Quality Assurance Plan

AI and Signal Processing Techniques for Parkinson's Finger-Tapping Assessment

Discipline Specific AI Project Assignment

University of Bradford
Faculty of Engineering and Digital Technologies

Content

Purpose of the document

This Quality Assurance Plan (QAP) defines how the DSP-D25 project team will manage quality across all activities related to:

- *analyzing finger-tapping videos from Parkinson's patients*
- *extracting meaningful movement features using signal processing*
- *building and validating an AI model to classify symptom severity (normal, moderate, severe)*

The QAP sets out:

- *working procedures and communication rules*
- *documentation and data control practices*
- *review, testing, and sign-off processes*
- *risk management and reporting routines*

The goal is to ensure that the project achieves its success criteria:

- *≥85% classification accuracy compared to clinical grading*
- *F1-score ≥0.85*
- *clinically relevant features extracted (≥10)*
- *delivery of a functional prototype that is usable on standard hardware*

This is an internal document for the project team, module staff, and client supervisor. It does ***not*** *override university policies or ethical requirements.*

Project Management

2.1 Management Structure

- ***Project Manager:*** *Ebuka*
  - *Coordinates overall work*
  - *Ensures milestones and deliverables are met*
  - *Oversees quality checks and risk monitoring*
- ***Team Members:***
  - *Ryan*
  - *Avyandra*
  - *Elvis*
- ***Client / Domain Stakeholder:***
  - *Prof. Ramzi (Chair / Client) – provides clinical/domain guidance*
- ***Project Sponsor / Module Leader:***
  - *Prof. Kulvinder Panesar – approves the project charter and overall direction*

Each member will have specific technical responsibilities (e.g. signal processing, feature extraction, model development, evaluation), but ***quality ownership is shared****.*

2.2 Decision-Making, Rules, and Rights

- *Technical and design decisions are discussed in weekly meetings.*
- *If the team cannot agree, the Project Manager makes the final call after consulting the client/supervisor.*
- *Any decision that changes the system architecture, dataset usage, or evaluation protocol must be recorded in a short note in the shared Google Drive and mentioned in the next meeting.*

Formal "voting" is not normally needed; if it is, simple majority is used (3 of 4 members).

Communication protocols

3.1 Face-to-Face Meetings

- ***Frequency:***
  - *At least* ***one in-person meeting per week*** *(after labs or lectures).*
- ***Agenda:***

    - *Progress review*
    - *Blockers and risks*
    - *Next sprint tasks*
- ***Minutes:***
    - *One-person (rotating) notes decisions and action items and uploads them to the shared Google Drive.*

3.2 Online Meetings

- ***Tools:*** *MS Teams / Zoom / WhatsApp call.*
- *Used when face-to-face meetings are not possible or before key deadlines.*
- *Quick check-ins for debugging, integration, or urgent issues.*

3.3 Email / Messaging

- ***Primary channel:*** *WhatsApp group chat for rapid coordination.*
- ***Secondary:*** *University email for anything involving supervisors or sharing formal documents.*
- *Any decision affecting the project scope, dataset, or evaluation must be summarized in writing (email or shared doc) so it's traceable.*

3.4 Collaborative Tools

- ***GitHub:***
    - *Central repository for* ***all code****, experiment notebooks, and configuration files.*
    - *Branching model: each new feature/experiment on its own branch; merge to main only after review.*
- ***Google Drive / OneDrive:***
    - *Storage for documents, meeting notes, diagrams, and presentation files.*
- ***Trello / Notion (optional):***
    - *Kanban board for tasks (To Do / In Progress / Done).*

Documentation and data control

4.1 Data Management & Confidentiality

- *Dataset:* ***Private Parkinson's patient dataset****, used under ethical approval; no raw identifying information will be shared outside authorized environments.*
- *Data must only be stored:*
    - *on secure university drives, or*
    - *on password-protected personal machines used for the project.*
- *No dataset files are to be uploaded to public GitHub repositories.*

All team members must respect ethical and confidentiality rules set by the university and supervisors.

4.2 File Naming and Versioning

Documents *(reports, slides, etc.):*
DSP-D25_<DocumentType>_<ShortTitle>_vX.Y.docx
Example: DSP-D25_QAPlan_v1.0.docx
Code and notebooks *follow a consistent structure in GitHub:*

- *data/ – scripts for loading/preprocessing (no raw data if restricted)*
- *features/ – feature extraction scripts*
- *models/ – model definitions and training scripts*
- *experiments/ – notebooks for specific experiments*
- *docs/ – diagrams, notes*

All documents should include:

- *clear title*
- *version number*
- *author(s)*
- *date*

4.3 Revision and Review Process

- *When a document is first drafted → marked as* ***v0.x (Draft)****.*

- *After internal review and edits → upgraded to **v1.0 (Initial final)**.*
- *Later minor updates → **v1.1, v1.2**, etc.*
- *Major rework → **v2.0**.*

*Changes are tracked in a short **History** table at the front of key documents (project charter, QA plan, main report, presentation).*

4.4 Virus and Integrity Checks

- *Any external files or libraries must be sourced from trusted official locations (e.g. PyPI, official repos).*
- *Data transferred via USB must be virus-scanned before use.*

Review Process

5. Review Process

5.1 Schedule and Quality Criteria

Key checkpoints (informal but agreed):

- ***Literature review milestone** – methods, related work understood*
- ***Signal processing pipeline draft** – basic feature extraction working*
- ***First working model** – baseline performance and metrics*
- ***Integration test** – full pipeline runs end-to-end*
- ***Pre-presentation review** – prototype and slides checked*

For each checkpoint, quality is assessed on:

- *Technical correctness*
- *Clarity and reproducibility*
- *Alignment with project objectives*
- *Stability and absence of obvious bugs*

5.2 Process for Deliverables (Internal)

1. ***Drafting:** One team member creates the initial version.*
2. ***Peer Review:** At least one other team member reviews the work (code or document), adding comments and suggestions.*
3. ***Revision:** Author addresses comments and updates the version.*
4. ***Sign-off:** Project Manager checks if it meets the agreed standard.*
5. ***Storage:** Final version stored in GitHub/Drive under the correct folder and naming convention.*

No complicated approval chain is needed, but nothing goes into the final presentation/report without at least one peer review.

Reporting and Monitoring

6.1 Internal Progress Reporting

- ***Weekly updates:***
  - *What was done*
  - *What is blocked*
  - *What is planned for next week*
- ***Sprint summaries:***
  - *At the end of each 2–3-week sprint, the team briefly summarizes progress vs. charter objectives:*
    - *features implemented*
    - *models tested*
    - *performance compared to targets*

These summaries can be simple bullet lists stored in a shared "Progress Log" document.

6.2 Supervisor / Client Updates

- *Use scheduled lab/project supervision slots to:*
  - *demo current progress*
  - *seek feedback*

- *confirm whether the direction is still aligned with expectations*

Any major change suggested by supervisors should be noted and reflected in the next sprint plan.

Risk management

Risk	*Description*	*Likelihood*	*Impact*	*Mitigation*
Data limitations	*Dataset may lack diversity or quality (e.g. limited variation in patients, lighting, skin tones)*	*Medium*	*High*	*Perform thorough exploratory analysis; use preprocessing and augmentation; clearly state limitations in report.*
Technical challenges	*Difficulty implementing signal processing and AI algorithms*	*Medium*	*High*	*Break tasks into smaller subtasks; use existing libraries; pair programming; ask for supervisor support early.*
Time constraints	*Delays in tasks leading to rushed work near deadlines*	*High*	*High*	*Plan realistic sprints; assign clear owners; prioritize core pipeline before extras; avoid last-minute scope changes.*
Model performance issues	*Overfitting, poor generalization to unseen samples*	*Medium*	*High*	*Use validation and test splits, regularization, data augmentation, and multiple model trials; monitor metrics.*
Tooling / environment issues	*Inconsistent environments across laptops*	*Medium*	*Medium*	*Use shared requirements.txt; document Python versions and key libraries; consider using Google Colab for heavy training.*

Publication and dissemination

*This is a **student project**, so dissemination is simple:*

- *Any external presentation (poster, blog, LinkedIn post) must:*
  - *be agreed by the team,*
  - *respect data confidentiality and ethics,*
  - *acknowledge the University of Bradford and supervisors.*

No confidential dataset details or raw patient images/identifiers are to be shared.

*Project outputs (e.g. code) can be shared on GitHub **only if**:*

- *the dataset itself is not included,*
- *any sensitive info is removed,*
- *supervisors are comfortable with it being public.*

Deliverables and Quality Standards (Internal View)

Key internal deliverables (based on charter milestones :

1. ***Project Charter*** *– completed and agreed (already done).*
2. ***Literature Review & Design Notes*** *– core AI and signal processing methods identified.*
3. ***Signal Processing Pipeline*** *– scripts to process videos and extract ≥10 clinically relevant features.*
4. ***ML Model(s)*** *– trained models for severity classification with target metrics:*
    - *Accuracy ≥85%*
    - *F1-score ≥0.85*
5. ***Integrated Prototype*** *– functional pipeline from input video/sequence to classification output.*
6. ***Final Presentation & Report*** *– clearly communicate method, results, limitations, and future work.*

Quality standard for each deliverable:

- *Technically sound and reproducible*
- *Clear structure and explanation*
- *Aligned with project aims and constraints*
- *Checked by at least one other team member before being treated as "final"*