**Software Design Specifications**

**Urdu Poetry Generator with Artistic Style**

**Version: 1.0**

|  |  |
| --- | --- |
| **Project Code** | F22-4302 |
| **Supervisor** | Ms. Sobia Iftikhar |
| **Co Supervisor** | Sir Sayed Yousuf |
|  |  |
| **Project Team** | Muhammad Anas (19k-1534)  Husnain Abbas (19k-1059)  Abdul Rehman (19k-1084) |
| **Submission Date** | 24-Nov-2022 |

**Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Name of Person** | **Date** | **Description of change** |
| 0.1 | Whole team | 30-Oct | Document Created |
| 0.2 | Husnain | 5-Nov | Added Introduction |
| 0.3 | Anas | 5-Nov | Added Overall System Description |
| 0.4 | Husnain | 12-Nov | Added External Interface Requirements |
| 0.5 | Abdul Rehman | 19-Nov | Added Functional Requirements |
| 0.6 | Anas | 19-Nov | Added Non-Functional Requirements |
| 1.0 | Whole team | 21-Nov | Finalized |

**Distribution List**

|  |  |
| --- | --- |
| **Name** | **Role** |
| Ms. Sobia Iftikhar | Supervisor |
| Sir Sayed Yousuf | Co-Supervisor |
|  | Jury |

**Document Sign-Off**

|  |  |  |
| --- | --- | --- |
| **Version** | **Sign-off Authority** | **Sign-off Date** |
| 1.0 | Supervisor | 24-Nov |
| 1.0 | Co-Supervisor | 24-Nov |
| 1.0 | Jury | TBD |

**Document Information**

|  |  |
| --- | --- |
| **Category** | **Information** |
| Customer | FAST-NU |
| Project | Urdu Poetry Generator with Artistic Style |
| Document | Software Design Specification |
| Document Version | 1.0 |
| Status | Draft |
| Author(s) | Husnain Abbas, Muhammad Anas, Abdul Rehman |
| Approver(s) | - |
| Issue Date | - |
| Document Location | - |
| Distribution | Advisor  Project Coordinator’s Office (through Advisor) |

**Definition of Terms, Acronyms and Abbreviations**

|  |  |
| --- | --- |
| **Term** | **Description** |
| **ASP** | Active Server Pages |
| **DD** | Design Specification |
| **SDS** | Software Design Specification |
| **RNN** | Recurrent Neural Network |
| **CNN** | Convolutional Neural Network |
| **GRU/Bi-GRU** | Gated Recurrent Unit/ Bi-directional Gated Recurrent Unit |
| **LSTM** | Long Short Term Memory |
| **BERT** | Bidirectional Encoder Representations from Transformers |
| **HAS2S** | Hierarchy Attention Sequence-to-Sequence model |
| **B/F-LM** | Backward/Forward-Language Model |

Table of Contents

[Introduction: 4](#_Toc120135328)

[1.1. Purpose of the Document 4](#_Toc120135329)

[1.2. Intended Audience 4](#_Toc120135330)

[1.3. Document Convention 4](#_Toc120135331)

[1.4. Project Overview 5](#_Toc120135332)

[1.5. Scope 5](#_Toc120135333)

[2. Design Considerations 6](#_Toc120135334)

[2.1. Assumptions and Dependencies 6](#_Toc120135335)

[2.2. Risks and Volatile Areas 6](#_Toc120135336)

[3. System Architecture 7](#_Toc120135337)

[3.1. System Level Architecture 7](#_Toc120135338)

[**High Level System Architecture** 7](#_Toc120135339)

[**System Component Diagram** 8](#_Toc120135340)

[3.2. Software Architecture 9](#_Toc120135341)

[**Layered Software Architecture** 9](#_Toc120135342)

[4. Design Strategy 10](#_Toc120135343)

[4.1. Future system extension or enhancement 10](#_Toc120135344)

[4.2. System reuse 10](#_Toc120135345)

[4.3. User Interface paradigm 10](#_Toc120135346)

[4.4. Data Management 10](#_Toc120135347)

[5. Detailed System Design 11](#_Toc120135348)

[5.1. Database Design 11](#_Toc120135349)

[5.2. Application Design 11](#_Toc120135350)

[5.2.1. Sequence Diagram 11](#_Toc120135351)

[5.2.2. State Diagram 12](#_Toc120135352)

[6. References 13](#_Toc120135353)

[7. Appendices 14](#_Toc120135354)

[List of TBD requirements: 14](#_Toc120135355)

[List of known issues: 14](#_Toc120135356)

[List of known design issues: 14](#_Toc120135357)

# Introduction:

Poetry has always been one of the important part of nation’s language and culture. Poetry has cultural as well as entertainment value and it is very popular among people especially adults and nowadays even young adults. Writing a poetry is not an easy task, in fact it’s an art to write poetry. Urdu is one of the most widely spoken languages in the world with more than 100 million speakers worldwide. In Pakistan, although pure poetry has been declining, however, poetry has always played an important role in education, mushairas as well as to pay tribute to martyrs and leaders/Sufis. Poetry is also a good way to communicate or express one’s feelings. Our goal is to create a platform where people can generate poetry based on their ideas as keywords which they provide as an input. The generated poetry will be based on “Ilm-e-Uroos” (art of creating poetry) and follow all Urdu poetry rules of rhythm, rhyme, meter and context. The purpose of this project is to help poets overcome writer’s block, help young or new poets understand rules of Urdu poetry since nowadays, not many poets follow the rules which results in poor quality of poetry as well as to provide entertainment and educational value.

## Purpose of the Document

The purpose of this document is to describe what our project will look like and how will it behaves, how will it be designed and using what techniques as well as explain the architecture of our system. This project provides overview on the working of the components within the system, system’s interactions with user or systems and overall design and architecture of the system. It also identifies the framework and a technology used for the development and tries to define the system architecture as well as design strategies use to build the web app for our system.

## Intended Audience

This document is intended to be reviewed by project supervisors as well as team lead and other team members if the need arises. It is also to be presented to project Juries as well as Externals who are evaluating the project.

## Document Convention

This document has been created on IEEE Standard System Requirements Specification Document. The font used in this document is Times New Roman. The font size is 12 pt. while the headings and sub-headings are according to MS Word’s pre-defined styles

## Project Overview

An online poetry generation website where user can generate Urdu poetry based on prominent Urdu poet’s styles and providing their ideas as keywords and the system will generate poetry in that context following all rules of Urdu poetry like rhythm, meter etc. The platform serves many purposes like overcoming writer’s block for anyone writing a poetry as well as students preparing for exams or competitions.

The web app is developed using HTML, CSS and JavaScript and its libraries like React JS and FAST API for backend. The model to generate poetry is developed using Python and libraries like Keras and Tensorflow. For designing the app, we first think about what problem we want to solve then we organize what information we have and make assumptions. We then create a prototype based on what knowledge we have and then test it.

## Scope

This software will be a tool for the authors to helps them writing a poetry. This system will be designed to maximize the author’s productivity by providing a platform to assist them while they are facing a writer’s block while writing poetry. This model can also help in preparing for poetry competitions or for Urdu poetry homework. It also provides entertainment value since poetry is a major source of entertainment. The model will generate highly accurate and precise poetry that could be used in poetic symposiums (mushairas). Poetry creation is viewed as a problem of sequence-to-sequence generation, and deep learning algorithms have emerged as a promising field. However, most research which has been published in this field is related to English, Chinese, Spanish, and Japanese. We want to put our part to empower Urdu literature by creating a model for Urdu poetry generation and make opportunities for further research.

# Design Considerations

## Assumptions and Dependencies

Following are the hardware assumptions for using the system

* **2 GHz** processor
* **4 GB** RAM
* **10 GB** HDD space
* Keyboard and Mouse
* Internet Browser (any)
* **Broadcom 802.11b** Network Adapter

Following are the design issues and how are we catering them in the system.

* **Security**: Any user sensitive information will be hashed as well as HTTP 1.1 which is more secure will be used. System will be thoroughly tested to check any security issue.
* **Quality of service:** The system will provide good quality of results and will be tested to ensure there are no quality issues.
* **Failure management:** Web app will be hosted on a web server provided by trusted vendor to ensure it is up all the time.

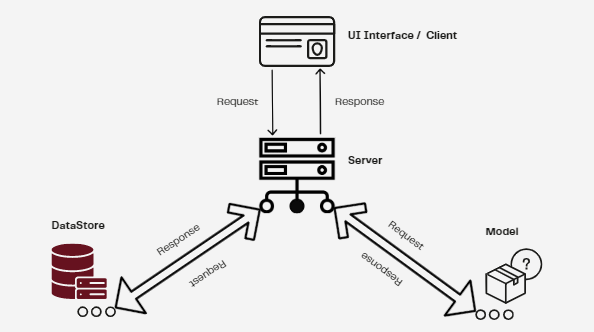
## Risks and Volatile Areas

The poetry generated by the system may vary depending on the given keywords and poet style but even if you provide same keywords, it will result in different generated poetry. The system could change if we decide to add different types of poetry in the future as well as if better performing model is released. The system is designed to easily adapt to any change like if a better performing model is released, we will thoroughly test it then simply integrate it with our system. Same will be done if new types of poetry or new poets styles are added.

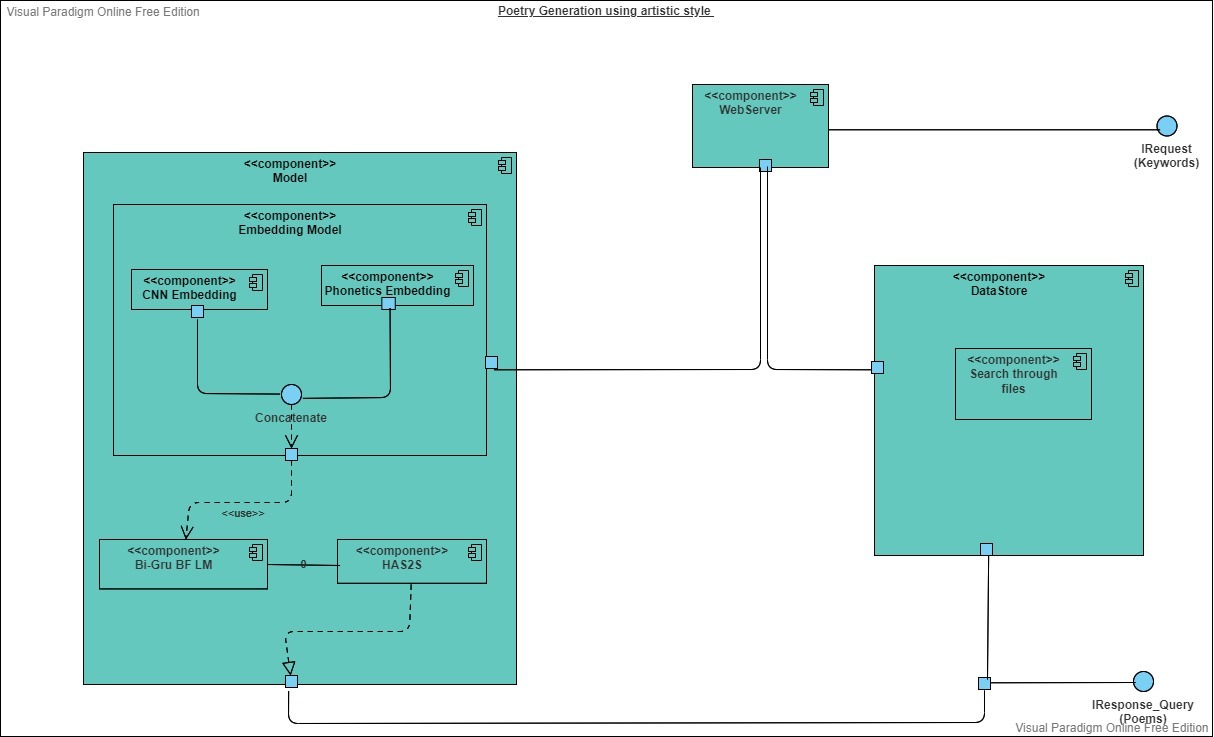
# System Architecture

Users can search or generate poems through our web application by entering keywords, and requests will be sent to the server. The server will decide that the request will either go to a data store or generative model. If the request is gone to the data store a list of poems will be rendered to the server. If the request is gone, the model will evaluate the keywords and generate the poem and render that poem to the server. The server will show the loading screen until the response is received. As the response is received the results are being displayed.

## System Level Architecture

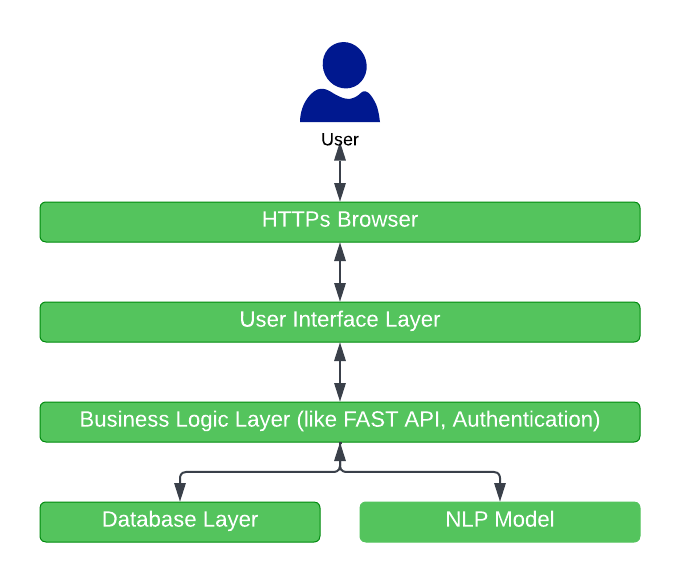
**

## **High Level System Architecture**



## **System Component Diagram**

## Software Architecture



## **Layered Software Architecture**

# Design Strategy

## Future system extension or enhancement

The system will be designed using latest technologies for both web app and model for generating poetry and in case if any new or better performing model is released, it will easily be integrated into the system if needed. For poetry generation, two models i.e. CNN sub word embedding using phonetics and GPT-2 will be used and the best performing model will be integrated in the system to enhance results. App may get future extensions like new poet’s styles or poetry type depending on time and resource constraints.

## System reuse

The system will be designed to maximize the reusability of its modules and components. Proper refactoring and testing will be performed to get rid of duplication, hard coding or unnecessary code. The app design will encourage the reuse as well as the model used for poetry generation will be built so as to be reused in different situations or scenarios.

## User Interface paradigm

The user interface is interactive and easy to navigate for even non-technical people. We will try to create as simple and interactive user interface as possible so that users can enjoy using our system and perform all tasks with an ease. The website will be fully responsive and fast to improve usability and user experience.

## Data Management

The data used for the training of the models which consists of Urdu poetry of different poets has been collected from online Urdu poetry platforms like Rekhta and Ranjish using data scraping. Data scraping has been performed using Python and its libraries like Response and Beautiful Soap. The data is collected in the form of csv file, there are multiple files for multiple poets in both Urdu and roman. After the collection of data, data formatting and cleaning was performed. It was then used to train our models and generate poetry.

The app expects only one data element from user and that is the keywords that represents user’s opinion or ideas they want to generate poetry about. The users will need to select poet’s style too like Mir Dard’s style is Sufism etc. Both of these things will be provided to the model and it will generate poetry according to poet’s style in the context of the provided keyword.

In the future, if any other poet style or poetry type is to be added, more data will be collected using the same techniques and then formatted and cleaned. Also, if current data failed to give adequate results, more data from different sources like videos, more websites, books etc. will be collected and used to train the models and get better results.

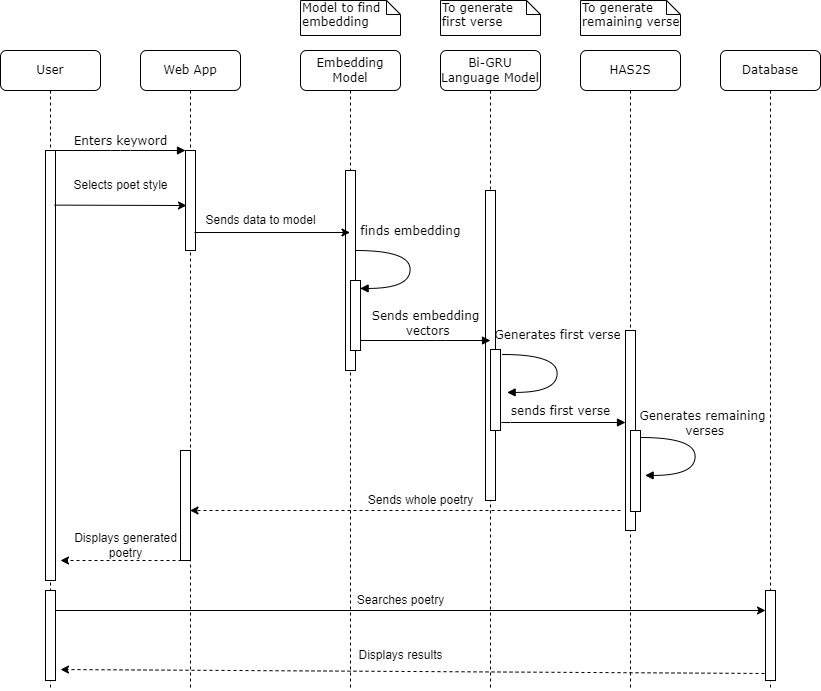
# Detailed System Design

## Database Design

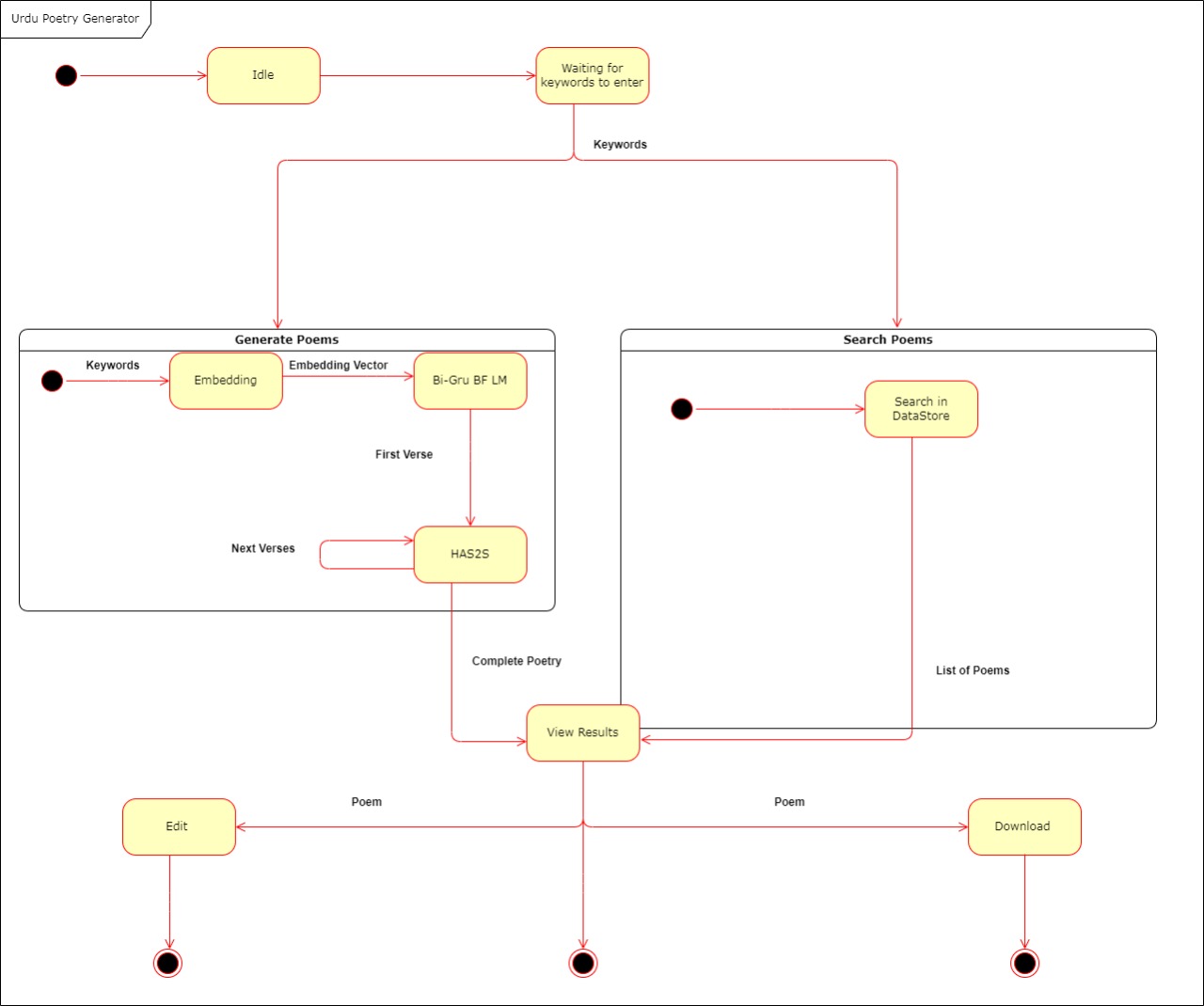
**Not applicable (Database is being sued to store poetry only there are no other relations)**

## Application Design

### Sequence Diagram



### State Diagram



# References

[1] <https://ashishjog2.wordpress.com/2014/09/23/article-what-is-beher/>

[2] <https://www.researchgate.net/publication/336860386_Arabic_Poem_Generation_Incorporating_Deep_Learning_and_Phonetic_CNNsubword_Embedding_Models>

[3] <https://www.cle.org.pk/Publication/papers/2016/Urdu%20Phonological%20Rules%20in%20Connected%20Speech%20.pdf>

[4] <https://arxiv.org/abs/1803.02994#:~:text=8%20Mar%202018%5D-,How%20Images%20Inspire%20Poems%3A%20Generating%20Classical%20Chinese,from%20Images%20with%20Memory%20Networks&text=With%20the%20recent%20advances%20of,its%20artistic%20and%20cultural%20value>.

[5] <https://aclanthology.org/P17-1016/>

# Appendices

### List of TBD requirements:

* User profile (Login/signup)
* Users can add poems to list of favorites
* More poet’s styles
* Different forms of poetry (poems, ghazals etc.)

### List of known issues:

* Providing unrelated input keywords may result in poor quality of generated poetry
* Results may take a little longer to display since the model generates poetry (it’s not same as simply retrieving from database)
* Providing poetry as an input keyword may result in poor quality of generated poetry

### List of known design issues:

* Since multiple models are being used and only the best performing model will be integrated within the final system, final design may somewhat differ from what described in this document.
* Role of database is to store poetry only, therefore, there may be no relations among entities (no ERD) and whatsoever.