**Light Statement of Work (SOW)**

**D**iabetic Retinopathy

| **Version** | **Date** | **Author** | **Description** | **Approver** | **SRS Reference** |
| --- | --- | --- | --- | --- | --- |
| **1.0** | **09/10/2024** | **Guy Yarhi** | **Initial draft** |  | **SRS\_v1.0** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Note:**

**Possible descriptions:**

**{Initial draft created, Revised system requirements, Added performance requirements, Major revision with new modules, Minor corrections and updates, Integration with other systems, Security improvements, User interface enhancements, Bug fixes and optimizations, Documentation updates}**

[Timeline & Milestones 3](#_Toc179381146)

[Week 1: Project Setup and Skeleton Definition 3](#_Toc179381147)

[Week 2: Skeleton System Architecture and Design 3](#_Toc179381148)

[Week 3: Data Collection and Preprocessing 4](#_Toc179381149)

[Week 4: Model Skeleton and Development 4](#_Toc179381150)

[Week 5: Feature Engineering Skeleton 5](#_Toc179381151)

[Week 6: Model Training and Evaluation (with Skeleton) 5](#_Toc179381152)

[Week 7: API Skeleton Integration 6](#_Toc179381153)

[Week 8: Testing and Documentation of Skeleton System 6](#_Toc179381154)

[Week 9: Final Review, Testing, and Deployment 7](#_Toc179381155)

[Key Dependencies 7](#_Toc179381156)

[Hardware Consideration for Speedup 8](#_Toc179381157)

[Version Control Strategy 8](#_Toc179381158)

[Resources 9](#_Toc179381159)

[Timeline Overview 9](#_Toc179381160)

**Project Overview**

This document outlines the high-level goals, tasks, and timelines for the development and testing of the system, including specific dependencies and resources. The project will be carried out in phases, starting with the setup and skeleton definition, leading to testing and final deployment.

# Timeline & Milestones

## Week 1: Project Setup and Skeleton Definition

**Task Overview:**

* Define project scope, architecture, and goals.
* Establish communication protocols and project management guidelines.
* Setup Python environment with Poetry.
* Create a skeleton for the system that includes the basic structure and dependencies.
* **Optional Consideration:** Given the computational requirements of the system, it is highly recommended to use **CUDA-enabled hardware** for faster training and execution of deep learning models. This will optimize the use of parallel computation capabilities of the GPU, thus speeding up processes significantly.

**Resources**:

* **DevPerson1** (Project Setup)
* **DevPerson2** (System Architecture)

**Checkpoint:**

* Project Kickoff and Skeleton Setup.

## Week 2: Skeleton System Architecture and Design

**Task Overview:**

* Design the overall system skeleton, ensuring modularity and separation of concerns.
* Design the basic structure of the system and database schemas.
* Define placeholders for model implementation.

**Resources**:

* **DevPerson1**
* **DevPerson3**

**Checkpoint:**

* Skeleton Architecture Demo.

## Week 3: Data Collection and Preprocessing

**Task Overview:**

* Organize dataset (2015 Diabetic Retinopathy Detection and 2019 Blindness Detection datasets).
* Preprocess images (resizing, cropping) and label preparation.
* Implement a skeleton for the data preprocessing pipeline.

**Dependencies:**

* Must wait for Skeleton Setup to be completed.

**Resources:**

* **DevPerson2**

**Checkpoint:**

* Data Preprocessing Skeleton Demo.

## Week 4: Model Skeleton and Development

**Task Overview:**

* Select the appropriate model (ResNet-18 or Self Neural Network).
* Build a skeleton of the model architecture using preprocessed data.

**Dependencies:**

* Wait for data preprocessing completion.

**Resources:**

* **DevPerson3**
* **DevPerson2**

**Checkpoint:**

* Model Skeleton Demo.

## Week 5: Feature Engineering Skeleton

**Task Overview:**

* Develop feature extraction methods to improve model performance.
* Integrate feature engineering with the model skeleton.

**Dependencies:**

* Wait for model skeleton to ensure feature alignment with the architecture.

**Resources:**

* **DevPerson4**
* **DevPerson2**

**Checkpoint:**

* Feature Engineering Skeleton Demo.

## Week 6: Model Training and Evaluation (with Skeleton)

**Task Overview:**

* Train the skeleton model using various hyperparameters.
* Evaluate model performance using metrics such as accuracy, precision, recall.

**Dependencies:**

* Wait for completion of the feature engineering skeleton.

**Resources:**

* **DevPerson2**
* **DevPerson3**

**Checkpoint:**

* Model Evaluation Demo.

## Week 7: API Skeleton Integration

**Task Overview:**

* Develop an API for sending images to the model.
* Build a skeleton of the API, focusing on endpoint creation and response formats.

**Dependencies:**

* Must wait for the model skeleton to be ready for integration.

**Resources:**

* **DevPerson3**

**Checkpoint:**

* API Skeleton Demo.

## Week 8: Testing and Documentation of Skeleton System

**Task Overview:**

* Write test cases to evaluate the functionality of the skeleton components (data preprocessing, model, API).
* Test each module independently for basic functionality.

**Dependencies:**

* Must wait for the API skeleton and model skeleton to be integrated.

**Resources:**

* **TestPerson1**

**Checkpoint:**

* Testing Skeleton Demo.

## Week 9: Final Review, Testing, and Deployment

**Task Overview:**

* Final code review and refactor if necessary.
* Conduct end-to-end testing of all components.
* Deploy the system and integrate it with HuggingFace.

**Dependencies:**

* Must wait for skeleton testing and API demo to be completed.

**Resources:**

* **DevPerson1**
* **DevPerson3**
* **TestPerson2**

**Checkpoint:**

* Final Demo and Deployment.

### Key Dependencies

* **Data Collection and Preprocessing →** Must be completed before Model Skeleton and Development.
* **Feature Engineering Skeleton →** Must wait until the Model Skeleton is done to align features with the model.
* **Model Training and Evaluation (Skeleton) →** Requires the completion of feature engineering skeleton.
* **API Skeleton Integration →** Requires the model skeleton to be functional before developing the API.
* **Testing and Documentation of Skeleton System →** Must occur after all skeleton components are integrated and tested.
* **Final Review and Deployment →** Can only occur after successful testing of skeleton components and integration.

### Hardware Consideration for Speedup

* **Optional:** Given the computational requirements of the system, it is highly recommended to use **CUDA-enabled hardware** for faster training and execution of the deep learning models. This will optimize the use of parallel computation capabilities of the GPU, thus speeding up processes significantly.

### Version Control Strategy

To maintain a systematic and organized approach to version control in our project, we will adhere to the following guidelines for saving models, data, and code:

1. Model Versions: Each time the model is updated or retrained, a new version will be saved with a unique identifier. The naming convention will be model\_vX.Y, where X represents major updates and Y represents minor changes or bug fixes. All model versions will be documented in the project log with details on the changes made, date, author, and any relevant notes.
2. Data Versions: Any modifications to the dataset, such as new data additions, cleaning processes, or feature engineering, will be saved as a new version. The data version naming will follow a similar convention, data\_vX.Y, and will be tracked in the project documentation. This ensures that we can reproduce experiments and maintain data integrity.
3. Code Versions: The source code will be versioned using a version control system like Git. Each significant change will be committed with a descriptive message, and branches will be used for different features or experiments. The main branch will hold the stable version, while feature branches will be merged back after thorough testing.
4. Project Documentation: All versions of the model, data, and code will be declared and updated in the project task management system. This will include a version history log, describing changes, authors, dates, and approvals. This practice ensures transparency, reproducibility, and efficient collaboration among team members.

By following these guidelines, we will maintain a robust version control system that facilitates tracking, managing, and replicating our work effectively.

## Resources

* **DevPerson1** (Project Setup, Skeleton Definition)
* **DevPerson2** (Data Preprocessing, Model Development)
* **DevPerson3** (API Integration, Model Training)
* **DevPerson4** (Feature Engineering)
* **TestPerson1** (Testing Skeleton)
* **TestPerson2** (Final Testing, Deployment)

## Timeline Overview

| **Week** | **Milestone** | **Duration** |
| --- | --- | --- |
| **Week 1** | Project Setup and Skeleton Definition | October 9 - 12 |
| **Week 2** | Skeleton System Architecture and Design | October 13 - 16 |
| **Week 3** | Data Collection and Preprocessing | October 17 - 20 |
| **Week 4** | Model Skeleton and Development | October 21 - 26 |
| **Week 5** | Feature Engineering Skeleton | October 27 - November 3 |
| **Week 6** | Model Training and Evaluation (Skeleton) | November 4 - 10 |
| **Week 7** | API Skeleton Integration | November 11 - 14 |
| **Week 8** | Testing and Documentation of Skeleton System | November 15 - 21 |
| **Week 9** | Final Review, Testing, and Deployment | December 2 - 9 |

**Conclusion**

This Statement of Work (SOW) document defines the project scope, tasks, milestones, dependencies, and resources. The project is structured in weekly phases, with clear checkpoints and assigned responsibilities. The use of **CUDA-enabled hardware** for faster deep learning model execution is an optional consideration that could significantly improve the system’s performance.