Title: Depression Detection System using Python
Project Domain : Machine Learning & Artificial Intelligence
Project Guide: Prof Laxmi bhagwat
Group no: P7

PRN	NAME	ROLL NO
1032211329	**ARYAN BANSAL**	**PB-25**
1032211208	**AERTH SARAOGI**	**PB-**
1032211401	**AKSHIT SINGH**	**PB-30**
1032211476	**SAI VENKTESH DUBEY**	**PB-33**

1. LITERATURE REVIEW

Sr No.	Publication Title with Authors [mention whether Journal or Conference paper]	Publication Year	Introduction	Gaps in Publication work	Future Research	Conclusion
1.	Predicting Depression: a comparative study of machine learning approaches based on language usage- **Luciana Mariñelarena Dondena, Edgardo Ferretti, M. Maragoudakis, Maximiliano Sapino, M. Errecalde**	2017- Journal	The introduction talks about depression, a condition that affects more than 332 million people around the world. Cases have been growing since 2000. People with depression often lose interest in things they once enjoyed. Sometimes, they even have thoughts about hurting themselves. This really shows how important it is to spot these signs early & to find good treatments like CBT.	We really need to take extra steps beyond just using classification techniques. Class imbalance in training data? It's still a tough problem.	Explore some new ways to manage class imbalance. There are different text mining techniques out there for checking psychological health.	Deep learning has really taken the lead when it comes to spotting depression. In fact, when you mix SMOTE with deep learning, you can get over 94% accuracy.

Sr No.	Publication Title with Authors [mention whether Journal or Conference paper]	Publication Year	Introduction	Gaps in Publication work	Future Research	Conclusion
2.	Ensemble Hybrid Learning Methods for Automated Depression Detection- LunaAnsari,ShaoxiogJi,ChenQian,Erik Cambria	2023 - Journal	The introduction shines a light on how depression is becoming more common. It might even be one of the biggest health problems by 2030! The text also looks at using language patterns to help figure out if someone has depression automatically, even though there are issues like stigma & misdiagnosis.	To improve how we detect depression, future studies should look into fairness, bias, uncertainty, & how we explain the results.	Think about using POS tags & working on imbalanced datasets. You can also look into how user personalities connect to depression indicators.	Ensemble models do a better job than hybrid ones for figuring out depression. By using a bunch of different language features, we see an improvement in how effectively we can find signals of depression.
3.	Automatic Detection of Depression by Using a Neural Network.- **Mahsa Raeiati Banadkooki , Corinna Mielke, Klaus-Hendrik Wolf , Reinhold Haux , Michael Marschollek**	2018 - Journal	The introduction mentions that depression affects over 300 million people worldwide! There's a big need for better ways to detect it. The authors took the PHQ-9 questionnaire and made it simpler by using PHQ-5. They changed the rules a bit for spotting depression.	This study looks at recent surveys and finds a big gap in automatically spotting depression. The authors have changed the standard cut-off from PHQ-9 to PHQ-5. But they say we need more proof to back this up.	In the future, research should focus on better depression detection models. Adding more data, like physiological information, could really help with diagnosis.	Neural networks have shown great success too. The PHQ-5 questionnaire, especially with a cut-off score of 8, is quite good at helping identify depression.
4.	Implementation and Analysis of Depression Detection Model using Emotion Artificial Intelligence- **Unnati Chawda, Shanu K Rakesh**	2019 - Journal	The introduction addresses how concerning depression can be for people's physical health and everyday life, including school. Early detection is super important. It even suggests using Natural Language Processing to look at tweets on Twitter for helpful insights!	One key issue highlighted in the study is how we diagnose depression early by using real-time data from social media. There's a need for models that can check language trends on Twitter and other platforms with Natural Language Processing.	Also, future studies should make emotion detection algorithms even more accurate. It can include different social media types & check out how culture and language play a role in showing mental health issues.	For building a depression detection model, emotion AI & natural language processing are super useful. With some Python code.

Sr No.	Publication Title with Authors [mention whether Journal or Conference paper]	Publication Year	Introduction	Gaps in Publication work	Future Research	Conclusion
5.	Depression Detection on Social Media- **RafaelSalas-Zárate, Giner Alor-Hernández,María del Pilar Salas-Zárate,Mario Andrés Paredes-Valverde,Maritza Bustos-López,José Luis Sánchez-Cervantes4**	2022 - Journal	As things get worse with depression, social media might be a way to help find symptoms quicker. The idea is to mix past data with real-time info—this could make detecting depression much more accurate. They're thinking about using machine learning and topic modeling to do this!	Also, the research points out a weakness when it comes to using messy social media data for spotting depressive symptoms effectively. It calls for better frameworks that blend advanced machine learning with big data processing to make sense of things.	Let's aim to improve depression detection algorithms by using various data sources & advanced natural language processing.	Also, Twitter data can be helpful in spotting symptoms of depression. A chatbot could help users who are feeling down; that's a neat idea.
6.	Automated detection of clinical depression based on convolution neural network model- **Dan-Dan Yan, Lu-Lu Zhao,Xin-Wang Song,Xiao-Han Zang, Li-Cai Yang**	2022 - Journal	The intro talks about Major Depressive Disorder. The goal is to find a way to pick up brain signals in people with MDD. This might help them get the support they need.	The paper talks about importance—there's a big difference in how we detect. Many current methods don't really have the objectivity needed. It's asking for some fresh ideas using deep learning. This means using convolutional neural networks on data for better diagnosis which sounds cool!	Future work should focus on making CNN models like EEGNet even better—aiming for above 94.27% accuracy! Using larger, diverse datasets & looking into other deep learning methods might lead to improved ways to detect depression.	CNNs have been great for detecting depression using EEG data. The EEGNet is fast & accurate, which is what we love to see.
7.	Development of an SVM-based Depression Detection Model using MFCC Feature Extraction- **V Maheshwar,N Venu Gopal,V.Naveen Kumar,D Pranavi, Y Padma Sai**	2022 - Journal	In this introduction, we see how clinical is becoming more common.To make things better, it suggests using deep learning, like CNNs, to look at EEG data. This can help spot depression more automatically.	It mentions that we need models that work for everyone. Right now, most systems get about 75 to 80%. That's not very good. Different symptoms & privacy make it even harder to make progress.	Strengthening depression detection models is crucial, too! It would be great to use varied datasets that capture different speech patterns and symptoms.	There's also an SVM-based model that uses MFCC feature extraction which hits 89% accuracy.

Sr No.	Publication Title with Authors [mention whether Journal or Conference paper]	Publication Year	Introduction	Gaps in Publication work	Future Research	Conclusion
8.	Depression Detection on Twitter Social Media Using Decision Tree- **MarcelloRasel Hidayatullah,Warih Maharani**	2022 - Journal	The introduction makes a strong case for catching depression early. There's a plan for a machine learning system using the DAIC-WOZ dataset. This could really boost how accurately we diagnose depression.	The paper says ways to detect depression overlook key details. More accurate and timely methods are needed—maybe we could use real-time social media data to help us better understand what's going on.	The authors suggest studying larger datasets and experimenting with different parameters and feature extraction methods.	We found that the best model reached 81.25% accuracy with an 85.71% F1 score. Tweaking things like TF-IDF features & tree depth made a difference in how well the model performed.
9.	Detection of Depression Symptoms Using Chatbots Based on Machine Learning- **Vitor Bastos,André Felipe Monteiro**	2020 - conference paper	The intro highlights that more students are feeling depressed these days. It suggests creating machine learning chatbots in an app for smartphones. This could help with conversations and figure out if someone is at risk of depression.	There's not much research about how well chatbots work in real life situations. We really need professionals to guide us with both ethical and technical improvements.	Getting psychology & psychiatry professionals involved in more complex tests is also a smart idea! We need to enhance the chatbot's conversations to keep users engaged.	While our proposed model shows good accuracy in tests, it still needs more work. Input from professionals would really help refine it.
10.	Intelligent Depression Detection System Using Effective Hyper-Scanning Techniques- **Minakshee Patil, Vijay M. Wadhai, Dhanashri H. Gawali, Akshita S. Chanchlani**	2020 - Journal	The introduction dives into how depression hurts not just mental health but physical health, too! Daily life gets impacted as well. There's a clear need for better ways to diagnose it. One idea is to use hyper-scanning & machine learning to analyze speech closely for accurate detection.	Spontaneous speech is actually better than read speech when it comes to spotting depression. Sometimes, just a short clip of talking can tell us just as much as a longer speech sample.	Let's dive deeper into using acoustic features for spotting depression. It's important to check different machine learning algorithms for classification effectiveness as well!	Interestingly, spontaneous speech worked better than read speech in classifying depression. Acoustic features such as MFCC, pitch, jitter, shimmer, and log energy turned out to be effective tools

2. REQUIREMENTS GATHERING

1. Introduction

The objective of the Depression Detection System is to create an accessible tool for early diagnosis of depression using both a diagnostic quiz and real-time facial expression analysis. This system targets users who may be experiencing symptoms of depression by providing an analysis that identifies the most likely type of depression (e.g., anxiety, PTSD, or bipolar disorder) and suggests potential treatment options

2. Stakeholder Analysis

- Primary Stakeholders:

- End Users: Individuals experiencing potential symptoms of depression who will interact with the system to assess their mental health status.

- Healthcare Professionals: May refer patients to use the system or use the system's outputs to support diagnosis.

- Development Team: Responsible for creating, testing, and deploying the system.

- Project Mentors: Guide the development team through feedback and oversight.

3. Functional Requirements

- **Quiz Module**:

- The system must include a diagnostic quiz consisting of multiple-choice questions to assess the user's mental health state.

- The quiz should be processed using a Naive Bayes classifier to determine the type of depression (anxiety, PTSD, bipolar disorder).

- The system should recommend clinics and expert assistance based on quiz results.

- It should generate an interactive map for the user to locate nearby treatment facilities.

- **Facial Expression Analysis Module**:

  - Users must be able to record a one-minute video where they talk about themselves.

  - A Convolutional Neural Network (CNN) must analyze the user's facial expressions to evaluate emotional states.

  - The system should process the data in real-time and refine the diagnosis based on facial analysis.

- **Frontend-Backend Integration**:

  - The frontend, developed using HTML, CSS, and JavaScript, must interface seamlessly with the Python-based backend.

  - User data, including quiz results and facial expression analysis, must be stored in a MySQL database.

- **Error Handling**:

  - The system must handle potential errors in both quiz results and emotion recognition, ensuring reliable outputs.

4. **Non-Functional Requirements**

- **Usability**:

  - The system must have an intuitive user interface, making it accessible for users with varying levels of technical expertise.

- **Performance**:

  - Real-time facial expression analysis must be efficient, with results available within seconds after the user’s video is recorded.

- **Scalability**:

- The system should be scalable to accommodate a large number of users and provide quick responses even under load.

- **Security**:

  - User data, including quiz answers and facial videos, must be securely stored and processed, ensuring privacy and confidentiality.

5. **Hardware and Software Requirements**

Hardware Requirements:

- **High-Performance GPU**: Essential for training the CNN model efficiently, especially when processing large datasets. A GPU like the **NVIDIA RTX 3090** or equivalent is recommended.
- **Microphone and Camera**: High-quality microphone and camera for recording facial expressions in real-time and ensuring clear audio/video input for accurate detection.
- **High-Capacity Storage**: **SSD storage** for handling large datasets, model checkpoints, and logs. Ensures faster data retrieval and processing.
- **Multi-Core CPU and RAM**: A powerful CPU (at least 8 cores) and **32GB RAM** to support real-time data processing and model training without delays.

Software Requirements:

- **Backend Processing and Machine Learning**:
  - **Python** with **FastAPI** for handling backend tasks, including machine learning processes.
- **Web Framework**:
  - **FastAPI** and **Node.js** for managing API endpoints and web services.
- **Database Management**:
  - **MySQL** or **MongoDB** for storing user data, quiz results, and real-time facial expression analysis.
- **Frontend Development**:
  - **HTML**, **CSS**, **JavaScript**, and **ReactJS** for building an interactive and user-friendly interface.
- **Machine Learning Libraries/Tools**:
  - **OpenCV** for real-time facial expression recognition.
  - **scikit-learn** for the Naive Bayes classifier used in the quiz analysis.

    - **TensorFlow/Keras** for the CNN (Convolutional Neural Network) model to detect emotions from facial expressions.
- **Version Control**:
    - **Git** for version control, with **GitHub** or **GitLab** for remote project repository management and team collaboration.

6. System Design

- High-Level Architecture:

- The system is divided into two main components: the Diagnostic Quiz Module and the Facial Expression Analysis Module. Both are integrated into a central server that communicates with the MySQL database to store and retrieve user data.

- Use Case:

- Users will initiate the process by taking the diagnostic quiz. Based on the results, they will proceed to record a video, which will be analyzed to further refine the diagnosis.

7. Constraints

- Data Privacy: The system must adhere to data privacy standards, ensuring that sensitive user information, including quiz answers and facial videos, is securely processed and stored.

- Accuracy of Emotion Recognition: The system may encounter difficulties in correctly identifying emotions from facial expressions, particularly in diverse lighting conditions or camera quality.

8. Risk Management

- Model Accuracy: Both the Naive Bayes classifier and CNN are susceptible to errors in prediction, potentially leading to incorrect diagnoses. Regular model updates and user feedback mechanisms should be implemented.

- Real-Time Performance: Real-time processing of facial expressions may slow down if the system encounters too many simultaneous users. Load testing should be conducted to mitigate this risk.

9. Future Enhancements

- Integration with Wearable Devices: Future versions of the system could integrate data from wearable health devices, providing even more personalized health recommendations.

- Improvement in Diagnosis: The facial expression recognition model could be expanded to capture more nuanced emotional states, improving the accuracy of the system.

3. PROBLEM STATEMENT AND JUSTIFICATION

Several key gaps have been identified in the literature on depression detection:

1. **Lack of Real-Time Data**: Most systems rely on static datasets and lack real-time physiological data, such as facial expressions.

2. **Emotion Recognition Challenges**: Current models struggle with accurately identifying emotions in diverse conditions.

3. **Bias and Fairness**: Many models lack fairness and often exhibit bias, with limited ability to explain results.

4. **Limited Integration**: Few studies effectively combine psychological assessments with physiological indicators like facial expressions for a more comprehensive diagnosis.

This project aims to address these gaps by developing a Depression Detection System that integrates a diagnostic quiz with real-time facial expression analysis. The system will use a Naive Bayes classifier for quiz-based depression detection and a Convolutional Neural Network (CNN) for real-time emotion recognition. This approach will:

- Incorporate Real-Time Data through facial expression analysis.

- Enhance Emotion Detection Accuracy using CNN models.

- Ensure Fairness and Reliability by providing a more comprehensive diagnostic tool that combines psychological and physiological data.

4. Individual Contributions for the Depression Detection System (4 Members)

Member 1:

- **Data Collection Leadership**: Responsible for collecting and curating datasets for both the quiz and real-time facial expression analysis. Ensures diversity in the dataset to capture different types of depression and emotional expressions.
- **Naive Bayes Classifier Development**: Leads the development and training of the Naive Bayes model for diagnostic quiz analysis, focusing on identifying different types of depression (e.g., anxiety, PTSD).
- **Feature Engineering**: Handles feature extraction and selection for quiz data to optimize model accuracy.
- **Model Documentation**: Documents the rationale behind feature selection, model development, and testing.

Member 2:

- **CNN Model Development**: Leads the design and training of the Convolutional Neural Network (CNN) model for real-time facial expression analysis, experimenting with different architectures to improve emotion detection accuracy.
- **System Integration**: Integrates both the Naive Bayes and CNN models into a cohesive system that functions in real-time.
- **Testing and Validation**: Conducts extensive testing of the integrated system with real-time data and various test cases to ensure robustness.
- **System Optimization**: Focuses on optimizing real-time performance, ensuring efficient processing of facial expressions and quiz data.

Member 3:

- **Frontend Development**: Designs and develops the user-friendly interface for the system, ensuring it's accessible and easy to use for all users.
- **Backend Integration**: Ensures smooth communication between the Python-based backend and the frontend. Handles integration between quiz results, facial analysis, and the MySQL database.
- **User Experience Testing**: Conducts tests to evaluate the usability and overall experience of the interface.
- **Performance Benchmarking**: Benchmarks system performance against existing solutions to identify strengths and areas for improvement.

Member 4:

- **Facial Expression Data Processing**: Implements data preprocessing techniques for real-time video inputs, including noise reduction and normalization for the CNN model.
- **Hyperparameter Tuning**: Leads efforts in hyperparameter tuning for both Naive Bayes and CNN models to achieve optimal accuracy.
- **Model Evaluation**: Evaluates model performance using cross-validation techniques and confusion matrices.
- **System Deployment**: Develops a strategy for deploying the system, considering scalability and cloud-based solutions.

5. Future Aspects: Plan Towards Project Completion

1. Implementation:

The project will proceed in the following stages:

- **Model Integration**:
  - Integrate the Naive Bayes classifier (for quiz-based depression detection) and the CNN (for facial emotion recognition) into the backend using **FastAPI**.
  - Ensure seamless communication between the backend, frontend (ReactJS), and the database (MySQL/MongoDB).
- **Real-time Facial Expression Analysis**:
  - Implement facial expression detection using **OpenCV** and **TensorFlow/Keras**.
  - Ensure the system can process live video inputs from users and produce immediate feedback.
- **Frontend-Backend Synchronization**:
  - Develop a responsive, user-friendly interface using **ReactJS**, where users can take the diagnostic quiz and upload videos for facial analysis.
  - Ensure real-time feedback from both the quiz and facial expression analysis is displayed to the user efficiently.
- **Data Handling and Storage**:
  - Implement robust data storage and retrieval processes in **MySQL/MongoDB** for storing quiz results, video analysis outputs, and user profiles.
  - Ensure efficient handling of real-time data and fast response times.

2. Testing:

Testing is crucial to ensuring the accuracy and performance of the system:

- **Unit Testing**:
  - Perform unit testing for individual components, including the Naive Bayes classifier and CNN model.
  - Test preprocessing steps like facial recognition, feature extraction, and quiz-based decision-making.
- **Integration Testing**:
  - Conduct integration testing between the frontend, backend, and database to ensure smooth data flow and functionality.
- **User Testing**:
  - Deploy the system in a controlled environment for user testing to gather feedback on usability and performance.
  - Monitor real-time emotion recognition and refine the model based on feedback to improve user experience.
- **Performance Testing**:
  - Test the system under various loads to ensure scalability, especially for real-time processing of multiple users.
  - Assess the system’s latency, ensuring facial recognition and quiz processing are handled in real-time without significant delays.

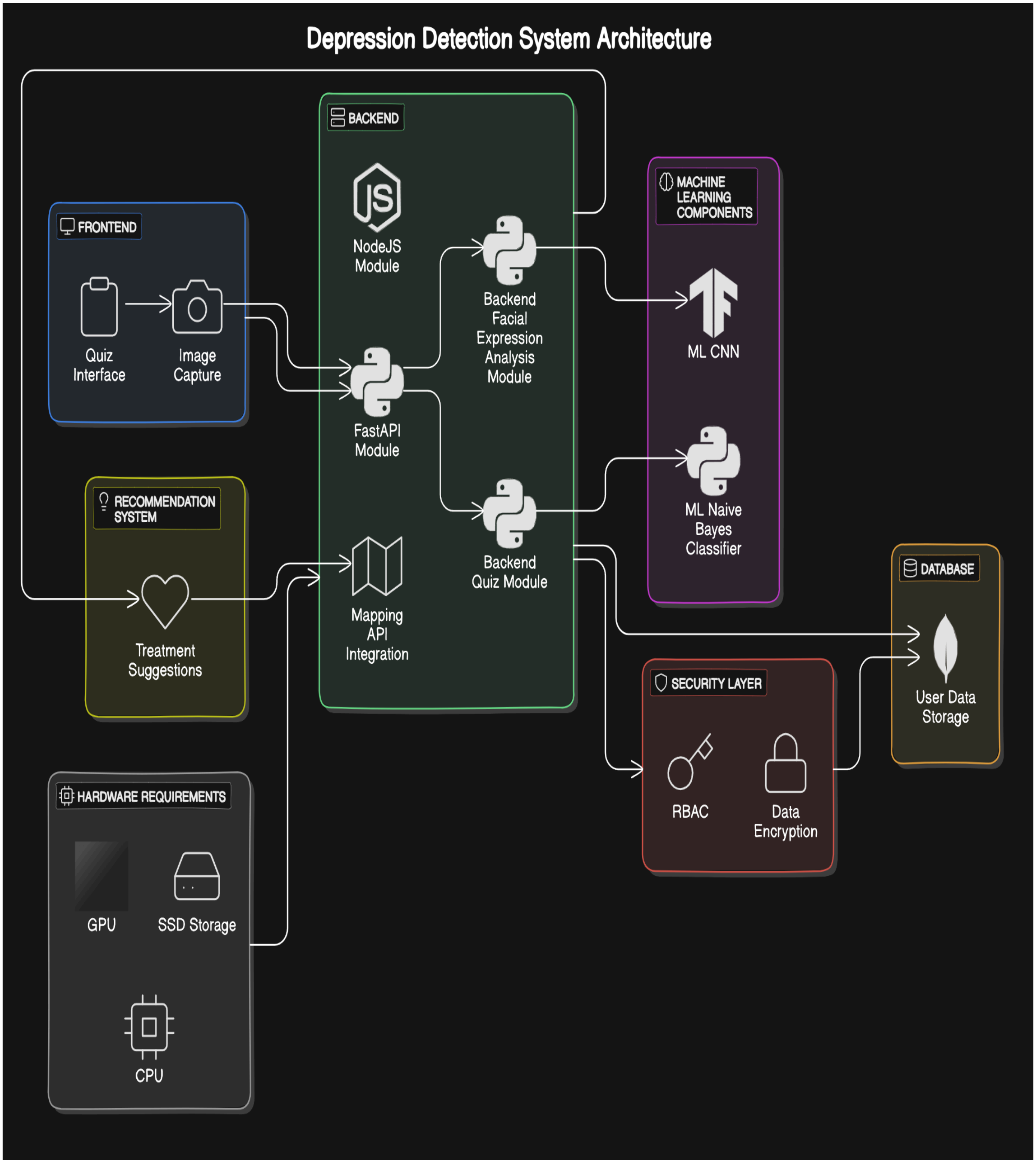
3. Deployment Strategies:

- **Local Deployment for Testing**:
  - Initially, the system will be deployed locally for internal testing. This will allow the team to simulate user interactions and test the system’s functionality in a controlled environment.
- **Cloud Deployment**:
  - For large-scale access, the system will be deployed on cloud platforms (e.g., **AWS**, **Heroku**, or **Azure**) to ensure scalability and availability.
  - Use **Docker** for containerizing the application, making deployment and scaling easier.
- **Data Privacy and Security**:
  - Ensure secure handling of sensitive user data, including quiz responses and facial videos, during deployment. Encrypt data in transit and at rest.
  - Implement authentication and role-based access control (RBAC) to protect user information.
- **Future Enhancements**:
  - Plan for updates to the models based on user feedback and system performance. The models can be fine-tuned for better accuracy and adapted to handle larger, more diverse datasets.

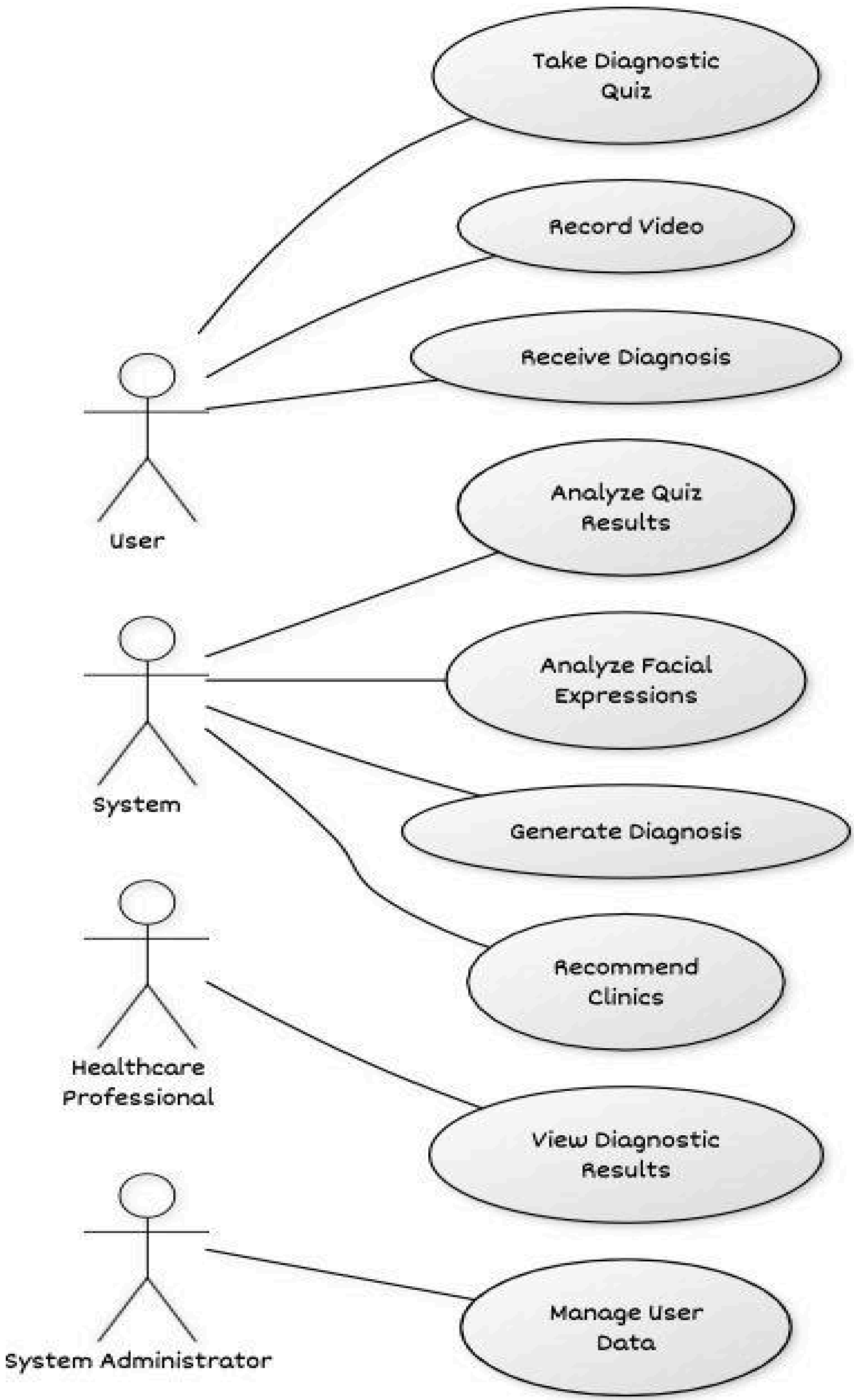
6. DIGRAMS

(a)Entire system

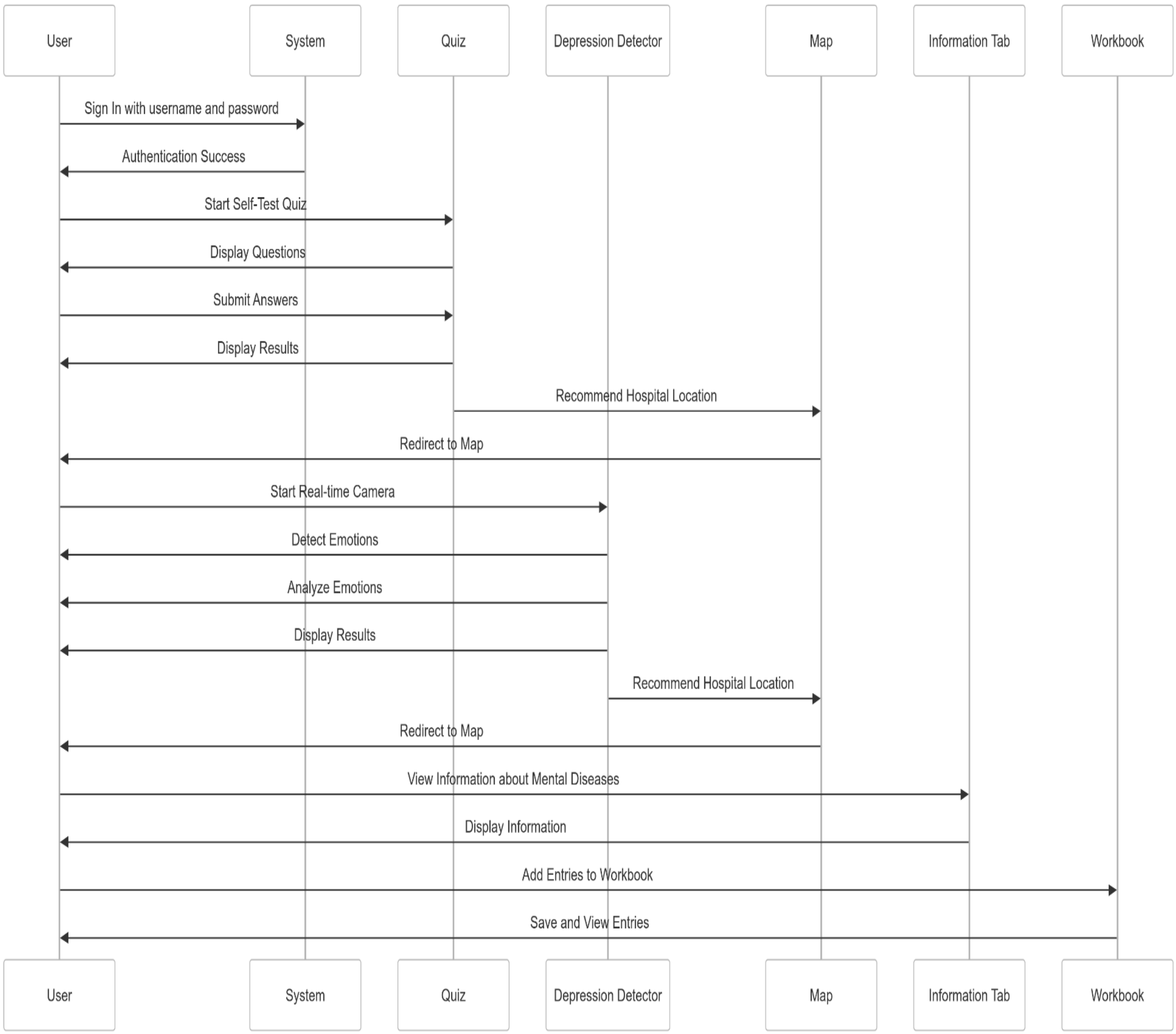
(i) Architecture diagram



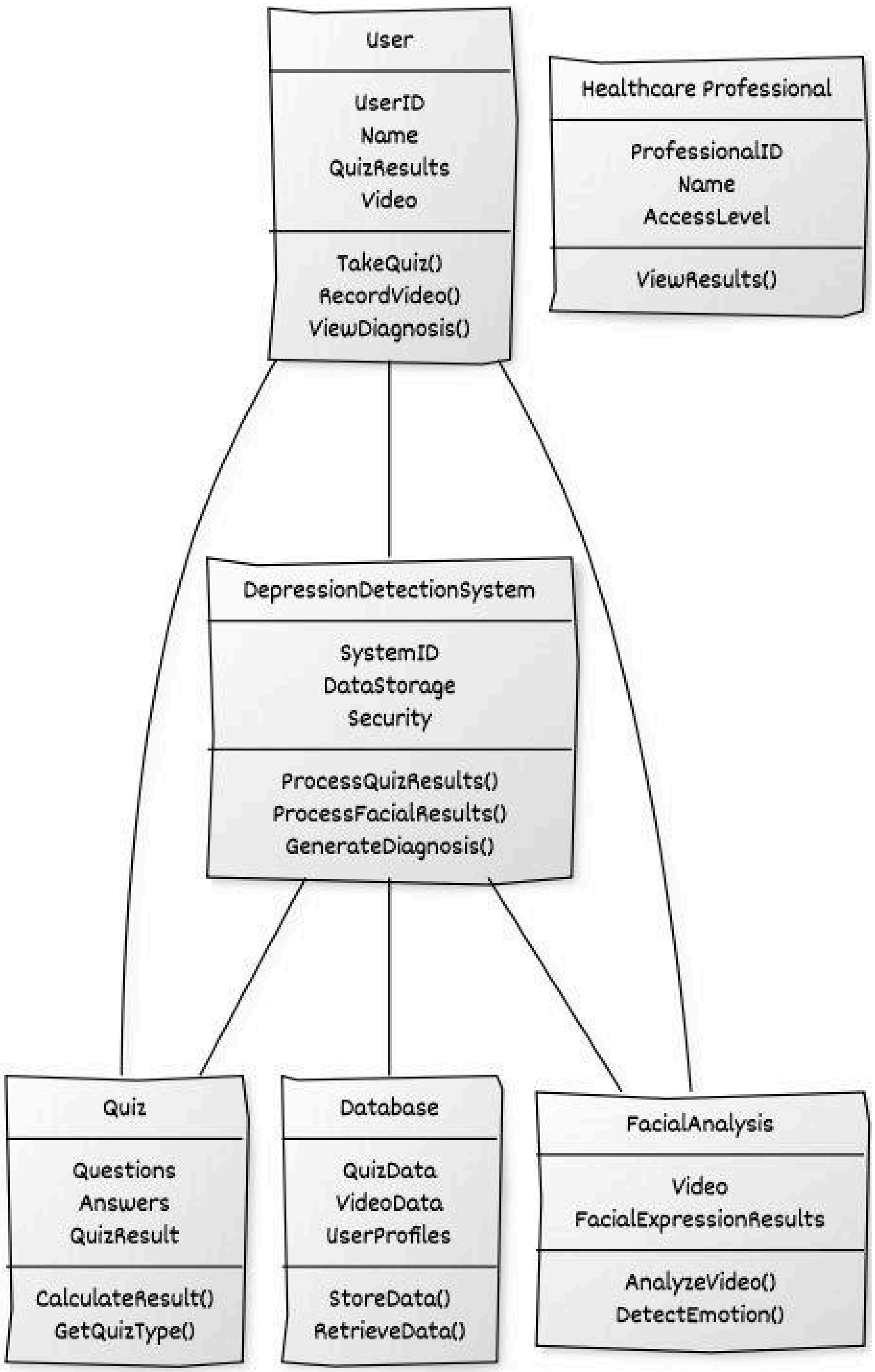
(ii) Use case diagram



(iii)Sequence Diagram



(iv) Class diagram



(v) Activity Diagram