card number and password** | **Task step:**  **Access Library system** |
| **2** | **User** | **Borrow book - User enters requests to borrow a book into Library system** | **Task step:**  **Borrow book** | **Task step: Check**  **availability** |
| **3** | **Library System** | **Check availability - Check if book has been reserved or is being borrowed in system records** | **Task step:**  **Issue book** | **Task step: Locate book and distribute to User** |

Other information:

|  |
| --- |
| ***Glossary*** |
| ***References*** |
| <Input references> |
| **N/A** |
| ***Assumptions:*** |
| <Input assumption> |
| 1. **It is assumed that the Librarian can access the Departmental Portal to fulfill their tasks.** 2. **It is assumed that the User is in possession of their library card number and password to sign into the library system.** 3. **E-book can only be viewed inside the library.** |
| ***Business Rules*** |
| <Input business rules> |
| 1. **Library system cannot give approval to Librarian if book is reserved by another User.** 2. **A book can be reserved by a maximum of 3 people at any particular time.** 3. **Any outstanding fines the User must have been paid in full before he/she is allowed to borrow any books.** |

##### BP-002 Reserve Book

<Diagram(s) for future business processes.>

…

###### 3.1.3 Functional Requirements

<State the Functional Requirements in this section in numbered tables or paragraphs by grouping them according to business nature or types of requirements and assigned with a unique requirement number, e.g. REQ- CRE-000, 001, 002, 003, etc. for ease of reference.>

##### List of Functional Requirements

<All functional requirements of the proposed IT system should be listed in the following table and then explained in detail one by one. Each requirement is assigned with a priority to indicate its importance, e.g. MUST (M), SHOULD (S), COULD (C) and WON’T (W). B/Ds may assign priorities using other ranking, e.g. High, Medium and Low.> List of Functional Requirements:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Req. ID*** | ***Requirement Title*** | ***Target Users*** | ***Priority*** |
| **REQ-SYS-001** | **Login** | **User/Librarian** | **M** |
| **…** | **…** | **…** | **…** |
| **REQ-BOR-001** | **Borrow book** | **User** | **M** |
| **REQ-BOR-002** | **Check availability** | **User** | **M** |
| **REQ-BOR-003** | **Notify User** | **User** | **M** |
| **…** | **…** | **…** | **…** |

##### REQ-SYS-001 Login (Newly Added Section)

User case specification:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Use Case ID:*** | **UC01** | | | |  |
| ***Use Case Name:*** | **Login** | | ***Version No:*** | | **01.002** |
| ***Purpose:*** | **User/Librarian login the system** | | | |  |
| ***Last Update by:*** | **Hamilton** | | ***On (date):*** | | **04 April 2016** |
| ***Approved by:*** | **Vettel** | | ***On (date):*** | | **05 April 2016** |
| ***User/Actor:*** | **User/Librarian** | | | |  |
| ***Business Owner Name:*** | **Alonso** | | | |  |
| ***Trigger:*** | **User or Librarian access the system protected login page** | | | |  |
| ***References:*** | **N/A** | | | |  |
| ***Frequency of Use:*** | **Daily** | | | |  |
| ***Volume of Use*** | **N/A** | | | |  |
| ***Preconditions:*** | **N/A** | | | |  |
| ***Post Conditions:*** | **N/A** | | | |  |
| ***Non-functional***  ***Requirements*** | **N/A** | | |  | |
| ***Assumptions:*** | **N/A** | | |  | |
| ***Business Rules:*** | **N/A** | | |  | |
|  | ***Main Flow 1*** | | |  | |
| ***Step*** | ***Actor*** | ***Description*** | | ***Alternate/Exception Flow*** | |
| ***1*** | **User/Libraria n** | **enters the ID and password** | |  | |
| ***2*** |  | **Process the authentication: if the ID and password are correct, the system authenticates the user and displays the relevant functions page. The use case ends.** | |  | |
| ***3*** |  | **if any or both of ID and password is/are invalid, the system directs user/librarian to alternate flow #1.1** | | **#1.1** | |
| ***Alternate/Excepti on Flow*** | **Alternate Flow#1.1** | | |  | |
| ***Step*** | ***Actor*** | ***Description*** | | ***Alternate /Exception Flow*** | |
| ***1*** |  | **The system prompts the error message. The use case ends** | |  | |

##### REQ-BOR-001 Borrow book

Requirements Description:

|  |  |
| --- | --- |
| ***Item*** | ***Description*** |
| Requirement ID | **REQ-BOR-001** |
| Requirement Title | **User selects book to borrow from library online** |
| Priority | **Must** |
| Functional Requirement Description | * **The User shall be able to create a request for borrowing a book in the library system after signing into the Library system.** * **If a book is wrongly selected, a remove option should be provided for removal of the wrongly selected book.** * **…** |
| Frequency of Use | **Daily** |
| Acceptance Criteria | **Same User ID must only select up to 3 books to borrow.** |
| Related Business Process | **Refer to BP-001.** |

##### REQ-BOR-002 Check availability

Requirements Description:

…

###### 3.1.4 Non-functional Requirements

<State the Non-Functional Requirements for the non-functional features such as audit, control and security, global business rules, data requirements, usability requirements, service level targets, user volume and equipment requirements, data growth and retention requirements, etc. that the proposed IT system must possess from a business perspective. The following proposed non-functional requirements can be changed or removed to suit project needs.>

##### List of Non-functional Requirements

<Provide a list of non-functional requirements.> List of Non-functional Requirements:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Req. ID*** | ***Category*** | ***Requirement Title*** | ***Target Users*** | ***Priority*** |
| **REQ-ACS1** | **Audit , Control & Security** | **System Audit** | **Librarian** | **M** |
| **REQ-ACS2** | **Audit , Control & Security** | **System Control** | **Librarian** | **M** |
| **REQ-ACS3** | **Audit , Control & Security** | **System Security** | **Librarian** | **M** |
| **REQ-ACS4** | **Audit , Control & Security** | **Backup and Recovery Requirements** | **Asst. Librarian** | **M** |
| **REQ-ACS5** | **Audit , Control & Security** | **Disaster Recovery Requirements** | **Asst. Librarian** | **M** |
| **REQ-GBR1** | **Global Business Rules** | **Global Business Rules** | **Librarian** | **M** |
| **REQ-DAR1** | **Data Requirements** | **Key Data Requirements** | **Librarian** | **M** |
| **REQ-USR1** | **Usability** | **General Usability Requirements** | **Librarian** | **M** |
| **REQ-SLT1** | **Service Level Targets** | **System Availability** | **Asst. Librarian** | **M** |
| **REQ-SLT2** | **Service Level Targets** | **System Performance** | **Asst. Librarian** | **M** |
| **REQ-DGR1** | **Data Growth and**  **Retention Requirements** | **Data Growth and Retention Requirements** | **Asst. Librarian** | **M** |
| **REQ-UER1** | **Number of Users & IT Equipment Requirement** | **Number of Users & IT Equipment Requirement** | **Asst. Librarian** | **M** |

##### REQ-ACS1 System Audit

Requirements Description:

|  |  |
| --- | --- |
| ***Item*** | ***Description*** |
| Requirement ID | **REQ-ACS3** |
| Category | **Audit , Control & Security** |
| Requirement title | **System Security** |
| Priority | **Must (except for those specially specified)** |
| Non-functional requirement description | 1. **All user passwords must not be displayed on screen during user input.**  **All security patches should be properly tested before installed to all system software/programs before production roll out.**  **…** |

##### REQ-ACS2 System Control

Requirements Description: ….

##### 3.2 TECHNICAL REQUIREMENTS

<State the Technical Requirements in this section in numbered tables or paragraphs by grouping them according to the type of requirement and assigned with a unique requirement number, e.g. TR-SBR-000, 001, 002, 003, etc. for ease of reference.>

###### 3.2.1 List of Technical Requirements

<Provide a list of technical requirements. The technical requirements supplement the nonfunctional requirements and are not raised by the Business Analyst, but from the IT project member(s) who raise(s) the technical requirements and is/are responsible for the acceptance of the requirements in UAT.>

List of Technical Requirements:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Req. ID*** | ***Requirement Title*** | ***Priority*** | ***Category*** | ***Responsible***  ***Team***  ***Member(s)*** |
| **TR-SBR-001** | **Server House Keeping** | **M** | **System Backup and Recovery Requirements** | **Systems**  **Analyst(s)/Project Manager(s)** |
| **TR-SBR-002** | **Backup, Recovery and System Archive** | **M** | **System Backup and Recovery Requirements** | **Systems**  **Analyst(s)/Project Manager(s)** |
| **TR-QRC-001** | **QR Code Standards** | **M** | **QR Code requirements to follow ISO/IEC 18004:2006 standards** | **Systems**  **Analyst(s)/Project Manager(s)** |
| **TR-DRR-001** | **….** | **….** | **Disaster Recovery Requirements** | **….** |

<The elements to be captured and documented for the Technical Requirements are: ** Requirement ID

*○* Specify a unique ID for each requirement entry.

* Requirement Title
  1. Title for the technical requirement. ** Priority

*○* State the priority of the non-functional requirement, e.g. “Must”, “Should”, “Could” and

“Won’t” or other ranking scheme e.g. “essential”, “beneficial if cost justified”, “subsequent enhancement”.

*○* Possibly have more level of classifications depending on the project situation

* Technical Requirements Description
  1. Describe the technical requirement of the system, i.e., “how” the system should work. ** Category (Examples)

*○* System Backup and Recovery Requirements

* + - Backup arrangements
    - Recovery procedures requirement under various system failures

*○* Disaster Recovery Requirements

* + - Minimum service level under disaster
    - Off-site backup arrangement
    - Recovery procedure
    - Time required to recover upon disaster

*○* Privacy Requirements

* + - Protection of personal data from unauthorised disclosure e.g. protection on personal identification document number.

*○* Technical Support Requirements

* + - Software and hardware support levels
    - Equipment maintenance and repair cycles
    - Test/diagnostic equipment

*○* Interface Requirements

* + - User groups
    - Content presentation
    - Application navigation

*○* Maintainability, Control and System Management Requirements

* + - System failure(s)
    - Operational readiness and success
    - System effectiveness evaluation and improvement

*○* Testing

* + - Design stage testing procedure

*○* Data Conversion

* + - Data conversion process
    - Data cleansing
    - Verification program

*○* User Experience

* + - Overall experience and satisfaction when a user is using a product or system
    - Details within the user interface functionality, behaviour, and design **Industry best practices as part of standard requirements.>

###### 3.2.2 TR-SBR-001 Server House Keeping

Technical requirements description:

|  |  |
| --- | --- |
| ***Item*** | ***Description*** |
| Requirement ID | **TR-SBR-001** |
| Requirement title | **Server House Keeping** |
| Priority | **Low** |
| Category | **System Backup and Recovery Requirements** |
| Technical requirement description | **System logs must be archived to backup tape weekly** |

***3.2.3 TR-SBR-002 Backup, Recovery and System Archive***

Technical requirements description:…

# **SYSTEM SPECIFICATION**

## FUNCTIONAL SPECIFICATION

### Required System Overview

|  |  |  |
| --- | --- | --- |
| *Business Needs* | *Major Features* | *System Related Functions* |
| *Music Pattern Learning* | *Learn melody patterns from provided sheet music and compose music from what the application has learned* | *Sheet Music Reading, Analysis, Analysis Storage, Music Composition* |

#### MMFI-01 Read Midi

|  |  |
| --- | --- |
| *Item* | *Description* |
| *Function ID* | *MMFI-01* |
| *Function Name* | *Read Midi* |
| *Category* | *Data Access layer* |
| *Function Description* | *This function reads the midi file and extracts the data stored in the file for later analysis* |
| *Mode* | *batch* |
| *Frequency* | *When invoked* |
| *Special Service Level Requirements* | *N/A* |
| *Data Integration and Conversions* | *N/A* |
| *Business Rules* | *Check if valid midi file* |

#### MMFI-02 Analyse Data

|  |  |
| --- | --- |
| *Item* | *Description* |
| *Function ID* | *MMFI-02* |
| *Function Name* | *Analyse Data* |
| *Category* | *Business Layer* |
| *Function Description* | *This Function takes the raw data that was gathered in the function ‘Read Midi” and does an analysis of the pattern and creates a neural mapping or updates the existing one.* |
| *Mode* | *Batch* |
| *Frequency* | *When invoked* |
| *Special Service Level Requirements* | *N/A* |
| *Data Integration and Conversions* | *N/A* |
| *Business Rules* | *If a neural mapping exists the application should use it and add what it learns from new sheet music to improve on what it has learned from previous experiences.* |

#### MMFI-03 Compose Music

|  |  |
| --- | --- |
| *Item* | *Description* |
| *Function ID* | *MMFI-03* |
| *Function Name* | *Compose Music* |
| *Category* | *Business Layer* |
| *Function Description* | *This function will make use of the neural network to compose sheet music in a chosen key and create a midi file with the sheet music.* |
| *Mode* | *batch* |
| *Frequency* | *When invoked* |
| *Special Service Level Requirements* | *N/A* |
| *Data Integration and Conversions* | *N/A* |
| *Business Rules* |  |

## ARCHITECTURE DESIGN

## Application Architecture

### Security

*Our neural network is opensource and freely available on Github, thus no security threats are applicable to this project.*

*The availability of the information of the system will be given according to the roles the people play in the actual development of the system.*

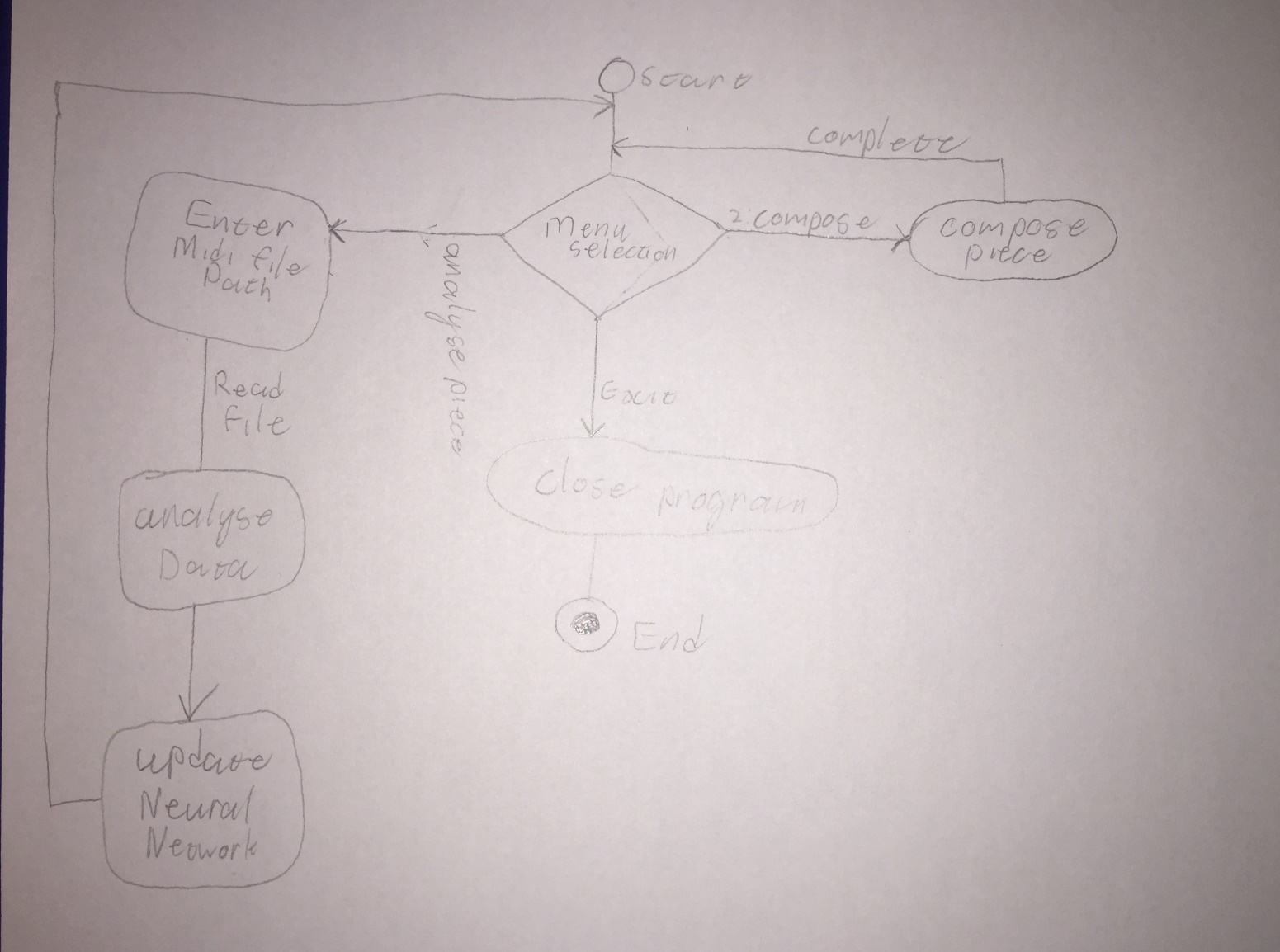
*The artificial neural network voltage security monitoring and control is used. The neural network uses its association mechanism, the inherent parallel information processing nature of the neural network, which provides the capability of fast computation, enables the neural network approach to meet the demands of real time monitoring control*

## SYSTEM DESIGN

### Common Frameworks

|  |  |
| --- | --- |
| *Validation* | * *Validate midi input* |
| *Exception handling* | * *Throw error when invalid input* * *Throw error when invalid file input* * *Throw error when a file cannot be composed* |

## Design

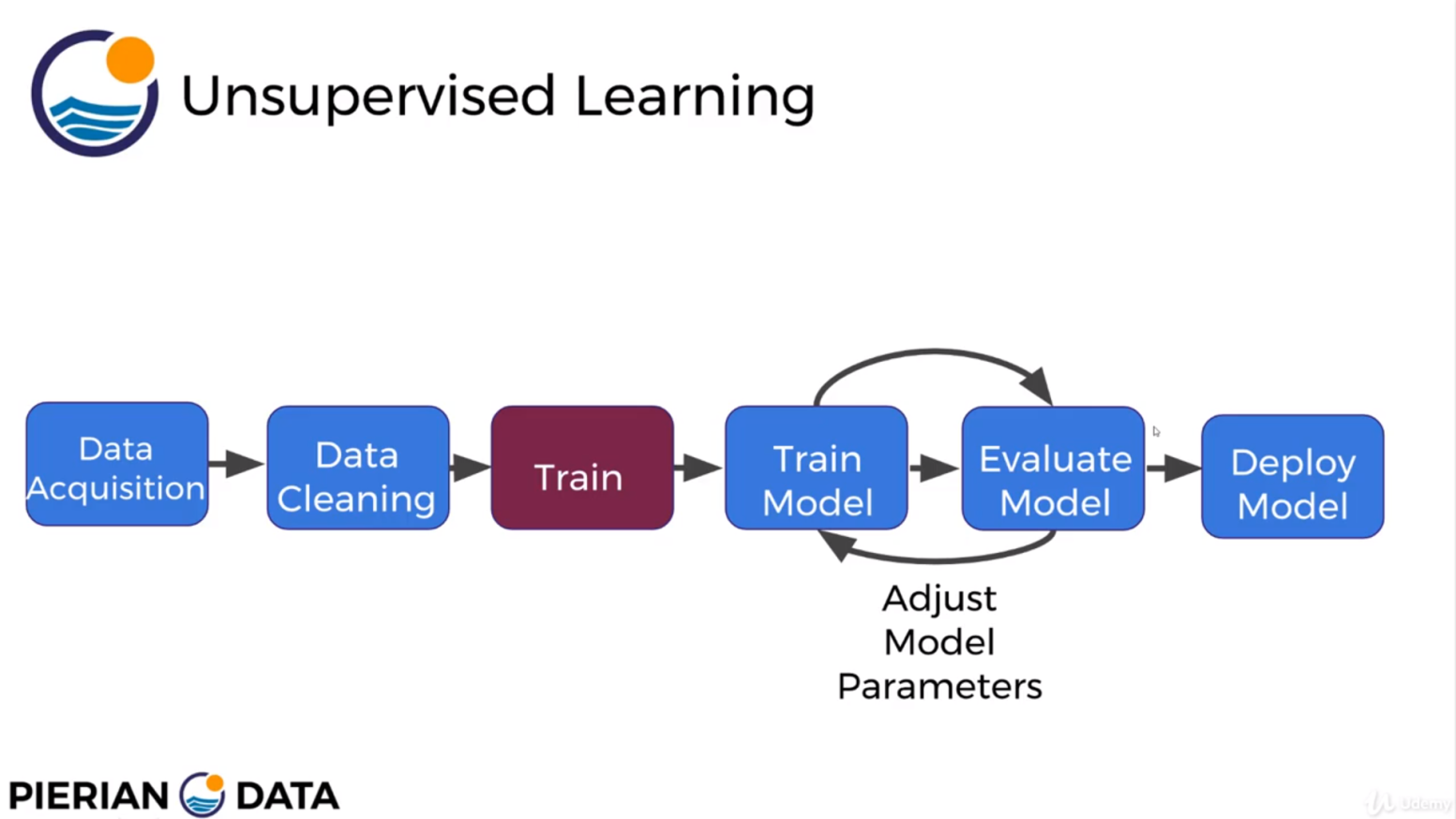
******

## User Experience Design

*The project will use a text interface with the options:*

1. *Select genre*
2. *Train model – using input from our database.*
3. *Compose piece*
4. *Exit*

## Data Model

**

#### 

##### Application

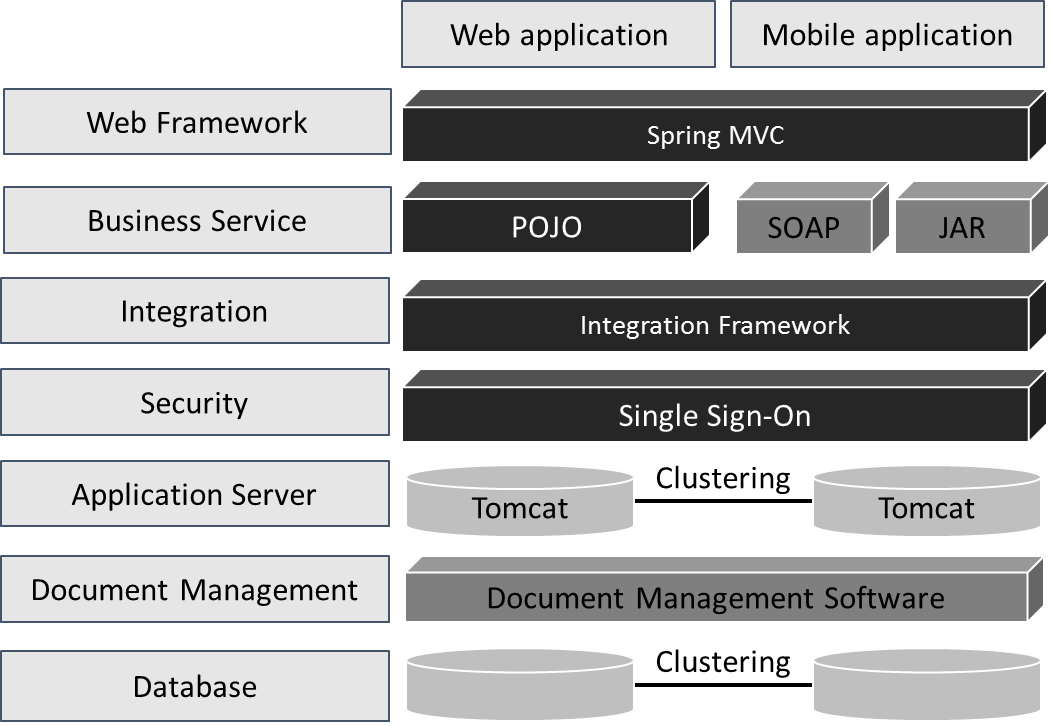
<Describe the architecture constraints, e.g. Platform only supports JDK v1.5>

<Describe the layout for application's high level technical solution. Include a visual diagram to illustrate the different layers in the application and their relationship if appropriate.> <Application Flowchart Diagram.> **Sample:**

|  |  |  |
| --- | --- | --- |
|  | ***Process Name*** |  |
| *System Flowchart* | *Start*  *Departmental*  *Portal*  *Library System*  *Document*  *Management*  *System*  *Mobile* | *End* |

<Divide the application into subsystems that implement the overall application functionality. Describe all the different layers in the application architecture, such as presentation layer, business layer and persistence layer.>

<Application Architecture Block Diagram.> **Sample:**

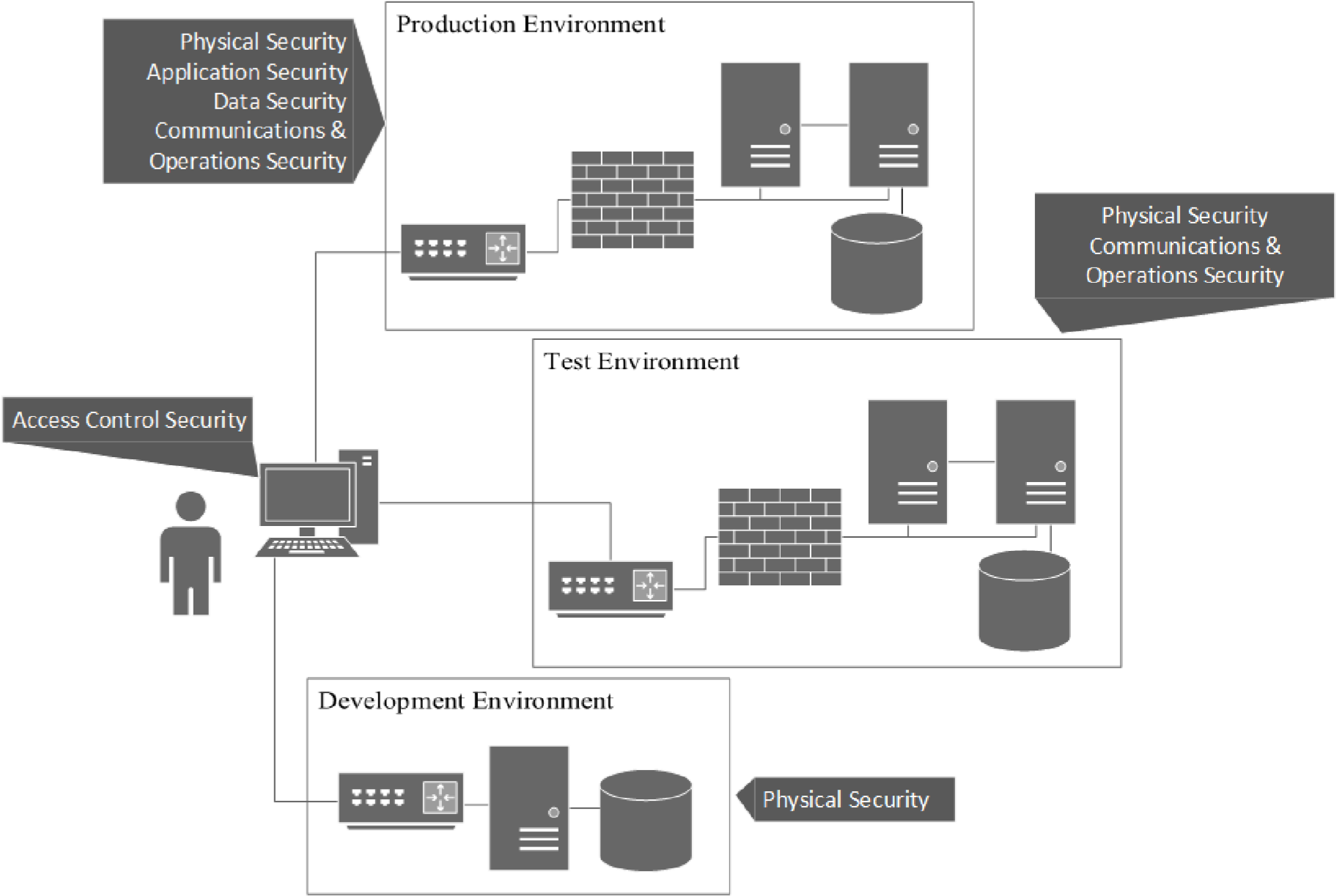


<Deployment Diagram.> **Sample:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Web application*  *Mobile application*   |  |  |  | | --- | --- | --- | | *Cluster 1* |  | *Cluster 2* | | |  |  |  | | --- | --- | --- | | *AppServer 1* |  | *AppServer 2* | |  | |  |  |  | | --- | --- | --- | | *AppServer 3* |  | *AppServer 4* | | | | |
|  | | |
|  | *DB*  *1*  *DB*  *2* |  |

<List the tools and software used for application development, security, integration, and other architecture components to develop the application.> Tools and software:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Tool / Software*** | ***Description*** | ***Finalised Tool / Software*** | ***Version*** |
| <Input the generic name or purpose of the tool or software> | <Input description> | <Input the specific or commercial name of the tool or software> | <Input version number of the tool or software> |
| **Database**  **Synchronisation** | **Oracle Data to manage all data information** | **Oracle / Active Data**  **Guard** | **11g** |



*Servers*

*Database*

*Firewall*

*Router*

*User*

*/*

*Terminal*

*Servers*

*Database*

*Firewall*

*Router*

*Database*

*Servers*

*Router*

##### Integration

<Describe the design approaches that will be taken for the various integration components of the application, such as web services, messaging, service orchestration, and batch.> <Application Communication Diagram.> **Sample:**

*Document*

*Management*

*System*

*Mobile library*

*application*

*Library System*

*Client interface*

*Accounting interface*

*Borrowing record interface*

*Client interface*

<Provide the necessary details about the deployment of the application.> <Integration Block diagram.> **Sample:**

*Mobile library*

*application*

*Document*

*management system*

*Library system*

*Message Bus*

*SOAP*

*JDBC*

*JSON*

***4.2.2 Data Architecture***

<Provide an overview of the content in Data section.>

##### Data Source

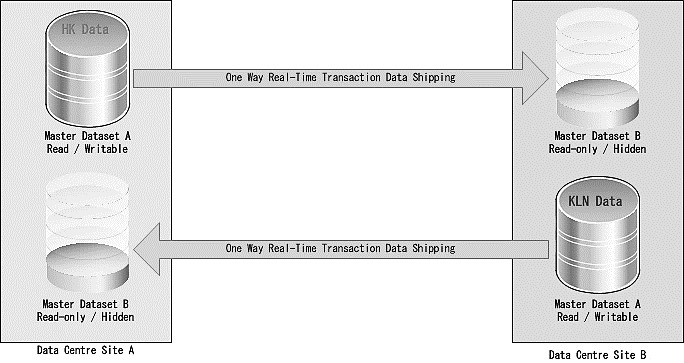
<Provide a list of data sources required for the data architecture design. Also describe the kind of data in these data sources and frequency of accessing this data.> Data sources:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Data Source*** | ***Description*** | ***Type of data*** | ***Frequency of access*** |
| **CRM System** | **Centralised CRM system**  **of all libraries in Hong Kong** | **Client Information** | **Once-off** |

##### Data Store Architecture

<Describe the core entities, data stores required by the system, the frequency of accessing the data and the flow of data across the different data stores.>

<Provide an overview of the core data entities and relationships which will be ingested and processed by the system.> <Data storage diagram.> **Sample:**



<Conceptual data model diagram.> **Sample:**

*Restriction*

*Client*

*Enquiry History*

*1*

*1*

*\**

*Address*

*1*

*\**

*\**

Core data entities:

|  |  |
| --- | --- |
| ***Entity*** | ***Entity Description*** |
| **Client** | **Personal information including DOB, HKID, gender** |
| **Address** | **Correspondence / Residential / Work address of Client** |
| **…** |  |

##### Data Retention and Archive

<Describe which and how the data are retained and archived. The elements that should be captured.>

Data retention and archive:

|  |  |  |
| --- | --- | --- |
| ***Data Element to be retained and archived*** | ***Archive Method and Frequency*** | ***Data Retention Policy*** |
| **Client record** | **Annually** | **Xx0000\_policy\_description** |

##### Data Conversion Architecture

<Identify sources of data to be extracted from one or multiple data stores and loaded into another, or multiple, target data stores and anticipated volume.> Data conversion:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Data*** | ***Source*** | ***Target*** | ***Anticipated Volume*** |
| <Data> | <Input source> | <Input target> | <Input anticipated volume of data> |
| **Client information** | **e.g. CRM system** | **Library System Database** | **10GB** |

<Define an approach for any data conversion required for a system implementation. A diagram may be created to illustrate the flow of data from the source to the target data stores.>

<Data migration diagram, if appropriate.> **Sample:**

*Customer*

*Relationship*

*Management*

*system*

*Library*

*System*

*Source*

*Staging*

*Transformation and*

*Data Quality*

*Target*

*Staging*

*Baseline*

*application*

*components*

*Data migration*

*technology*

*components*

*Target*

*application*

*components*

##### 4.3 SYSTEM DESIGN

<Provide an overview of the system design and use a block diagram to depict system diagram if required.>

###### 4.3.1 Application

<Provide an overview of the content in Application section.> <Class diagram(s).> **Sample:**

*Library System*

*Borrow Engine*

*Common*

*Client*

*Item*

*Document*

*Management*

*System*

**Borrow Engine is a component to manage borrowing book by Client.**

##### Design Application

###### 4.3.1.1.1 Describe Common Frameworks

<Class diagram(s).> **Sample:**

*Authentication*

*Spring Controller*

*Entity*

*Common*

*Controller*

*Exception*

*handling*

*Logging*

<Identify system components:

* Security - Assess sensitivities and define security control
  1. Review the functionality of each subsystem and the corresponding personally identifiable information (e.g., Hong Kong Identity card number) collected, used, stored and shared.

*○* For each subsystem, define security controls such as identity and access management, database security, network security and business continuity controls.

*○* Follow existing guidelines to help classify the subsystems and define security controls requirements.

*○* Develop and document a security controls implementation overview diagram which demonstrates where security controls will be implemented based on a user’s view. This will help the user and system owners understand where controls are placed and how data will be protected.

* Validation - determine how validations will be performed in the application
  1. It is recommended that any non-trivial business validations be located in the business services layer instead of the presentation layer. This makes the validation logic available for use by other parts of the application which are not invoked through the UI.
* Transaction - The changes on a single entity occurrence as triggered by an event is “effect”. The collection of all effects of an event is a transaction, which is either effective completely

(i.e., "committed”) or cancelled at all (i.e., "rollback”).

* Logging - define the approach to record and retrieve diagnostic information.
  1. Diagnostic information may include the amount of time needed to execute a critical method, the number of transactions committed per second, or the number of users currently with active sessions.
* Exception handling - define the approach to exception handling and how the exception will eventually be handled and logged and how it will be presented to the user.
  1. In general, the application should support unchecked exceptions. The guiding principle of exception handling in the application is that an exception should only be caught, and similarly a method should only declare a checked exception, if there is some valid response the invoker can make in response to that exception.
* Reference table - describe the approach to manage static look-up table or reference table information, such as country list, error messages.
* Internationalisation - describe the approach to handle internationalisation of the application. If internationalisation is not necessary, this should be stated. Considerations would include:
  1. Support for multi-lingual and other internationalised content.

*○* How database-backed internationalised content will be delivered. *○* Support for different character sets.>

**Sample:**

|  |  |
| --- | --- |
| ***Security*** | * **Support two-factor authentication** * **HTTPS encryption** * **Use parameterised SQL queries** |
| ***Validation*** | * **Input validation at the presentation layer using validation controls** * **Business rule validation logic in domain object** |
| ***Transaction*** | * **Use multiple active result sets to allow multiple queries to be executed using the same connection** * **Implement compensating methods to revert the data store to its previous state in case an operation within the transaction fails.** |
| ***Logging*** | * **Use Log4j for implementing logging** * **No sensitive information in logs** * **All logged message are time-stamped and tagged with the name of the generating controller** |
| ***Exception handling*** | * **Retry process for operations where data source errors or timeouts occur** * **Exceptions also posted in the Windows Event Logs** |
| ***Reference table*** | * **Reference table for list of countries that can be maintained** * **Use Address Data Infrastructure** |
| ***Internationalisation*** | ** **Support for Hong Kong Chinese Character Set** |
| ***Sample code artefact*** | **Nil** |

###### 4.3.1.1.2 Describe each subsystem into components

<Include a diagram to show the components of each subsystems.> <Class Diagram(s).> **Sample:**

**Library borrowing record subsystem**

*Library*

*-*

*String code*

*Member*

*-*

*Integer ID*

*Cart*

*Items*

*-*

*String Barcode*

*-*

*Borrow*

*()*

*Book*

*Video*

*Magazine*

*1*

*\**

*\**

*1*

*Borrow*

*Controller*

*Common*

*Controller*

*Borrow Record*

*\**

*1*

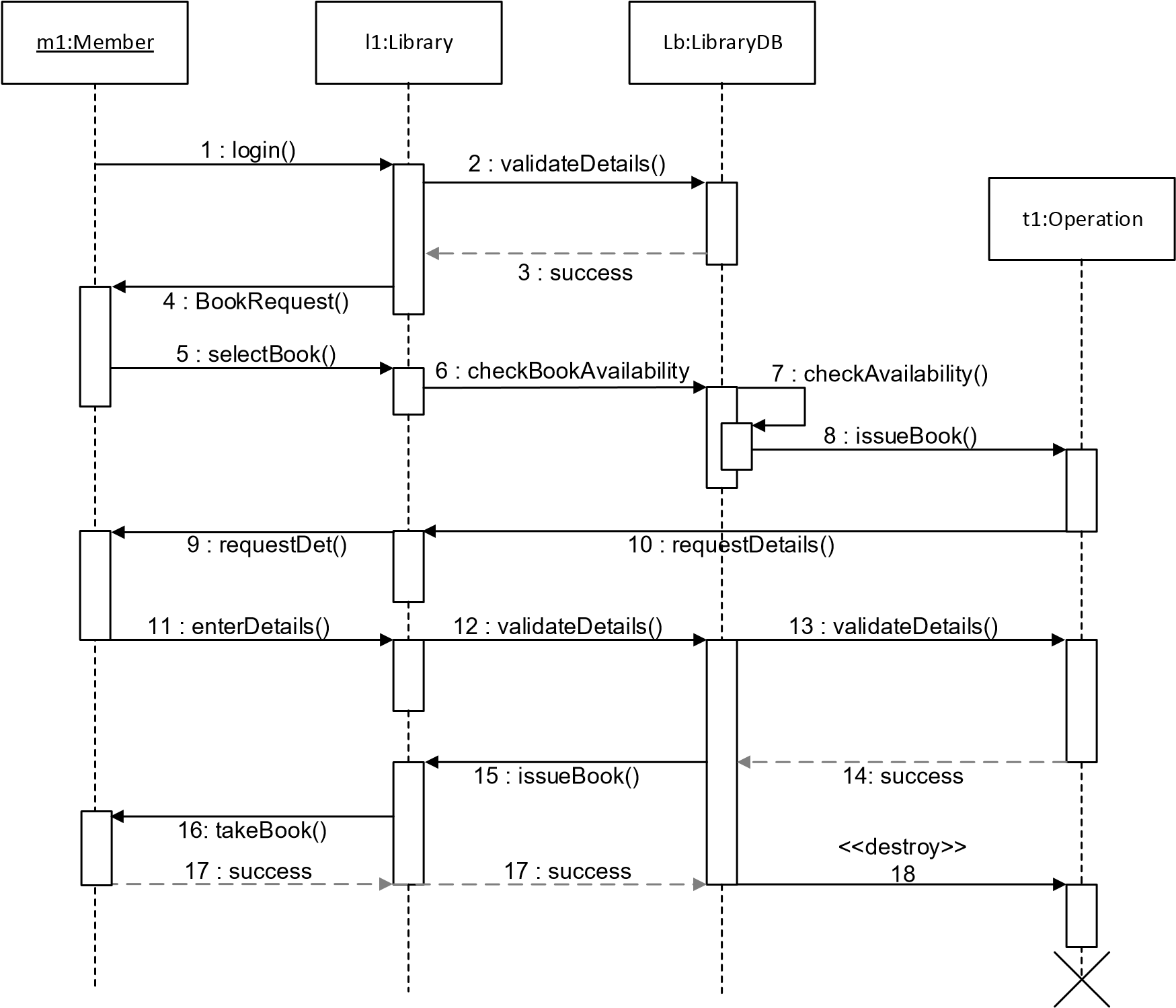
*Borrow Engine*

*Client*

*Item*

*Common*

<Include diagrams such as a sequence or activity diagram to identify the internal flow of control between components.> <Sequence Diagram(s).> **Sample:**



<Activity Diagram(s).> **Sample:**

*Check Availability of Book*

*Validate Member Code*

*Display*

*"*

*Book not found*

*"*

*Display*

*"*

*Not a valid member*

*"*

*Book available*

*else*

*else*

*Check no*

*.*

*of book issued to member*

*Display*

*"*

*No more books can be issued*

*"*

*Max quota exceeded*

*Create transaction*

*else*

*Update Book Status*

*Update member details*

*Enter Book Code*

*Enter Member Code*

*Add Transaction Details*

*Display*

*"*

*Book issued*

*"*

<Include diagrams such as state diagram to describe the behavior of a single object in response to a series of events in a system.> <State Diagram(s).> **Sample:**

*Membership Application*

*Submitted*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Do*

*/*

*Waitingl*

*Pending Pre*

*-*

*Approval*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Do*

*/*

*Waiting*

*Do*

*/*

*Process preapproval*

*Pending Approval*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Do*

*/*

*Get application report*

*Do*

*/*

*process approval*

*Application Rejected*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Do*

*/*

*output log*

*Application Closed*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Do*

*/*

*Data updating*

*[*

*Pre*

*-*

*approval*

*=*

*Yes*

*]*

*[*

*Pre*

*-*

*approval*

*=*

*No*

*]*

*[*

*Approval*

*=*

*No*

*]*

*[*

*Approval*

*=*

*Yes*

*]*

*Membership*

*record update*

*Application*

*Closure*

<Describe the design of implementation of business rules.> <List the business rules and capture their details.> Business rules description:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Rule#*** | ***Rule*** | ***Rule***  ***Attributes*** | ***Rule***  ***Conditions*** | ***Rule***  ***Actions*** | ***Rule***  ***Priority*** | ***Rule***  ***Validity*** | ***Dependency*** |
| <Input number> | <Input name defined> | <Document the attributes and operation performed > | <Describe the conditions that the rule checks  > | <Input actions> | <Input the priority of the rule> | <Input the constraints under which the rule is valid > | <Describe the dependency on another rule> |
| **1** | **Book\_issued** | **User.ID**  **No.Book.Issued** | **Book.Issued**  **=”Max quota exceeded”** | **Perform check whether No.Book.I ssued =< 3** | **1** | **Rule is applicable**  **only after**  **Nov 2014** | **N/A** |

##### Design Integration

###### 4.3.1.2.1 <Describe the integration design.>

<Describe the integration flows with each end system, and the type of message integration exposed by the interfaces with the end systems. Include diagrams to show the interfaces.>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Interface***  ***Name*** | ***Interface***  ***Frequency***  ***Type*** | ***Actors Involved*** | ***Context goal*** | ***Preconditions*** | ***Post conditions*** |
| <Input the interface  name> | <Input the frequency and schedule for the invocation of the interface > | <Document the end systems and any other actor using the interface> | <Document the functionality performed by the interface> | <Document any preconditions for the interface> | <Document any post conditions performed after the interface successfully completes the business functionality> |
| **Client interface** | **On demand** | **Document management system** | **The interface fetches all client information when refreshed** | **N/A** | **Request log is saved after successful execution** |

###### 4.3.1.2.2 <Include data mapping and transformation rules.>

Data control description:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **<Source System>**  ***Data Element*** | ***Required (Y/N)*** | **<Target System> *Data***  ***Element*** | ***Required (Y/N)*** | ***Mapping Logic*** |
| <Input field name> | <Input> | <Input field name> | <Input> | <Define any mapping rules between the two fields> |
| **Author.Name** | **Y** | **Author.LastName Author.FirstName** | **Y** | **Segregate**  **Author.Name into Author.LastName and**  **Author.FirstName** |

###### 4.3.1.2.3 <Describe design of integration sub-system.>

<Include diagrams to depict the logic implemented.>

<Provide details on sub process names, input and output fields, and data types. Define details on handling of exceptions, usage of global variables and implementation of logging and security frameworks. Document details on usage of any proprietary Application Programming Interface (API) names or design time libraries, field types, field names, and input/output details that will be used.>

##### Design Data Conversion

<Describe the source data entities, their business services and technical descriptions.> Source data entity description:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Source*** | ***Source Data Entity*** | ***Destination*** | ***Target Data Entity*** | ***Transformation/ Cleansing Rules*** | ***Notes*** |
| <Input source location> | <Input source data entity> | <Input target location> | <Input target data entity> | <Describe data transformation that is to occur> | <Describe any timing constraints or anything unique about the conversion> |
| **CRM** | **(Old) Client name** | **New Library System** | **(New) Client name** | **Cleanse preceding spaces** | **N/A** |

<Describe the expected results of the data conversion process.>

<Describe any business rules that need to be executed during the conversion process.> Data conversion results:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Source*** |  |  |  |  | ***Target*** |  |  |  |  |
| ***Data Entity*** | ***Description*** | ***Allowable Values*** | ***Field Type*** | ***Field Length*** | ***Cardinality*** | ***Mapping Rule*** | ***Data Entity*** | ***Field Type*** | ***Field Length*** |
| <Input> | <Input> | <Input> | <Input> | <Input> | <Input> | <Input> | <Input> | <Input> | <Input  > |
| **(Old) Client name** | **Name of**  **Client** | **Any alphabets** | **String** | **50** | **Mandatory** | **One-toone mapping** | **(New) Client name** | **String** | **50** |

<Document the data conversion tasks that must be carried out in advance or during the conversion runs, and create process flow and data flow diagrams to depict dependencies.> <Document error handling requirements and include error handling design in process and data flows.>

<Describe design of extraction programs, and conversion, cleansing, and loading programs.>

##### User Experience Design

<Include the UI information architecture such as a site map to provide a top-down view of how the users will interact with various pages, functions, and content and identify the screens or pages that would be developed for the application and their relations.>

<UI information architecture.> **Sample:**

*About*

*Home*

*Search*

*Using the Library*

*News*

*Events*

*Locations*

*Search by*

*Title*

*Advanced*

*Search*

*Borrow*

*,*

*renew*

*,*

*reserve*

*FAQ*

*Accessibility*

*Services*

<Describe any implementation guidelines of user interface components, and text copy of any content, error messages, labels, and other static content.>

<For each mock-up, include the picture of the mock-up, and describe the elements presented within the mock-ups, screen actions and messages if applicable.>

***4.3.2 Data Model***

<Provide an overview of the content in Data.>

##### Logical Data Model

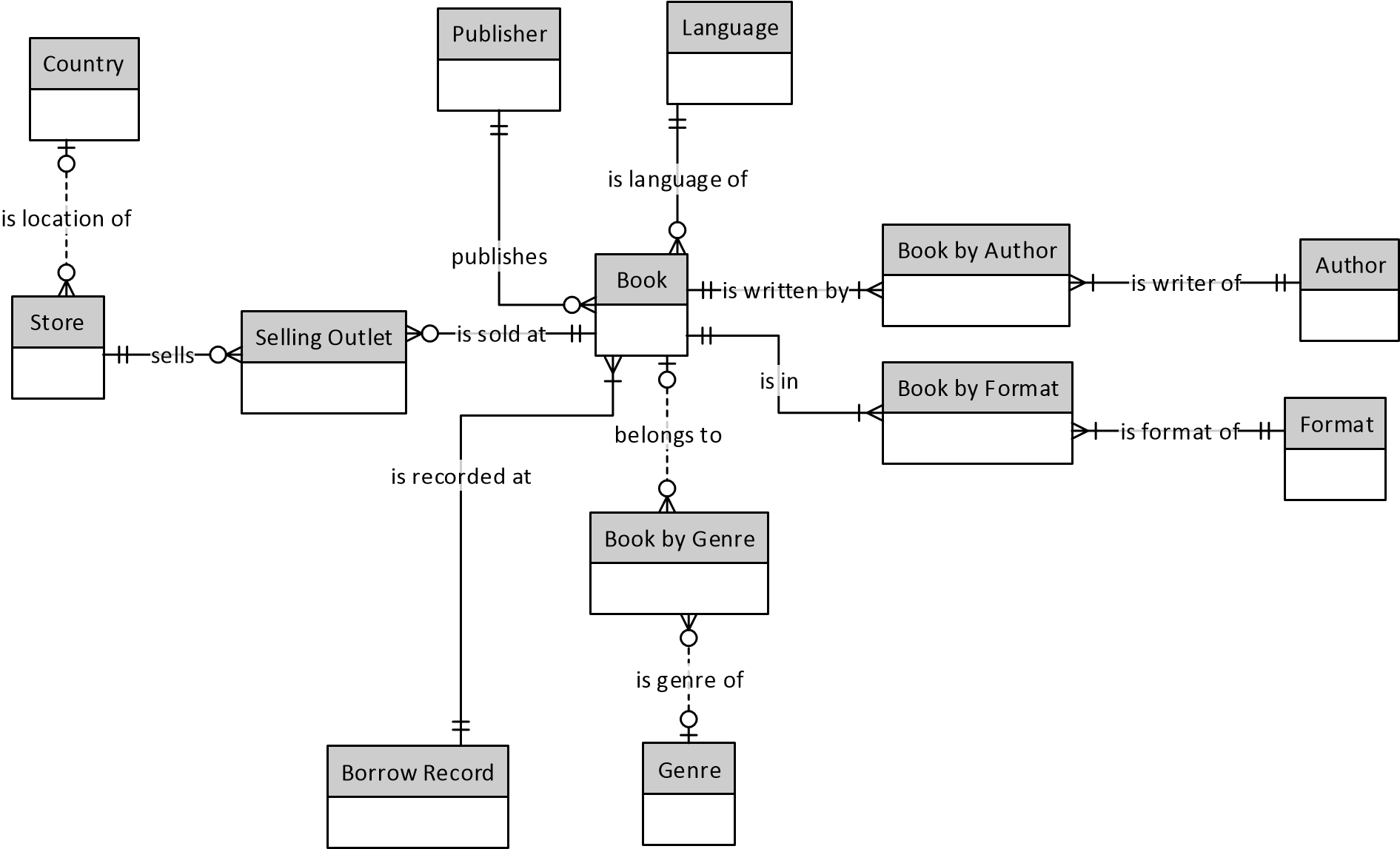
<Identify all entity types that will have one or more data attributes, such as First Name, Last Name, Middle Initial, and so on.> Logical data entity description:

|  |  |
| --- | --- |
| ***Logical Data Entity*** | ***Logical Data Entity Description*** |
| **BORROW RECORD** | **Includes a history record of borrowed book materials, such as written, printed, illustrated, or blank sheets, made of ink, paper, parchment, or other materials, registered to one User during a specific period.**    **Book title, ISBN and date in/out will be unique of each record** |

<Include a Logical Data Model diagram that depicts the following:

* Logical Data Entity Groupings – a logical data entity grouping is a collection of logically grouped attributes that are related to one another based on characteristics of those attributes. This may include but is not limited to: people, places, things, and concepts of interest to the business.
* Attributes – an attribute is a representation of a single elementary unit of business information.
* Relationships – a relationship shows how the logical data entity groupings are related, including cardinality (one-to-one, one-to-many or many-to-many). In relational logical models, many-to-many relationships must be resolved.>

<Logical data model diagram.> **Sample:**



##### Physical Data Model

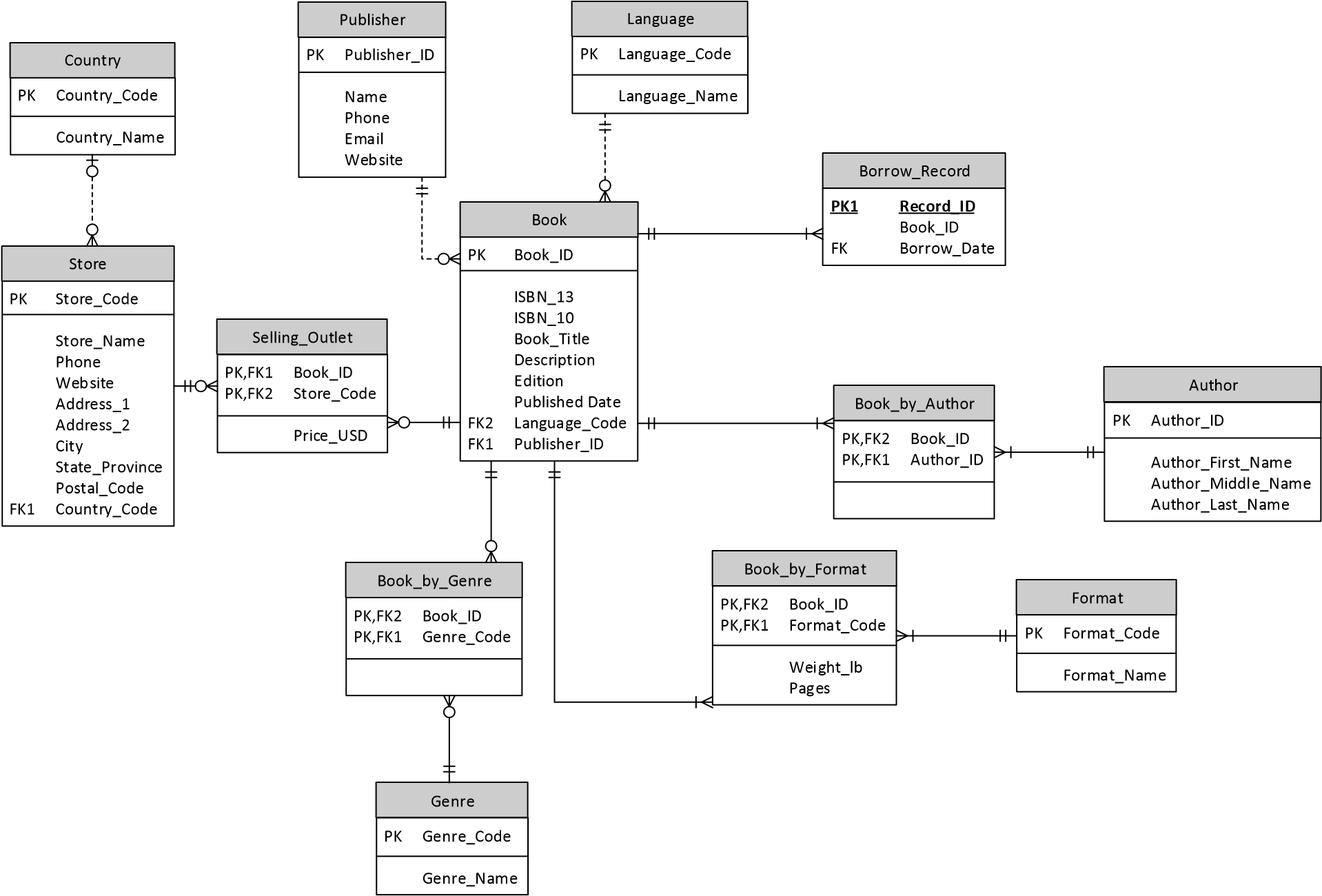
<Include the details of entities:

* Table Name: provide name of the table
* Field Name: provide name of the field
* Field Format: define the logical format of the field, e.g. Integer, Char
* Field Length: specify name of the field (e.g. Borrow Date)
* Description: provide a brief description of the field (e.g. Date book was issued to User)
* Mandatory: specify if the field is mandatory
* Primary Key: specify whether the field is used as primary key
* Foreign Key: specify the field name of another table for foreign key>

Physical data entity description:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Table Name*** | ***Field Name*** | ***Field Format*** | ***Field Length*** | ***Description*** | ***Mandatory*** | ***Primary Key*** | ***Foreign Key*** |
| <Input> | <Input> | <Input> | <Input> | <Input> | <Input Y/N> | <Input  Y/N> | <Input> |
| **BORROW RECORD** | **RECORD\_ID** | **VARCHAR** | **1000** | **Borrow**  **Record ID** | **Y** | **N** | **N/A** |

<Include a physical data model diagram.> <Physical data model diagram.> **Sample:**



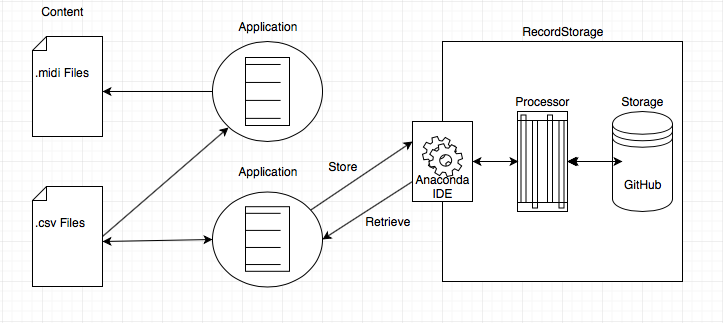
#### 5 5 TECHNICAL SYSTEM OPTION

##### 5.1 TECHNICAL SYSTEM ARCHITECTURE

###### 5.1.1 Network Architecture

Our application won’t require a network Architectture.

###### 5.1.2 Storage Architecture



The storage architecture consists of a database filled with the paths of the csv files which the program needs to analyse and preprosess to produce, the music/melody required. The application then uses the preprossed files to produce an output csv file which will then connect to another program which will convert it into a .medi file.

###### 5.1.3 Platform Architecture

GitHub] is a web-based hosting service for version control using git. GitHub is designed for collaborating on coding projects. Nonetheless, it is also a potentially great resource for researchers to make their data publicly available. Specifically you can use it to:

* store data in the cloud for future use (for free),
* track changes,
* make data publicly available for replication,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Environment*** | ***Machine*** | ***Hardware*** | ***Description*** | ***Software*** |
| Music  Generation | Personal Computer | Minimum 3 GB disk space to download and install Anaconda. | Application | * 32- or 64-bit computer. * Windows, macOS or Linux. * Python 2.7, 3.4, 3.5 or 3.6. * pycosat. * PyYaml. * Requests. * Anaconda IDE * Tensorflow |

The neural network will be trained to recognize/analyze melodies in sheet music -. midi files and learn from them. We will be using Tensorflow on the Anaconda IDE to develop our system. After we have trained the neural net to create its own melodies we can then add other musical components such as a drum beat and a bass line. We will use a Long Short-Term Memory network. They are a type of Recurrent Neural Network that can efficiently learn via gradient descent. Using a gating mechanism, LSTMs are able to recognise and encode long-term patterns. LSTMs are extremely useful to solve problems where the network has to remember information for a long period of time as is the case in music generation.

MelodyMaker’s boundaries will be the input data received from users that used the method previously. The melodies from data that has already been stored will form the interface that will enable the program to create new songs and melodies.

The program will look at aspects such as repetition and structure and use this to refine the new melodies that are to be written. Using these repetition and structure repetitions the program will be able to ensure that the melodies ”sound” harmonious and that it is audible to the listener.

##### 5.2 SIZING MODEL

* **Data Storage**

MIDI files are structured into chunks, a single header chunk followed by one or more track chunks. Each chunk consists of:

* A 4-byte chunk type (ascii)
* A 4-byte length (32 bits, msb first)
* length bytes of data

There are two types of chunks:

* Header Chunks, which have a chunk type of "MThd"

The data part of a header chunk contains three 16-bit fields. These fields specify the format, number of tracks, and timing for the MIDI file.

The length of the header chunk is 6-bytes. However software, which reads MIDI files, is required to honor the length field, even if it is greater than expected. Any unexpected data must be ignored.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Header Chunk*** | | | | |
| ***Chunk Type*** | ***Length*** | ***Data*** | | |
| 4 bytes  (ascii) | 4 bytes  (32-bit binary) | Length*(= 6 bytes)* | | |
| *16-bit* | *16-bit* | *16-bit* |
| *MThd* | Length | Format | Tracks | Division |

* Track Chunks, which have a chunk type of "MTrk"

The data part of a track chunk contains one or more delta time pairs. The delta time is not optional, but zero is a valid delta-time.

|  |  |  |
| --- | --- | --- |
| ***Track Chunk*** | | |
| ***Type*** | ***Length*** | ***Data*** |
| 4 bytes  (ascii) | 4 bytes  (32-bit binary) | Length bytes   (binary data) |
| MTrk | Length | Delta time, Event |

All functions in the system had been considered in the sizing analysis.

• The sizing model will cater for the projected 2-year growth of the system.

• 1 day = 8 working hours; 1 month = 30 working days.

• Transaction volume for Year 0 is determined based on that of existing system.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Entity***  ***Name*** | ***Annual Growth Rate (%)*** | ***Record Length (byte)*** | ***No. of Records*** | ***Record Storage (MB)*** | | |
| *Yr 0* | *Yr 1* | *Yr 2* |
| Midi.files | 10.00% | 4 | 3000 | 0.12 | 0.13 | 0.15 |

* **Transaction Rate**
* **Data Access**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Function***  ***Name*** | ***Transaction***  ***Volume of***  ***Year 0*** | ***Annual***  ***Growth***  ***Rate***  ***(%)*** | ***Mode*** | ***Transaction Rate (Hourly Peak)*** | | |
| ***Yr 0*** | ***Yr 1*** | ***Yr 2*** |
| Search Music Genre | 4,620,691 | 5% | Online | 2,406.61 | 2,526.94 | 2,653.28 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Function***  ***Name*** | ***Entities*** | ***Avg. No. of***  ***Records***  ***Accessed*** | ***Yr 2***  ***Transaction***  ***Rate***  ***(Hourly***  ***Peak)*** | ***Retrieve*** |
| Access Application | USER | 300 | 4.8 | 1350 |

|  |
| --- |
| ***Benefits***   * No Benefits   ***IF we should sell***   * Increased Sale: R30 000 |
| ***Development Cost: R2400***   * Udemy Courses: R2400.00 |
| ***Operational Cost: R0.00***   * Hardware: R0.00 * Software: R0.00 * Operational: R0.0 |
| ***Total Cost: R2400.00*** |

5.3 COST / BENEFIT EVALUATION

##### 5.4 IMPACT ANALYSIS

* **Summary on system change/enhancement**

**Current State:** The neural network will be trained to recognize/analyse melodies in sheet music -. midi files and learn from them.

**Future State:** After we have trained the neural net to create its own melodies we can then add other musical components such as a drum beat and a bass line.

* **Effect on organisation levels**

The application’s quality will be maximised throughout the neural network with the implementation of components such as a drum beat and a bass line. Implementation of the components will result in improvement of the MelodyMaker which will increase the application’s quality.

* **Significant changes in user operating procedures**

UserWill be able togenerating high quality music for their personal use.

* **Implementation considerations**

In addition to the identified impacts that will result from the deployment of the new systems, the transitional implementation period will also bring about Project changes. A amount effort and time will be required to facilitate the implementation of the new systems.

* **Change management**

Prior to system deployment, User acceptance test is required to assess if the functions developed within the application fulfil and work as specified within the functional specification document. Team members are required to participate in the UAT and test the functions to make sure the functions work as specified.

**Resulting Impact**: Due to the additional effort, the project team will have less time in performing their normal daily operational tasks.

**Recommended Solution:** Extending the daily time spent on developing the project to maximize our application’s efficiency.

* **Data Migration**

To ensure we reach our quality goal, an extraordinary amount of midi files will need be required. This will increase the time of the process, transferring data between file formats.

**Resulting Impact**: Due to the additional effort, the project team will have less time in performing their normal daily operational tasks.

**Recommended Solution:** Extending the daily time spent on developing the project to maximize the time spent on transferring data between file formats.

* **Risk Analysis**

With extensive changes required in implementing the music generation application, there are certain risks involved in completing the project. To ensure success in the system implementation, potential risks and the associated mitigation solutions are identified.

**Potential Risk**: Blues music incorporates more complex rhythmic and harmonic structure, adding additional components will add complexity, making learning inherently more challenging.

**Recommended Solution:** Extend the training and fine-tuning of the network , to ensure our model is able to learn to compose music that are both syntactically correct and also able to fool humans.

##### 5.5 IMPLEMENTATION PLAN

** Implementation Strategy

1. **Prepare the infrastructure.** This strategy includes a review of hardware, software, communications, etc. When you are ready for implementation, the production infrastructure needs to be in place.
2. **Coordinate with the organizations involved in implementation.** Part of the implementation work is to coordinate the with team members that have a role to play.
3. **Install the production solution.** Here our solution will be moved from development to test. If there are major changes to a current solution, we may have a lot less flexibility in terms of when the new solution moves to production, since the solution might need to be brought down for a period of time. We have to make sure all of your production components are implemented successfully, including new hardware, databases, and program code.
4. **Convert the data.** Changing our data from one format to another, this needs to take place once the infrastructure and the solution are implemented.
5. **Perform final verification in production.** Implementation of our testing method will take place to ensure everything is working as we expect. This may involve a combination of development and client personnel. The first check is just to make sure everything is up and appears okay. The second check is to actually push data around in the solution, to make sure that the solution is operating, as it should.
6. **Implement new processes and procedures.** Implementation of the new solution that requires success of our application will take place here. These changes will be implemented at the same time that the actual solution is deployed.
7. **Monitor the solution.** The project team will spend some period of time monitoring the implemented solution. If there are problems that come up immediately after implementation, the project team will address and fix them.

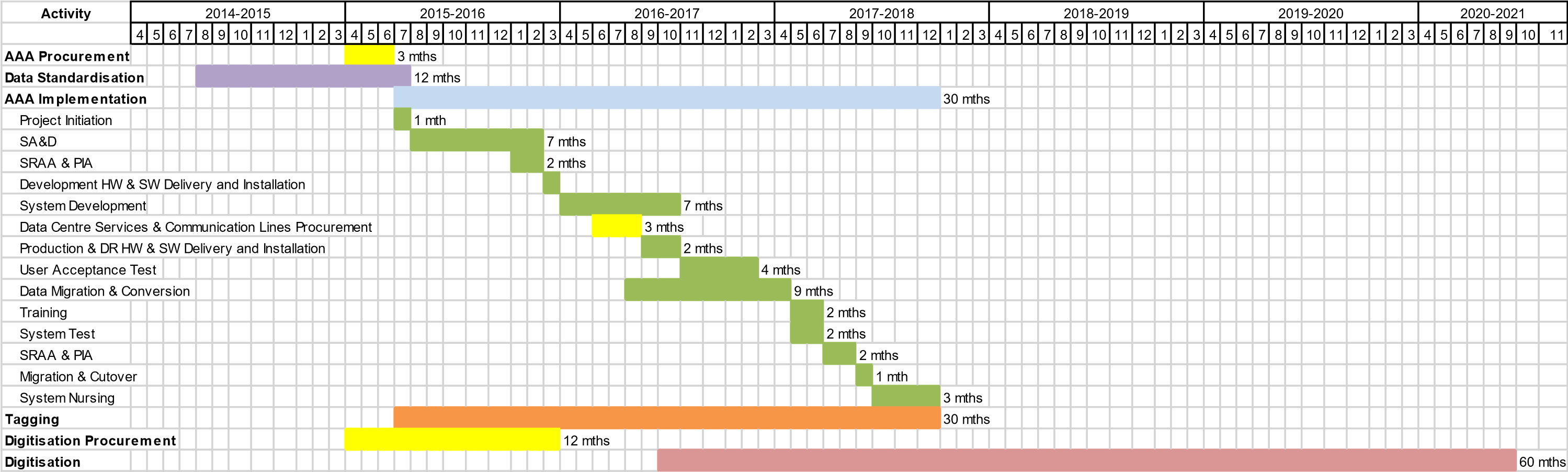
** Implementation Schedule

|  |  |  |
| --- | --- | --- |
| **Deliverable** | **Description** | **Planned Date** |
| Milestone 1  Final project proposal document submission | This includes the final proposal that will be assessed before commencement of the project. It is the first item of the development process that will be used to guide all other items on a basic level. | 2018-03-21 |
| Milestone 2  Planning document submission | Much like the proposal, the planning document will be used to guide the development of all other items, however, it will be used on a more in-depth scale. It provides guides and steps in a predefined manner that allows the configuration of all items to be executed as harmoniously as possible. | 2018-04-13 |
| Milestone 3  System Analysis and Design document submission | Here we identify, break down and assess all major and minor aspects of the development of the system, the environment, effects and the functionality of the system itself. | 2018-05-04 |
| Milestone 4  Technical 1: Database Design | This is technical aspect of the system. The database is one of the most important physical components of the system. Once this has been correctly developed, it can also be used for Class creation which is the next item. | 2018-05-25 |
| Milestone 5  Technical 2 : Class Design | This item can be based on the elements found in the database design. It is a technical item that defines the basis of code development of the system. | 2018-06-14 |
| Milestone 6  Technical 3 : Object Behaviour Model | This technical item works with thee classes and database as a baseline. However, it is also the baseline for coding the system. It will be assessed and defined before coding commences for the purpose of simplifying the intense complicated nature of coding. | 2018-07-04 |
| Milestone 7  Technical 4 : Coding | This is the item that configures all previous items. It is the physical compilation of all planning up until this point. The system is physically created here. | 2018-08-04 |
| Test plan document submission | Here we will identify our approach to testing the application. | 2018-09-14 |
| Milestone 8  Testing | This is where we will implement our testing strategy that we have identified in the test plan document for our application. | 2018-10-22 |
| Milestone 9  Implementation | This is where the implementation process of our application will be executed. | 2018-10-29 |
| Project Submission | The project is in final form and must be ready to be summited on the given date. | 2018-11-01 |

###### **With Sample Content**

***System Analysis & Design Report Technical System Option***

<Gantt Chart Diagram.> **Sample:**



**Key**

|  |  |
| --- | --- |
| **AAA** | **System Name** |
| **SRAA** | **Security Risk Assessment and Audit** |
| **PIA** | **Privacy Impact Assessment** |
| **HW** | **Hardware** |
| **SW** | **Software** |
| **DR** | **Disaster Recovery** |

*53*