**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

**PROJECT NAME**

# OF

## DDD DEPARTMENT

### *yyyy mm dd*

### Table of Contents

#### 5 TECHNICAL SYSTEM OPTION .............................................................. 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

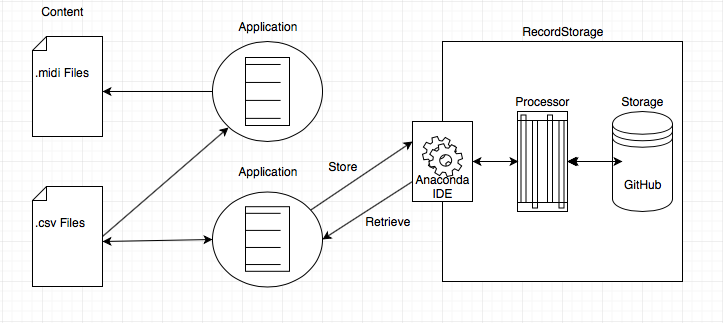
#### 5 TECHNICAL SYSTEM OPTION

##### 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 |

##### 5.3 COST / BENEFIT EVALUATION

|  |
| --- |
| **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** |

##### 5.4 IMPACT ANALYSIS

* **Summary on system change/enhancement**

**Curent State:** The neural network will be trained to recognize/analyze 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 spended on developing the project to maiximize our application’s efficency.

* **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 spended on developing the project to maiximize the time spended on transferring data between file formats.

* **Risk Analysis**

With extensive changes required in implementing the music geneartion apllication, 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 an 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 appproach to testing the application. | 2018-09-14 |
| Milestone 8  Testing | This is were 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 were the implementation process of our application will be excuted. | 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 |