



UNIVERSIDADE D
COIMBRA

Gonçalo Cardoso Nogueira

**EPIWEB: A WEB-BASED APPLICATION FOR
DETECTING AND PREDICTING EPILEPTIC
SEIZURES**

Dissertation in the context of the Master in Biomedical Engineering, specialization in Clinical Informatics and Bioinformatics, advised by Professor César Teixeira, Phd and Professor Nuno Laranjeiro, Phd and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

September 2023



DEPARTAMENTO DE
ENGENHARIA INFORMÁTICA
FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

Gonçalo Cardoso Nogueira

EPIWEB: A WEB-BASED APPLICATION FOR DETECTING AND PREDICTING EPILEPTIC SEIZURES

**Dissertation in the context of the Master in Biomedical Engineering,
specialization in Clinical Informatics and Bioinformatics, advised by
Professor César Teixeira, Phd and Professor Nuno Laranjeiro, Phd and
presented to the Department of Informatics Engineering of the Faculty of
Sciences and Technology of the University of Coimbra.**

September 2023



DEPARTAMENTO DE
ENGENHARIA INFORMÁTICA
FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

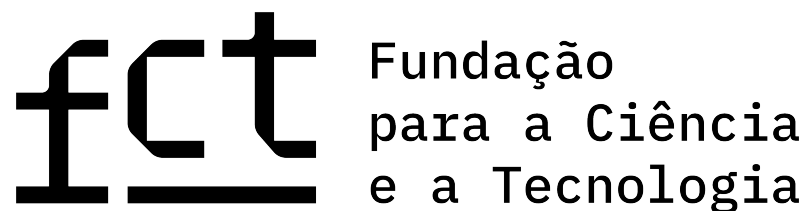
Gonçalo Cardoso Nogueira

EPIWEB: APLICAÇÃO WEB PARA DETEÇÃO E PREVISÃO DE CRISES EPILÉTICAS

Dissertação no âmbito do Mestrado em Engenharia Biomédica, especialização em Informática Clínica e Bioinformática, orientada pelo Professor Doutor César Teixeira e pelo Professor Doutor Nuno Laranjeiro e apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Setembro 2023

This work is funded by FCT- Foundation for Science and Technology, I.P., within the scope of the projects: CISUC - UID/CEC/00326/2020 with funds from the European Social Fund, through the Regional Operational Program Centro 2020; and project RECoD - PTDC/EEI-EEE/5788/2020 financed with national funds (PIDDAC) via the Portuguese State Budget.



Acknowledgements

Having reached the completion of this dissertation and Biomedical Engineering Master Degree, I would like to address in Portuguese all the recognitions and acknowledgements.

O último ano refletiu-se em inúmeros e exigentes desafios e momentos de stress que resultaram em muitas noites sem dormir, dúvidas, momentos de solidão e ansiedade. Este trabalho representa o culminar de uma jornada de cinco anos que agora termina numa forma que não poderia imaginar. No entanto, o encerrar deste caminho tornou-se possível apenas com o apoio e a companhia de pessoas excecionais, a quem devo a minha profunda gratidão .

Em primeiro lugar, quero expressar os meus sinceros agradecimentos ao Professor César Teixeira e ao Professor Nuno Laranjeiro. A vossa orientação e disponibilidade conduziram-me nesta tarefa exigente. As vossas sugestões e ajuda incansáveis foram verdadeiramente essenciais.

Quero também expressar a minha profunda gratidão a todos os meus amigos, aqueles que também estiveram ao meu lado nos bons e maus momentos. Aprendemos, rimos e apoiámo-nos de inúmeras formas ao longo destes últimos cinco anos. Sem vocês, isto não teria sido possível.

A ti, Catarina, um agradecimento especial...

À minha família, sois o alicerce sobre o qual construí a pessoa que sou hoje. À minha Mãe, Pai e Irmãos, agradeços-vos da forma mais pura pelo vosso apoio inabalável, ensinamentos imensuráveis e compreensão profunda. Especialmente nestas últimas semanas, tornaram este desfecho possível. Levarei comigo aquilo que me transmitiram ao longo de toda a minha vida. Mais uma vez, obrigado por me tornarem na pessoa que sou.

A Coimbra, a cidade que testemunhou o meu crescimento ao longo destes cinco anos. Muito como a própria vida, com os seus altos e baixos, desde o Mondego, a Alta, a Venâncio Rodrigues e a Antero de Quental, será para sempre o que agora tenho o privilégio de chamar de Casa.

Thank you all for not giving up on me.

Abstract

This dissertation explores the major obstacles presented by epilepsy, a neurological disorder that causes frequent seizures. The study presents EPIWEB, a user-friendly web platform devised to improve the identification and forecasting of epileptic seizures. By utilizing web development, data analysis, and machine learning techniques, EPIWEB delivers an easy-to-use interface to create, arrange, and scrutinize EEG data.

EPIWEB provides researchers with key tools for visualising raw EEG data, extracting features, and utilising classification algorithms. EPIWEB's architecture, grounded in the Django framework and tailored for scalability, guarantees smooth data processing integration. Importantly, the platform emphasises user authentication and data protection to secure sensitive patient information.

This dissertation explores in detail the core concepts that form the basis of predicting epileptic seizures. It encompasses the analysis of EEG signals, feature extraction, and an extensive evaluation of past research, including the important EPILAB project.

This work addresses potential risks and implements strategies to mitigate challenges arising from the ambitious project scope and evolving requirements. EPIWEB undergoes thorough validation against functional requirements, demonstrating its capacity and potential in analyzing and predicting epileptic seizures. The platform's proficiency in managing patient data, conducting research, and displaying EEG information is convincingly demonstrated.

In conclusion, EPIWEB shows potential for advancing the field of epileptic seizure detection and prediction. Its noteworthy attributes makes it a valuable resource for academics and professionals in this pivotal field.

Keywords

Epilepsy, Web Application, EEG Signal Processing, Machine Learning, Seizure Detection, Seizure Prediction.

Resumo

Esta dissertação explora as principais adversidades levantadas pela epilepsia, um distúrbio neurológico que causa a frequente ocorrência de convulsões. O estudo apresenta o EPIWEB, uma plataforma web de fácil utilização desenvolvida para melhorar a detecção e a previsão de crises epilépticas. Ao recorrer a *Web Development*, análise de dados e técnicas de aprendizagem computacional, o EPIWEB oferece uma interface amigável para criar, organizar e analisar dados de EEG e estudos sobre eles.

O EPIWEB fornece a investigadores ferramentas importantes para visualizar dados brutos de EEG, extrair *features* e modelar algoritmos de classificação. A arquitetura do EPIWEB, baseada na *framework* Django, garante uma fácil integração dos algoritmos de processamento de dados. É importante realçar que a plataforma permite a autenticação de utilizadores e a proteção de dados para proteger os dados dos pacientes.

Esta dissertação explora em detalhe os conceitos fundamentais que formam a base da previsão de crises epilépticas. Abrange a análise de sinais EEG, extração de características e uma exploração do atual estado de arte, incluindo o importante projeto EPILAB.

Ao longo do trabalho são abordados potenciais riscos para a implementação e as estratégias para mitigar os desafios decorrentes do desenvolvimento promissor do projeto e dos seus requisitos. O EPIWEB passa por uma validação cuidada face aos requisitos funcionais, demonstrando potencial na análise e previsão de crises epilépticas. A eficiência da plataforma na gestão de dados de pacientes, na construção de estudos e na exibição gráfica de informações de EEG é demonstrada de forma convincente.

Concluindo, o EPIWEB mostra potencial para contribuir no campo da detecção e previsão de crises epilépticas. As suas capacidades revelam um recurso valioso para fins académicos e profissionais neste campo da epilepsia.

Palavras-Chave

Epilepsia, *Web Application*, Processamento de Sinal EEG, Aprendizagem Computacional, Detecção de Crises Epilépticas, Previsão de Crises Epilépticas.

Contents

1	Introduction	1
1.1	Problem and Motivation	1
1.2	Context	2
1.3	Research Goals	3
1.4	Outline	4
2	Background Concepts	7
2.1	Epilepsy	7
2.1.1	Definition and Classification	7
2.1.2	EEG Signal	9
2.2	Seizure Prediction	10
2.2.1	Main concept	10
2.2.2	Detection vs. Prediction	11
2.3	The Role of Projects like EPILEPSIAE	11
2.3.1	Databases	11
2.3.2	EPILEPSIAE and EPILAB	12
3	State of Art	15
3.1	Seizure Prediction Pipeline	15
3.1.1	Pre-processing	16
3.1.2	Feature Extraction	16
3.1.3	Feature Selection	16
3.1.4	Classification	17
3.1.5	Post-processing and Evaluation	17
3.2	Underlying Architecture of EPILAB	17
3.2.1	Overview	18
3.2.2	Limitations	20
3.3	Initial Prototype of EPIWEB	21
3.3.1	Architecture Overview	22
3.3.2	Functionalities	23
3.3.3	Early Exploration and Limitations	23
3.4	Existing Web Platforms with Similar Features	23
4	Methodology	25
4.1	Project Management Methodology	25
4.2	Planning	27
4.3	Risk Analysis	27
5	Requirements Analysis	31

5.1	Actors	31
5.2	User Stories and Functional Requirements	32
5.2.1	User Stories Examples	33
5.2.2	Functional Requirements	34
5.3	Non-functional Requirements	36
5.4	User Interface	37
5.4.1	Authentication	38
5.4.2	Home Page	38
5.4.3	Raw Data	39
5.4.4	Studies	40
6	Technologies	43
6.1	Backend	43
6.1.1	Integration of EPILAB Scripts	43
6.1.2	Backend Frameworks	44
6.1.3	Data Persistence	46
6.2	Frontend	46
6.2.1	Fundamental Concepts	47
6.2.2	Data Visualization Libraries	48
6.2.3	Frontend Frameworks	48
7	Architecture	51
7.1	Technologies	51
7.2	MVT	52
7.3	C4 Model	54
7.4	Entity-Relationship Diagram	59
8	Development	63
8.1	Important Practices	63
8.2	Challenges	66
8.3	Outcomes	67
9	Validation and Verification	81
10	Conclusion	85
10.1	Future Work	86
	References	87
	Appendix A List of User Stories	95
A.1	User Stories	95
A.1.1	Epic Story 1: Authentication	95
A.1.2	Epic Story 2: Enhancing User Management	96
A.1.3	Epic Story 3: Raw Data Management	97
A.1.4	Epic Story 4: Studies Management	99

Acronyms

ANN Artificial Neural Network.

CISUC Centre for Informatics and Systems of the University of Coimbra.

CSRF Cross-Site Request Forgery.

CSS Cascading Style Sheets.

DEI Department of Informatics Engineering.

DOM Document Object Model.

E-R Entity-Relationship.

ECG Electrocardiogram.

EEG Electroencephalogram.

HTML HyperText Markup Language.

HTTP Hypertext Transfer Protocol.

ILAE International League Against Epilepsy.

MDS Multidimensional Scaling.

MVC Model-View-Controller.

MVT Model-View-Template.

ORM Object Relational Mapping.

PCA Principal Components Analysis.

SVM Support Vector Machine.

UI User Interface.

US User Story.

List of Figures

2.1	The international 10-20 system used for electrode placement. From [Sazgar and Young, 2019].	9
2.2	Different episodes in a Electroencephalogram (EEG) signal, including interictal, preictal, ictal and postictal. From [Cui et al., 2018]. . .	10
2.3	Illustrates the conceptual framework of the planned high-level architecture for EPIWEB and highlights the crucial role that EPILEPSIAE’s EPILAB and database will play in its development process.	13
3.1	Flowchart illustrating the standard pipeline for seizure prediction. Adapted from [Bou Assi et al., 2017]	15
3.2	EPILAB flowchart. (A) The initiation of a new study involves creating it from raw data or precomputed feature data; (B) When commencing a study from raw data, EEG and/or ECG features need to be calculated; (C) Using either imported or computed features, prediction algorithms can be formulated and assessed; (D) Imported or computed features can undergo dimensionality reduction; (E) Throughout the study, data and outcomes can be visually or textually presented. From [Teixeira et al., 2011].	19
3.3	List of features that can be computed by EPILAB. From [Teixeira et al., 2011].	20
3.4	Model-View-Controller design. From [javatpoint].	22
4.1	Development Methodology: Iterative Approach with Weekly Feedback Loops (Agile based).	26
4.2	Gantt diagram.	27
5.1	EPIWEB - Authentication page sketch.	38
5.2	EPILAB - Main GUI.	39
5.3	EPIWEB - Raw data fast details on list page sketch.	39
5.4	EPILAB - Information of the raw data GUI.	39
5.5	EPIWEB - Raw data visualization sketch.	40
5.6	EPILAB - Raw data visualization.	40
5.7	EPIWEB - Study creation page sketch.	40
5.8	EPILAB - Feature extraction study creation page GUI.	40
5.9	EPIWEB - Features selection study creation page sketch.	41
5.10	EPILAB - Feature selection study creation page GUI.	41
5.11	EPIWEB - Extraction study information page sketch.	41
5.12	EPILAB - Feature extraction study information page GUI.	41

7.1	Django Architecture seen as Model-View-Template (MVT). From [Pinkham, 2016].	54
7.2	Enter Caption. From [C4 Model].	55
7.3	Context diagram.	56
7.4	Containers diagram.	57
7.5	Components diagram.	58
7.6	Components diagram for Core component.	58
7.7	Entity-Relationship model diagram.	61
7.8	Physical model diagram.	62
8.1	EPIWEB - Landing page.	68
8.2	EPIWEB - Login page.	68
8.3	EPIWEB - Create a new account page.	68
8.4	EPIWEB - User authenticated home page.	69
8.5	EPIWEB - Patient raw data management page.	70
8.6	EPIWEB - Patient raw data management page.	70
8.7	EPIWEB - Patient raw data information page.	71
8.8	EPIWEB - Patient raw data graphical visualization.	72
8.9	EPIWEB - Studies management page.	73
8.10	EPIWEB - Extraction study creation page.	73
8.11	EPIWEB - Study creation confirmation form.	74
8.12	EPIWEB - Extraction study information page.	74
8.13	EPIWEB - Extraction study edit form.	75
8.14	EPIWEB - Extraction study graphical visualization.	75
8.15	EPIWEB - Selection/Reduction study creation page.	76
8.16	EPIWEB - Selection/Reduction study information page.	76
8.17	EPIWEB - Classification study creation page.	77
8.18	EPIWEB - Classification study information page.	77
8.19	EPIWEB - Real time study queue update system on home page.	78
8.20	EPIWEB - Real time study queue update system on notifications element.	78
8.21	EPIWEB - Classification models page.	79
8.22	EPIWEB - Sharing raw data/studies system.	79

List of Tables

4.1	Risks classification.	30
5.1	List of functional requirements.	36
6.1	Backend frameworks usage on 2023 according to Statistics and Data. From [Statistics & Data, 2023].	45
6.2	Frontend frameworks usage on 2022 according to State of JS survey. From [StateOfJs].	49
9.1	List of functional requirements with the indication if they were developed.	82

Chapter 1

Introduction

This chapter contains a briefly overview of the problem and the motivation behind this study, section 1.1, as well as the context surrounding it, section 1.2. The main objectives and expected contributions are also discussed (section 1.3). Lastly, the structure of this document is explained (section 1.4).

1.1 Problem and Motivation

Forecasting and identifying seizures is a realm with considerable potential. If we can correctly anticipate seizures, it will revolutionize the management of epilepsy, greatly elevating the quality of life for those affected by it.

Traditional practices for EEG analysis and seizure prediction frequently employ unique software platforms, such as the MATLAB-based EPILAB toolbox. Although EPILAB boasts advanced features for analyzing brainwave signals, it also faces obstacles related to availability, intuitiveness, and expandability. The requirement to acquire and maintain MATLAB licenses, combined with the processing constraints of users' computers, can impede the broad acceptance of EPILAB and restrict its capacity for instantaneous predictive prowess.

This is where EPIWEB steps in with a fresh perspective and a clear objective: using a web application, technology can be used to bridge the gap between analysing EEG signals and predicting epileptic seizures. Migrating the features of the EPILAB toolbox to an web-based environment will help counter the constraints associated with conventional desktop applications. The migration underway will improve accessibility, user experience, and adaptability in EEG signal analysis and seizure detection, benefiting both experts and people afflicted.

The increasing prominence of digital platforms has created a unique chance

to level the playing field in the domain of medical evaluations. This project goal is driven by the conviction that an online application can reshape the manner in which epilepsy is managed, via a straightforward user interface, easy data processing, and forecasting models. By utilizing the current trend for online tools and the flexibility of Python programming languages, this project operates under the premise that reliable seizure predictions are easily accessible with just a few clicks.

1.2 Context

Epilepsy, a chronic neurological disorder, casts a shadow of uncertainty over the lives of millions worldwide. The hallmark of epilepsy lies in its unpredictability, with seizures emerging suddenly and disrupting daily activities. These seizures, characterized by abnormal electrical activity in the brain, can range from momentary lapses in awareness to convulsive episodes. This complex and multifaceted condition demands not only accurate diagnosis and treatment but also proactive measures to anticipate and manage its impact.

Within the realm of epilepsy management, the quest for reliable seizure prediction has assumed paramount importance. The capacity to predict an upcoming seizure offers significant potential for transformation, enabling those with epilepsy, their carers, and healthcare experts to take timely action. This pursuit has given rise to an interdisciplinary convergence, where medical science intersects with data processing and machine learning.

Machine learning, a subset of artificial intelligence, has emerged as a promising approach to predicting seizures. By leveraging advanced algorithms, machine learning seeks to unravel the intricate patterns embedded within EEG signals. These EEG signals, recordings of the brain's electrical activity, encapsulate valuable insights into the underlying neurological processes, offering a window into the brain's intricate workings during seizures.

Several databases have become available over time, to address these challenges, such as the one from the University of Freiburg, the University of Bonn, the Children's Hospital Boston (CHB-MIT) and the European Database on Epilepsy (EPILEPSIAE). The EPILEPSIAE database was made publicly available in 2012 and it contains long-term EEG recordings, extensive metadata, and standardized annotations, which will aid in the comparison and reproducibility of studies in the field.

EPILAB is a software package developed using Matlab to help community

study the prediction of epileptic seizures, product of the European project EPILEP-SIAE.

1.3 Research Goals

The primary goal of this work is to develop an innovative web application EPIWEB that leverages the capabilities of the EPILAB toolbox while addressing its limitations. The project aims to provide an accessible platform for processing EEG signals and predicting epileptic seizures. To achieve this overarching objective, it was defined specific research goals that guide the approach and development process.

- **Adaptation of EPILAB to Web Environment:** The first research goal involves adapting the EPILAB toolbox to a web-based environment. This includes redesigning the user interface, optimizing the algorithms for web execution, and ensuring compatibility with modern web technologies.
- **Implementation of EEG Signal Processing Algorithms:** It is aimed to implement a range of EEG signal processing algorithms within the EPIWEB application. This involves replicating the functionalities offered by EPILAB into web-compatible code, allowing users to process and analyze EEG data effectively.
- **Development of Predictive Models:** Building upon the foundation of processed EEG data, it is intended to develop machine learning predictive models that can accurately forecast epileptic seizures. These models will consider both binary (2-class) and multiclass (4-class) classification scenarios, accommodating different diagnostic needs.
- **Signal Analysis and Visualization**
- **Creation of User-Friendly Web Interface:** An essential aspect of EPIWEB is its user interface. The project is committed to designing an intuitive and user-friendly web interface that enables users to interact with EEG data, configure parameters for analysis, visualize results, and access predictive insights.
- **Validation and Evaluation of Performance:** EPIWEB plans to assess the accuracy of predictive models, compare results with existing methods, and gather feedback from domain experts and potential users.

By achieving these research goals, this project seeks to offer a new solution that combines the potential of EEG signal analysis with the accessibility and scalability of web technologies. In the following chapters, this dissertation will examine the methodology, technical specifics, experimental outcomes, and implications of EPIWEB, presenting a comprehensive exploration of this approach.

1.4 Outline

Beyond the introduction, this document is divided into 8 chapters that aim to clarify the development of the entire work.

The Chapter 2, **Background Concepts**, provides an overview of epilepsy, the EEG signal, seizure prediction concepts, and efforts in the field such as the EPILEP-SIAE and EPILAB projects.

The Chapter 3, **State of Art**, showcases the state of art concerning EEG-based seizure prediction. It also identifies and explains the features and architectures from EPILAB and a first prototype of EPIWEB, previous developed.

The Chapter 4, **Methodology**, outlines the development methodology employed in this project along with its corresponding planning. It also identifies potential risks associated with the development of the platform.

The Chapter 5, **Requirements**, User Stories are introduced, along with a comprehensive overview of all the requirements that the platform aims to address. This chapter also take a closer look to both functional and non-functional requirements, examining, evaluating, prioritizing each of them. It also provides a comprehensive breakdown of the User Interface (UI) sketches that have been designed for the user interface of the EPIWEB frontend system. It is taken into consideration the windows that have been developed for the EPILAB toolbox.

The Chapter 6, **Technologies**, multiple technologies, libraries, and frameworks are scrutinized and explained for building the backend and frontend. Their benefits are explored in relation to the platform's development requirements.

The Chapter 7, **Architecture**, introduces the selected specifications and technologies for the EPIWEB architecture. It is also provide an overview of both the conceptual and physical models for the database.

The Chapter 8, **Development**, presents the risk analysis, the challenges and difficulties that were encountered. Some outcomes from this work are also presented.

The Chapter 9, **Validation and Verification** elucidates the process of validation and verification of the platform, employing also the alignment to the functional requirements previous depicted.

The Chapter 10, **Conclusion**, presents the final thoughts and the future prospects and potential directions for EPIWEB.

Chapter 2

Background Concepts

This chapter serves as an introduction to the fundamental concepts essential for contextualizing the subsequent content of this document. It begins by providing an overview of epilepsy as a medical condition, encompassing its clinical definition, classification, and the pivotal role of data such as Electroencephalogram (EEG) in diagnosis, detection, treatment, and prediction (section 2.1.1). Subsequently, this document explores the role of computational learning in the field of epilepsy, detailing numerous challenges and technologies particular to this area (section 2.2). To conclude, noteworthy collaborative initiatives addressing these challenges, such as EPILEPSIAE and the internally developed EPILAB under the first initiative, are highlighted (section 2.3).

2.1 Epilepsy

Epilepsy stands as a profound challenge within the realm of neurological disorders. It is a chronic brain disorder that results in recurrent seizures. Seizures can be mild or severe and may cause loss of consciousness or control of bodily functions. It is caused by excessive electrical activity in brain cells and affects 50 million people worldwide. A person is diagnosed with epilepsy after having two or more unprovoked seizures [World Health Organization, 2022].

2.1.1 Definition and Classification

The International League Against Epilepsy (ILAE) defines epilepsy as a "chronic brain disorder characterized by an enduring predisposition to generate epileptic seizures and by the neurobiological, cognitive, psychological, and social consequences of this condition" [Fisher et al., 2014].

However, this and other ones conceptual definitions fail to encompass all clinical scenarios comprehensively. Therefore, ILAE Task Force proposed an operational definition of epilepsy [Fisher et al., 2014]. According to this operational definition, epilepsy is characterized as a brain disorder marked by any of the following conditions:

- The occurrence of a minimum of two unprovoked seizures with a temporal gap of 24 hours or more between them.
- A single unprovoked seizure, coupled with a probability of experiencing additional seizures, akin to the general recurrence risk (minimum 60%) following two unprovoked seizures, anticipated within the subsequent 10 years.
- Diagnosis of a specific epilepsy syndrome.

The **classification** of epileptic seizures forms a cornerstone of understanding the complexity of epilepsy and starts by identifying whether the initial manifestations (onset) of the seizure are focal, generalized, or unknown.

- **Focal** (partial) seizures: originate in specific brain areas and may propagate to involve the entire brain.
- **Generalized** seizures: involve both hemispheres from onset.

The ILAE's classification efforts aim to create a common language for health-care professionals and researchers, ensuring that individuals' diverse experiences with epilepsy can be understood and effectively communicated. By categorizing epilepsy types, including seizure types and epilepsy syndromes, these classifications contribute to accurate diagnosis and targeted treatment strategies [Berg et al., 2010; Fisher et al., 2014].

Epilepsy is not limited by demographics and affects individuals of all ages and from all geographical backgrounds. Epilepsy, with a global prevalence of approximately 1%, is considered one of the most common neurological disorders [Kobau et al., 2015]. The impact of this condition extends beyond the physical realm, as individuals with epilepsy frequently encounter challenges in psychosocial domains. The stigma surrounding epilepsy can foster feelings of isolation and discrimination, compelling the need for a comprehensive approach to care that addresses both medical and psycho-social aspects [Jacoby and Baker, 2008].

2.1.2 EEG Signal

EEG is an important tool for studying brain activity and understanding complex neurological processes. EEG signals record the electrical activity of the brain's neurons, giving us useful information about cognitive functions, emotions, and conditions like epilepsy [Niedermeyer and da Silva, 2005].

To acquire EEG signals, electrodes are placed on the scalp to detect and record the voltage fluctuations produced by brain activity, as depicted on figure 2.1.

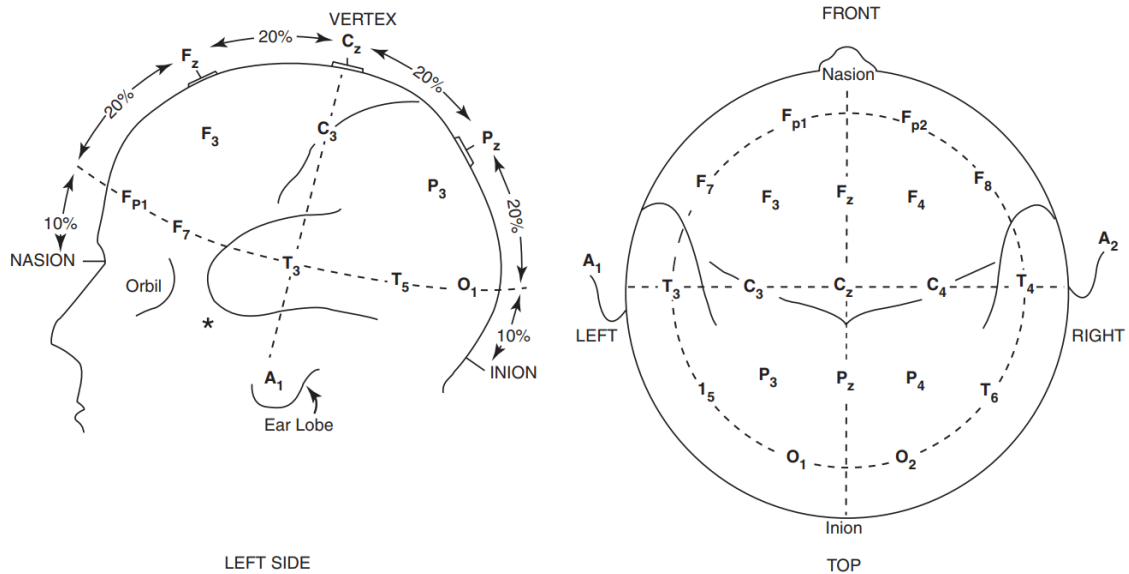


Figure 2.1: The international 10-20 system used for electrode placement. From [Sazgar and Young, 2019].

EEG signals are a time-series data that appear as a mix of rhythmic oscillations and momentary events called event-related potentials (ERPs). It illustrates voltage fluctuations over space and time, demonstrating significant changes in electrical potentials, that originated from the excitatory and inhibitory postsynaptic potentials. So, these signals exhibit diverse frequency components, ranging from slow delta waves to fast gamma oscillations [Acharya et al., 2013; Buzsáki et al., 2012].

Given that epileptic seizures stem from disruptions in the brain's electric and physiological system, leading to abrupt and synchronized electrical discharges among groups of neurons, it becomes evident why the EEG has emerged as the principal clinical instrument for managing patients afflicted by epilepsy [Acharya et al., 2013].

In clinical research, the seizure process has four phases that can be identified through an EEG signal: interictal, preictal, ictal, and postictal. The ictal phase is

the outbreak of epilepsy, while the preictal and postictal phases are before and after it, respectively. The interictal phase is between seizures, and EEG signals during this phase lack seizure symptoms. See figure 2.2 for an overview of EEG signal divisions [Cui et al., 2018].

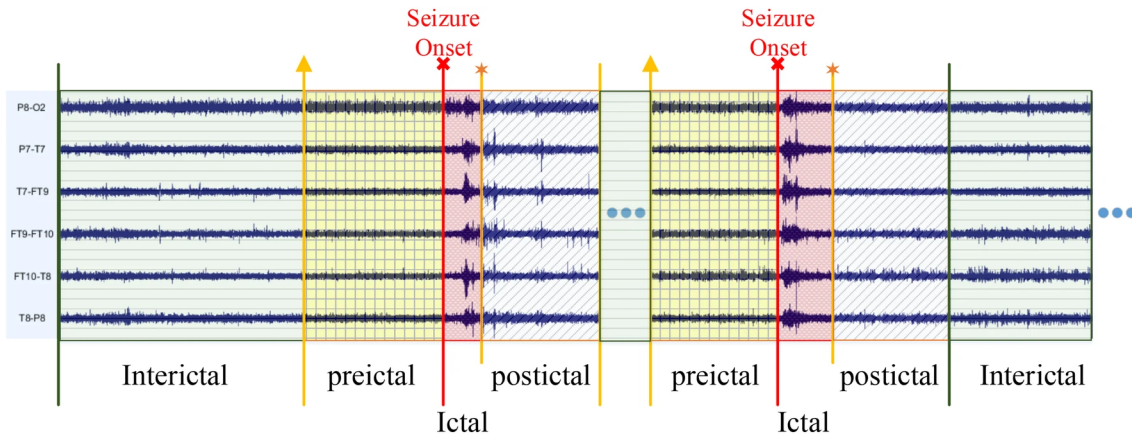


Figure 2.2: Different episodes in a EEG signal, including interictal, preictal, ictal and postictal. From [Cui et al., 2018].

EEG research places great importance on seizure prediction, particularly in the management of epilepsy [Gotman, 1982]. EEG signals contain complex patterns associated with imminent seizures. Sophisticated signal processing techniques, such as time-frequency analysis and machine learning, are utilized to identify the subtle changes preceding seizures. Such insights hold potential for developing predictive models and personalized treatments [Mormann et al., 2007].

2.2 Seizure Prediction

2.2.1 Main concept

Machine learning techniques have rapidly integrated into the medical field, leading to a significant transformation in recent times. Epilepsy has also benefited from these advances, like other fields. The use of classification models to predict seizures has high potential for enhancing the quality of life of individuals with epilepsy [Daoud and Bayoumi, 2019].

It is a challenge that can be seen as a classification task, which involves distinguishing between preictal and interictal brain states. An alarm is triggered when the preictal state, which signals an impending seizure, is detected in the middle of the prevailing interictal states [Daoud and Bayoumi, 2019]. You can also see this concept illustrated in figure 2.2. The predicted time corresponds to the period

before the seizure onset (seizure beginning), when the preictal state is identified. This can give them enough time to prepare, take preventive measures, or seek assistance. Importantly, this predictive system must maintain a stringent control over the number of false alarms generated to prevent undue stress for the patient [Bou Assi et al., 2017].

2.2.2 Detection vs. Prediction

An additional area of research, which is parallel but separate, concentrates on detecting seizures. The algorithms in this field aim to quickly detect the beginning (onset) of a seizure. Despite these algorithms not giving the patient any time to react - unlike seizure prediction algorithms - they help medical professionals to locate the source of the epilepsy and decide on appropriate treatments. Both methodologies for detection and prediction can be combined in closed-loop intervention systems, but prediction stands out as it enables even more timely interventions [Ramgopal et al., 2014]. At its core, seizure prediction aims to identify the preictal phase, while seizure detection focuses on recognising the precise onset of a seizure. In principle, seizure detection can be regarded as a substudy of the more extensive framework of the seizure prediction problematic. [Ren et al., 2022].

2.3 The Role of Projects like EPILEPSIAE

2.3.1 Databases

Difficulties in evaluating, standardising, and reproducing studies have been encountered in the field of seizure prediction. A lack of high-quality, long-term EEG data has contributed to these challenges. Several databases, including those from collaborative projects, have become available to address these challenges and are shaping the understanding, tools, and methodologies employed in epilepsy research and patient care. Notable examples, that are widely used, include:

- University of Bonn: a publicly accessible EEG database provided by the University of Bonn, compiled by Andrzejak et al [Andrzejak et al., 2001]. This database comprises five distinct datasets, each characterized by specific properties and data collection conditions.
- European Database on Epilepsy (EPILEPSIAE): it is part of the EU-funded EPILEPSIAE project. It contains high-quality, long-term EEG data enriched

with clinical metadata, collaboratively managed by epilepsy centers in Coimbra, Paris, and Freiburg.[Ihle et al., 2012b]

- Children’s Hospital Boston (CHB-MIT): also known as the PhysioNet EEG dataset (36), consists of EEG recordings from 24 pediatric patients. These recordings were acquired at Boston Children’s Hospital [Goldberger et al., 2000].
- American Epilepsy Society: intracranial EEG (iEEG) datasets provided online on Kaggle [American Epilepsy Society, 2014].

2.3.2 EPILEPSIAE and EPILAB

The EU FP7 Project, EPILEPSIAE (Evolving Platform for Improving the Living Expectations of Patients Suffering from IctAl Events), is aimed at improving the quality of life for individuals who experience epileptic events, by employing brain-computer interaction technology. The main objective of the project is to use Information and Communication Technologies (ICT) to give epileptic patients the power to monitor their own risks and increase their safety in their daily activities. This way, they can further enhance their social integration. It is noteworthy that the project includes developing software, hardware, and wireless devices as essential parts of this initiative [epi].

The European Database on Epilepsy was produced by that project [Ihle et al., 2012a]. It is the largest online database currently available for predicting seizures. Its primary aim is to develop algorithms that can predict seizures by monitoring EEG data for signs of an upcoming seizure. This comprehensive dataset contains high-quality long-term electroencephalogram recordings collected from both scalp and intracranial electrodes (iEEG) along with complementary clinical and imaging data. The data repository involves 275 subjects, from which 225 EEG recordings were carried out using scalp electrodes while the rest were intracranial. It also encompasses more than 10 times the number of seizures as compared to previous compilations. The database includes standardized, annotated EEG and Electrocardiogram (ECG) recordings of patients along with comprehensive metadata. The metadata provides essential technical and clinical details regarding the recordings as well as pertinent patient-specific clinical information [Klatt et al., 2012].

EPILAB is a software package developed using Matlab® to help community study the prediction of epileptic seizures, also a product from EPILEPSIAE [Teixeira et al., 2011]. The tool provides a graphical user interface for analyzing data, allows for the integration of new prediction methodologies through its object-

oriented design and incorporates key concepts essential for predicting seizures. It is designed to process long-term recordings of EEG and ECG signals, and it can analyze both EEG and ECG data stored in various formats. EPILAB can extract over 35 features from EEG and ECG data using univariate and multivariate data analysis. These features can then be used to make predictions, which can be done using optimized thresholds or machine learning algorithms like artificial neural networks, cellular neuronal networks, and support vector machines. The architecture, features and algorithms of EPILAB are covered in more detail in the section 3.2.

Projects such as EPILEPSIAE and EPILAB have made significant progress in the field and serve as a basis for the further development of EPIWEB. EPILEPSIAE's focus on integrating data and discovering knowledge is in line with EPIWEB's aim to create a reliable platform that processes EEG signals and accurately predicts seizures. EPILAB's advanced algorithmic expertise is guiding the development of EPIWEB's predictive models, enhancing the web application's investigation capabilities.

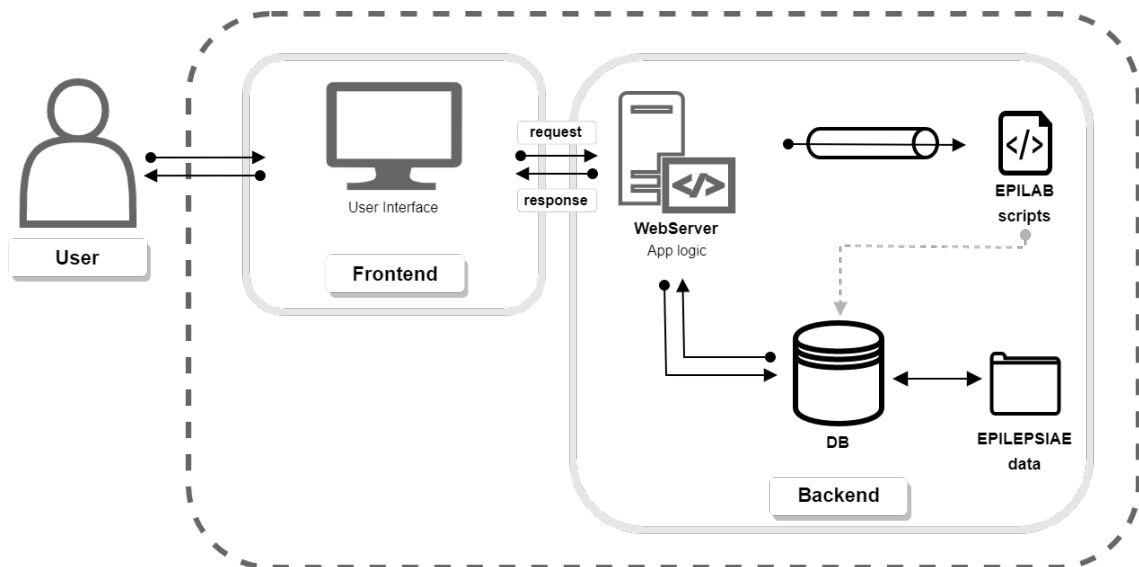


Figure 2.3: Illustrates the conceptual framework of the planned high-level architecture for EPIWEB and highlights the crucial role that EPILEPSIAE's EPILAB and database will play in its development process.

Chapter 3

State of Art

This chapter provides an extensive investigation of the current methodologies, tools, and platforms in the field of EEG-based seizure prediction. Section 3.1 explains the current framework and essential components for predicting seizures. Section 3.2 provides a comprehensive description of EPILAB’s algorithms, features, capabilities and some addressing aspects. Section 3.3 demonstrates the first prototype of EPIWEB. Lastly, section 3.4 analyses and compares available web solutions for epileptic seizure prediction.

3.1 Seizure Prediction Pipeline

The common methodologies used to forecast seizures stick to a set of steps that revolve around signal processing and machine learning. The following figure 3.1 provides a clear and concise depiction of these steps.

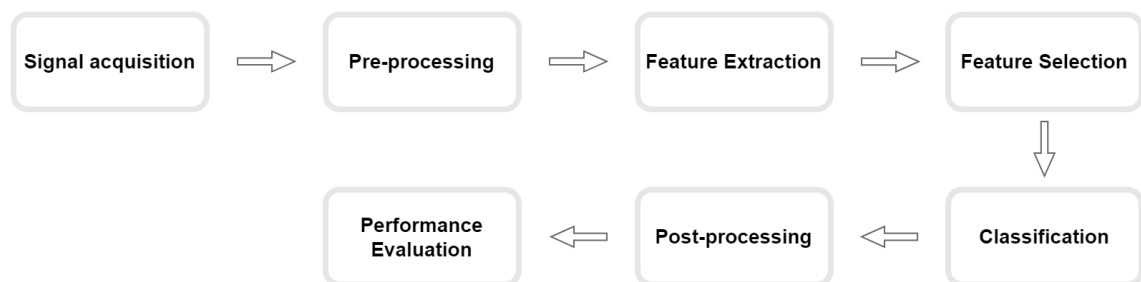


Figure 3.1: Flowchart illustrating the standard pipeline for seizure prediction. Adapted from [Bou Assi et al., 2017]

3.1.1 Pre-processing

Pre-processing is essential in improving the quality of Electroencephalogram (EEG) data by reducing noise and artifacts. Various filter techniques, such as band-pass, notch, and adaptive filters, are utilized to filter out interference from external sources, thereby enhancing the signal-to-noise ratio. Optimal pre-processing ensures that subsequent stages of feature extraction and classification rely on reliable and uncontaminated EEG data [Bou Assi et al., 2017].

3.1.2 Feature Extraction

In order to extract features from pre-processed EEG recordings, it is necessary to divide the recordings into smaller windows [Bou Assi et al., 2017], which is called as data segmentation.

Feature extraction involves the derivation of informative attributes from EEG signals. Certain features rely on single-channel measures, univariate features, whereas others incorporate insights derived from a combination of multiple channels, called multivariate features [Mormann et al., 2007]. Some techniques include time-domain features (e.g. statistical features like mean and variance), frequency-domain features (e.g. power spectral density), and time-frequency domain features (e.g. wavelet coefficients). For instance, wavelet transform captures frequency content variations over time and aids in identifying transient patterns associated with seizures (specific and temporary changes in EEG signals that occur before the onset of a seizure).

3.1.3 Feature Selection

Feature selection aims to identify a subset of relevant features that contribute most to predictive performance. After extracting numerous features from EEG data, a high-dimensional space is formed. To streamline computation, enhance classification, and prevent overfitting, diverse methods for feature selection or reduction are utilized [Bou Assi et al., 2017]. These methods prioritize features with strong discriminatory power and eliminate redundancy. Dimensionality reduction can be achieved through Principal Components Analysis (PCA). PCA transforms high-dimensional data into a lower-dimensional, orthogonal feature space, where new features reflect the principal components ranked by variance. Another technique for reducing data dimension could be Multidimensional Scaling (MDS).

3.1.4 Classification

Classification algorithms have a crucial role in distinguishing between seizure and non-seizure states. The most notable methods are the ones based on machine learning classifiers (Bouassi, et al., 2017). Support Vector Machine (SVM) maximise the margin between classes by identifying the optimal hyperplane. Artificial Neural Network (ANN) learn hierarchical features from EEG data, capturing intricate patterns automatically. The performance of each algorithm depends on the selected features, the size of the dataset and the desired model complexity.

3.1.5 Post-processing and Evaluation

Post-processing improves classification outputs to increase predictive accuracy. Regularization, also known as post-processing [Bou Assi et al., 2017], reduces abrupt fluctuations and false positives, controlling the generation of alarms. This enhances the specificity of the model. Approaches that consider temporal signal dynamics, such as Kalman filtering [Chisci et al., 2010] or the firing power technique [Teixeira et al., 2011], have been used. Post-processing methodologies ensure that predictive models generate coherent and clinically meaningful results.

Similar to other classification tasks, the performance of algorithms must be evaluated using performance metrics, like accuracy (ACC), specificity (SP), sensitivity (SE), false positive rate (FPR) and F2 score.

3.2 Underlying Architecture of EPILAB

As already mentioned, EPILAB is a Matlab toolbox developed by [Teixeira et al., 2011], as part of the EPILEPSIAE project. MATLAB is famous for its advanced computing and analysis features, which enable scientists, engineers, and researchers to overcome numerous challenges in various fields [TheMathWorksInc., 2022]. Nevertheless, its command-line interface could be daunting for users who aren't experienced in programming or who prefer a more graphical interface [TheMathWorksInc., 2023]. Graphical user interface applications bridge the gap by enabling users to take MATLAB's capabilities without requiring extensive programming expertise.

3.2.1 Overview

As shown in the following figure 3.2, EPILAB comprises five main modules, each with its own goals and functionalities. These modules are labeled as Study Creation, Feature Extraction, Dimensionality Reduction & Feature Selection, Prediction Algorithms and Visualization & Navigation. The last is not an independent module but a common functionality of all the other units.

Study Creation

A new study can be created by the user from raw data, i.e., signals that have been acquired from a specific patient. The data generally has a multi-file organization, and therefore, EPILAB has the capability to systematically analyze directories of files. This generates an internal map that allows the processing of all data as if it were contained within a single file. Previously computed datasets (e.g. set of extracted features) can also be used as an initial point for a new study.

Feature Extraction

Features can be calculated from Electrocardiogram (ECG) and EEG signals, depending from user preferences. These measures can be univariate or multivariate and they are calculated on a window-by-window basis (data segmentation as already explained). Figure 3.3 provides a comprehensive list of the features that EPILAB can extract using a large number of its own algorithms. Users can choose to use filters before computing features. Specifically, three Infinite Impulse Response (IIR) Butterworth filters are available: low-pass, high-pass, and notch.

Dimensionality Reduction & Feature Selection

EPILAB has incorporated two distinct approaches for feature selection: supervised, reliant on target classification, while the other was non-supervised. The non-supervised method facilitates feature ranking by evaluating the ratio between global and local variances and the supervised one, minimum redundancy–maximum relevance (mRMR) ranks features by simultaneously minimizing redundancy among them and maximizing their relevance to a specified target classification. EPILAB incorporates also algorithms for reducing dimensionality, primarily centered around PCA and MDS.

Prediction Algorithms

EPILAB uses traditional classification algorithms as part of its tools to predict epileptic seizures: ANNs, SVMs, and cellular neural networks (CNNs). For creating a seizure prediction model through classification methods, the initial step involves deciding the classifier inputs and the temporal division of data into training and testing sets (holdout cross-validation). EPILAB considers either two or four cerebral states, leading to two or four-class classification, each class representing an episode in an EEG signal (see figure 2.2).

Visualization & Navigation

In every preceding module, presenting information or graphical representations of the datasets, both processed (extracted features or classification results) and raw, is advantageous. EPILAB seeks a user-friendly solution to facilitate such visualizations, whether in text or graphical format;

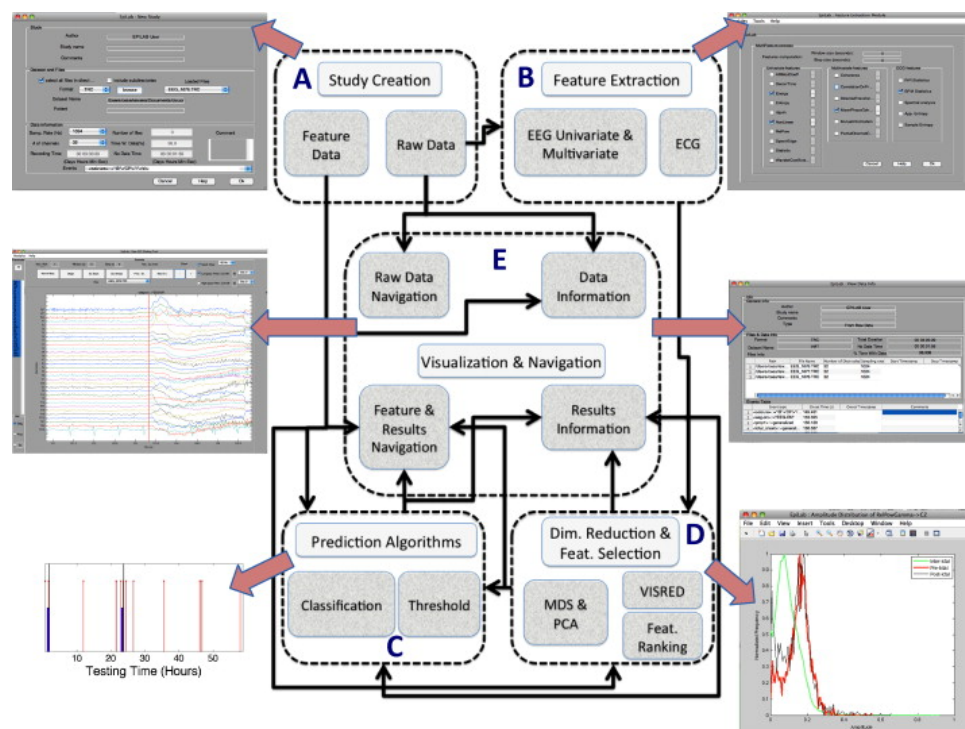


Figure 3.2: EPILAB flowchart. (A) The initiation of a new study involves creating it from raw data or precomputed feature data; (B) When commencing a study from raw data, EEG and/or ECG features need to be calculated; (C) Using either imported or computed features, prediction algorithms can be formulated and assessed; (D) Imported or computed features can undergo dimensionality reduction; (E) Throughout the study, data and outcomes can be visually or textually presented. From [Teixeira et al., 2011].

Univariate	Time domain	statistical moment	First statistical moment of EEG amplitudes (mean)
			Second statistical moment of EEG amplitudes (variance)
			Third statistical moment of EEG amplitudes (skewness)
			Fourth statistical moment of EEG amplitudes (kurtosis)
		Energy	
	Freq. domain	Linear modeling	Autoregressive coefficients using different methods and orders
			Mean-Squared error of the estimated AR models
		Relative power	spectral band delta (0.1-4Hz)
			spectral band theta (4-8Hz)
			spectral band alpha (8-15Hz)
			spectral band beta (15-30Hz)
			spectral band gamma (>30 Hz)
		Spectral edge	Spectral edge frequency
			Spectral edge power
		Hjorth parameters	Hjorth mobility
			Hjorth complexity
		Decorrelation time	
		Largest Lyapunov exponent	
		Correlation dimension	
		Entropy	
Cross Frequency Coupling			
Nonlinear			
Time and Freq.	Energy of the wavelet coefficients (different mother wavelets and decomposition levels)		
Bi/Multivariate	Coherence		
	Correlation on the prob. of recurrence		
	Directed transfer function		
	Mean phase coherence		
	Mutual information		
	Partial directed coherence		
ECG features	Time domain	RR-statistics	Mean
			Variance
			Minimum
			Maximum
		BPM-statistics	Mean
			Variance
			Minimum
			Maximum
	Freq. domain	Very low freq. (<0.04 Hz)	
		Low freq. (0.04–0.15 Hz)	
High freq. (0.15–0.4 Hz)			
	Approximate entropy (describing complexity and irregularity of the RR intervals)		

Figure 3.3: List of features that can be computed by EPILAB. From [Teixeira et al., 2011].

3.2.2 Limitations

The EPILAB toolbox provides useful tools for predicting seizures, but has some limitations mainly because it is a MATLAB toolbox. These limitations include:

- Software Ownership: To use MATLAB, users must, first, obtain a paid li-

cense, which restricts its usage and access to those with enough financial resources. Therefore, only individuals and institutions with the necessary financial resources can use and have access to the EPILAB toolbox.

- **Difficulty Learning to Use:** Besides its user-friendly GUI, users new to this language that may want to implement their own kind of algorithms may find difficulties on MATLAB syntax and programming. This could make it difficult for some people who are more comfortable with other programming languages to adopt MATLAB.
- **Scalability Limitations:** EPILAB's MATLAB implementation is likely to demand significant resources, particularly when dealing with computationally intensive tasks that involve large datasets. This may limit its scalability on systems having restricted computational resources.
- **It is dependent on the platform:** MATLAB relies on specific platforms, which means it might not work well on different operating systems. This may cause problems with compatibility when collaborating or sharing projects.
- **Integrating MATLAB-based functionality into web applications:** Incorporating MATLAB-based features into web applications can be complicated and often requires extra workarounds or tools. This may make it difficult to deploy EPILAB's features smoothly in a web environment.
- **Accessibility Limitations:** Users might face accessibility issues with EPILAB due to the requirement to install MATLAB on their machines, especially when they do not have MATLAB licenses.
- **Interoperability Limitations:** Integrating EPILAB with other data analysis or web development tools outside MATLAB's ecosystem could be challenging, thereby limiting its interoperability.

3.3 Initial Prototype of EPIWEB

An initial work, developed by colleague Carolina Pereira, was presented as a first prototype of EPIWEB and so a start point for the work. It had the same objective of the present work, a foundation for a comprehensive and user-friendly environment for EEG-based seizure prediction, based on EPILAB. This section discusses the layout and essential parts of the initial model.

3.3.1 Architecture Overview

The first version of Carolina’s EPIWEB platform uses a modular design that includes Java Maven, Servlets, and Python scripts to create a dynamic web application. Maven makes it easy to handle project dependencies and streamline deployment. [The Apache Software Foundation]. The prototype uses a Model-View-Controller (MVC) design [Baeldung]. A diagram of this design and its flow are shown on figure 3.4.

The Model holds the application’s fundamental logic, such as data processing and classification algorithms. JavaBean classes compose the Models.

The View is created with Jakarta Server Pages (JSP), allowing the generation of dynamic web pages with Java code inside. JavaServer Pages Standard Tag Library (JSTL) improves the View layer with its tag-based approach, simplifying the integration of dynamic content and logic into web pages.

The Controller layer consists of Servlets that invoke the Model and send data to the View. These Servlets handle user requests and control data flow. Users start Servlet execution by interacting with the web interface, which manages Hypertext Transfer Protocol (HTTP) requests and responses, enabling real-time communication. Servlet execution handles the EEG data processing and runs Python scripts for EPILAB algorithms. This feature enables users to upload their EEG data, set prediction parameters, and view results.

For data storage, the prototype incorporates a PostgreSQL database. The PostgreSQL database serves as the backend repository for storing EEG data, feature extraction results, and prediction outcomes.

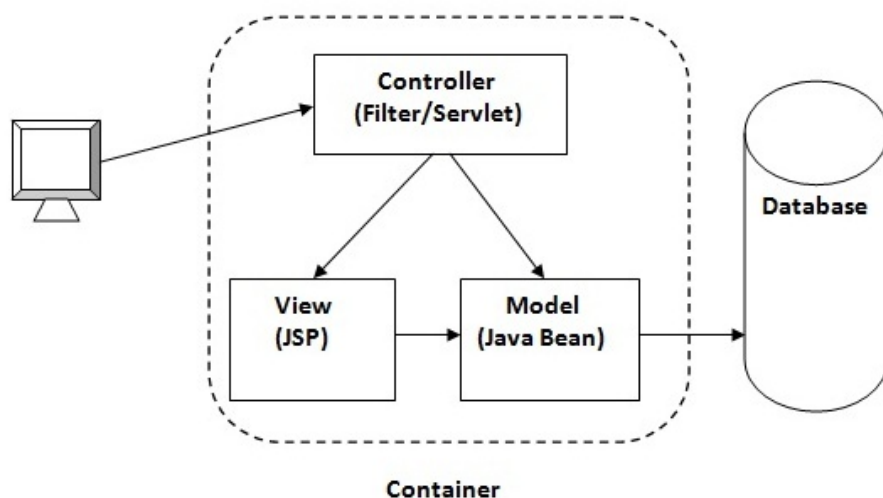


Figure 3.4: Model-View-Controller design. From [javatpoint].

3.3.2 Functionalities

To carry out the scientific component of this work, which involves preparing the data, selecting relevant features, reducing dimensions, and classifying data, Carolina chose to replicate EPILAB's MATLAB scripts using Python programming language.

Therefore, Python scripts are vital components of the prototype, that implement EPILAB algorithms for processing EEG data and predicting seizures. To seamlessly invoke Python scripts, Java's Process Builder has been utilised [Oracle Corporation]. It launches Python scripts by creating a native operating system process. This integration enables the prototype to employ EPILAB's algorithms within a web-based Java application.

In Python, a subset of the feature extraction algorithms from EPILAB's module were replicated. This subset includes algorithms like statistical moments and wavelet coefficients computation in the time domain. Furthermore, the visualization module and graphical signal analysis were also constructed. To accomplish this, Carolina utilized the Plotly.js library.

This prototype uses a template as the basis for its User Interface (UI) design, aiming for a user-friendly interface that is both visually pleasing and well-organised.

3.3.3 Early Exploration and Limitations

The initial version of EPIWEB developed by Carolina is a significant exploration of the possibility of a web-based platform for predicting seizures. Although the prototype demonstrates important functions, the prototype is simply a starting point towards the all-encompassing EPIWEB platform. The design and characteristics of the prototype are open to further improvement and enhancement.

3.4 Existing Web Platforms with Similar Features

Although the field of epilepsy seizure prediction using EEG is still under research and development, it is important to mention that currently, no existing platforms similar to EPIWEB have been found. EPIWEB's integration of EPILAB algorithms in a user-friendly web application that is designed for preictal and ictal period prediction is innovative, given the lack of similar existing platforms.

The lack of similar platforms highlights the distinctive contributions that EPIWEB intends to offer for epilepsy research and seizure prediction. EPIWEB fills a significant void in the area by merging EPILAB's reliable algorithms with a web-based platform, delivering a comprehensive tool to predict seizures and scrutinize EEG data in an effortless way, to the community.

Chapter 4

Methodology

Firstly, a comprehensive overview of the development organization will be provided, specifically focusing on the development methodology employed, as outlined in section 4.1. Next, section 4.2 introduces the project's plan and compares the anticipated timelines with those executed. Section 4.3 will feature a reflective exploration of the risks that were identified prior to embarking on the development phase.

4.1 Project Management Methodology

The achievement of the EPIWEB project relies on a methodical project management approach. The development process follows methodology to promote effective communication, efficient progress, and adapting to changing requirements. The methodology takes inspiration from agile principles that aim to allow a dynamic environment of software development and collaboration between the solo developer and project advisors [Beck et al., 2001]. There are key aspects to explain about the methodology. Among them, refer to the following:

- **Iterative Development and Weekly Meetings:** The project management methodology is based on iterative development cycles, facilitating incremental progress, continuous feedback and flexibility to make necessary adjustments. Weekly meetings with project advisors are a vital element of this methodology, serving as checkpoints to review progress, discuss challenges and align development efforts with project objectives. The feedback gathered during these meetings provides guidance for subsequent development steps.
- **Flexibility and Adaptability:** The project management methodology places

great emphasis on adaptability. Acknowledging the potential for modifications to the project's requirements and priorities, the methodology allows for effortless adjustments to the development plan. The iterative nature of this approach facilitates the inclusion of changes, improvements, and enhancements as the project progresses.

- **Communication:** Open communication between the solo developer and project advisors are essential to the approach. Regular interactions, both during weekly meetings and through ongoing discussions, guarantee that the project stays focused on its goals. The approach promotes collaborative decision-making, knowledge exchange, and brainstorming of ideas to overcome challenges.
- **Risk Management and Problem-Solving:** The project management methodology emphasizes proactive risk management and creative problem-solving. By identifying potential challenges early in the development process, strategies can be formulated to address them. When unexpected issues arise, the collaborative environment established by the methodology allows for rapid problem-solving and the exploration of innovative solutions.

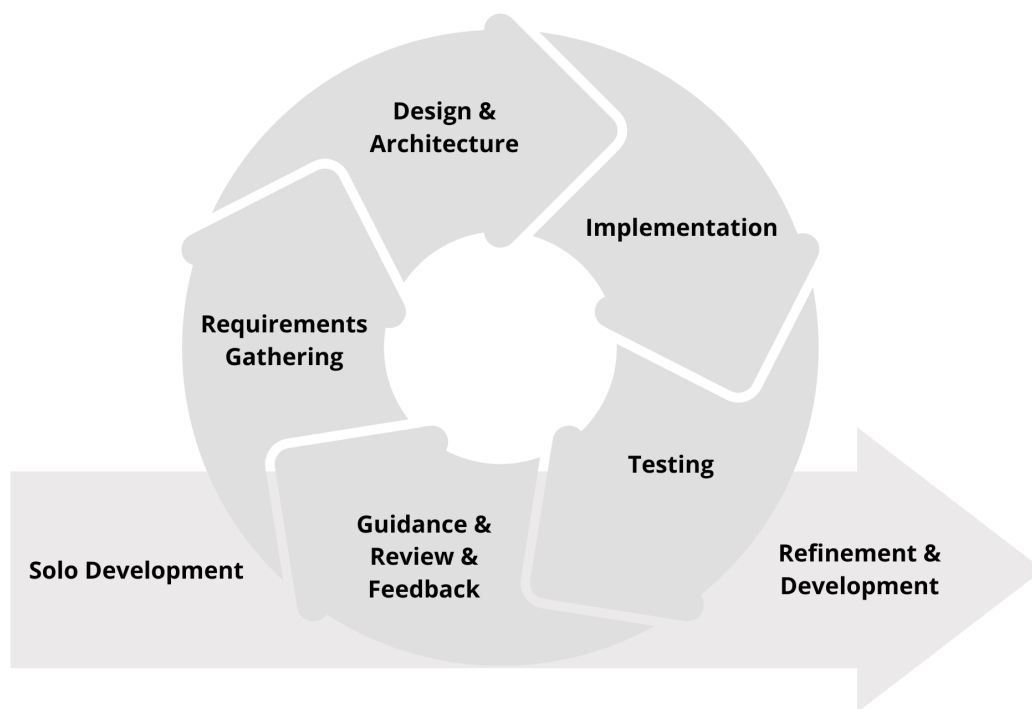


Figure 4.1: Development Methodology: Iterative Approach with Weekly Feedback Loops (Agile based).

4.2 Planning

At the beginning of the project, a detailed work plan was created to specify the order of tasks and timelines. This plan acted as a guide for project development and aimed to guarantee efficient task execution. A simple Gantt chart, as shown in Figure 4.2, provides a visual representation of the project schedule, highlighting tasks and delays timeframes. This chart effectively illustrates the evolving timeline of the project, capturing both planned milestones and any deviations from the schedule.

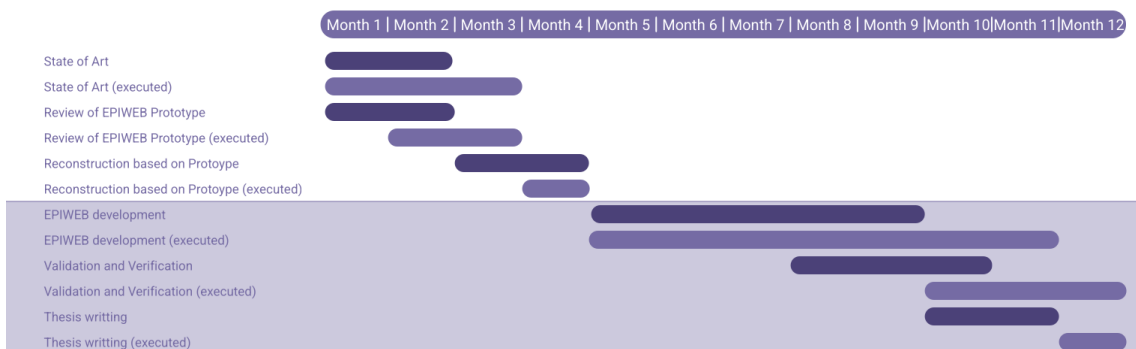


Figure 4.2: Gantt diagram.

4.3 Risk Analysis

Throughout the development of the EPIWEB project, a comprehensive risk analysis was conducted to identify potential challenges and uncertainties that could impact the project's progress, timeline, and outcomes. Each identified risk was assigned with the following structure:

- **ID**
- **Description**
- **Mitigation Strategy**
- **Probability:**
 - Low
 - Medium
 - High
- **Impact:**

- Low: Does not contribute positively to the project's progress.
- Medium: Interferes with the development of the project and can cause changes to the final product.
- High: Puts the project's completion at risk.

Below, it is detailed potential risks associated with the project and their corresponding mitigation plans, aimed at minimising.

R1 - Insufficient Familiarity with Tools and Technologies

- **ID:** 1
- **Description:** This risk revolves around the possibility that the developer may not have sufficient expertise or familiarity with the chosen frameworks, like Django. Inexperience with the framework could lead to inefficiencies, errors, or delays in development.
- **Mitigation Strategy:** To reduce this risk, the developer could proactively enrol in online tutorials and conduct an extensive study of the documentation. These actions are intended to enhance knowledge and skills with the selected framework.
- **Probability:** Medium.
- **Impact:** Medium.

R2 - Changes in project requirements

- **ID:** 2
- **Description:** This risk considers the possibility of evolving project requirements due to changing needs or priorities. These modifications have the potential to lead to changes in project goals or alterations to project deliverables.
- **Mitigation Strategy:** To mitigate this risk, the developer must hold frequent meetings with project advisors. Regular meetings ensured that the project keep to the original requirements and objectives. Any suggested alterations may be reviewed and deliberated before being implemented. In addition to the hours outlined in the project plan, the developer will be required to work additional hours to finalize the implementation and validation of all features.

- **Probability:** High.
- **Impact:** Medium.

R3 - Complexity of Electroencephalogram (EEG) data processing algorithms

- **ID:** 3
- **Description:** This risk presents difficulties that may arise when integrating external libraries, such as PyWavelets and scikit-learn, into the project to replicate EPILAB algorithms. Integration challenges could result in delays or unexpected issues.
- **Mitigation Strategy:** The developer must invest time in extensive research, know in depth EPILAB scripts, and thoroughly reviewed documentation related to the external libraries.
- **Probability:** Medium.
- **Impact:** Medium.

R4 - Hardware limitations affecting performance

- **ID:** 4
- **Description:** This risk concerns potential hardware limitations that could affect the performance of the web application. Inadequate hardware resources may lead to slow processing or decreased system responsiveness.
- **Mitigation Strategy:** To address this risk, the developer must focus on optimizing code to minimize resource-intensive operations. Additionally, Centre for Informatics and Systems of the University of Coimbra (CISUC) Lab provides hardware to overcome this possibility.
- **Probability:** Low.
- **Impact:** Low.

R5 - Size of work too ambitious

- **ID:** 5

- **Description:** There is a possibility that the project’s goals may be too ambitious, causing difficulties in efficiently managing and completing all planned tasks within the given resources and deadline. An exceedingly ambitious project scope can burden available resources, affect the quality of work, and delay the project’s completion.
- **Mitigation Strategy:** Prioritisation of identified requirements is necessary to understand which ones should be implemented. If it is not feasible to complete all functionalities within the available time, requirements priorities need to be redefined.
- **Probability:** High.
- **Impact:** High.

To aid comprehension and see the risks associated with the project development more clearly, Table 4.1 gives a summary of the probability and impact of each risk.

Impact \ Probability	Low	Medium	High
	Low	R4	
Medium		R1 R3	R2
High			R5

Table 4.1: Risks classification.

Chapter 5

Requirements Analysis

This chapter sets the first step for EPIWEB implementation by examining its requirements and functionalities. By analyzing its actors, functional and non-functional needs, and User Interface (UI) visuals, it is created a comprehensive plan for constructing a strong and user-oriented application.

The Requirements chapter serves as a link between user expectations and technical implementation. These requirements were defined and identified based on meetings with the project advisors, to address the product's needs, not forgetting the previous EPILAB solution. Through prioritized requirements and functional User Stories, the objective is to ensure a valuable user experience. In addition to core functionalities, this chapter explores broader aspects such as non-functional attributes like usability, performance, and security. Sketches also provide a visual preview of user interactions. This exploration establishes the framework for architecture design and implementation, as well as guidelines that must be adhered to during EPIWEB implementation. This aligns with our mission to create an advanced tool for the epilepsy field.

This chapter is organised into three sections. Section 5.1 identifies the actors involved in the application. Section 5.2 presents the User Stories developed to specify the functional requirements Sections 5.3 and 5.4 provide explanations of the functional and non-functional requirements. Finally, section 5.4 outlines a low-level prototyping process using sketches.

5.1 Actors

In this phase, EPIWEB aims to serve a limited number of user groups, with straightforward needs and roles within the platform. Knowing these actors is im-

portant in planning EPIWEB's functionalities to suit their specific requirements.

Authenticated Users

Authenticated users are a primary group that may include enthusiasts, researchers and analysts who use EPIWEB to access advanced tools and functionalities. This group has the power to create accounts, sign in and contribute to the platform's data processing and analysis activities. Their roles encompass:

- Creating and managing classification models.
- Uploading and processing EEG data.
- Accessing detailed predictions and analyses.
- Exploring visual representations of results.
- Leveraging advanced features for research and diagnosis.

Non-authenticated Users

Non-authenticated users constitute individuals who interact with EPIWEB's public-facing features before creating an account. This user group includes:

- Individuals seeking information about epileptic seizures and prediction.
- Those exploring the platform's functionalities before deciding to register.
- Participants interested in understanding the scope and benefits of EPIWEB.

5.2 User Stories and Functional Requirements

In requirements gathering, User Stories can be a key strategy. This methodology prioritises the end user by presenting a concise description of product features from their perspective.

A widely used approach for gathering requirements is the formulation of User Stories. This approach gives highest importance to considering the viewpoint of the consumer. A User Story (US) comprises a concise, informal narrative of features, crafted to show the perspective of the user and how they engage with the product [Rehkopf, 2022]. These stories enable the extraction of the value added

to the user experience when executing specific tasks. The structure of a typical US follows this pattern:

"As [type of user], I want [intention], for [purpose]."

This structured approach to requirement gathering enhances communication, prioritization, and clarity in the project development.

During the start of the project, meetings with the advisors were held to discuss the platform's desired features. One of the main objectives is to reproduce the EPILAB toolbox's functionalities in the web application realm. Therefore, several User Stories have taken inspiration from the way EPILAB presents its interface and script results. These align with the initial prerequisites discussed with thesis advisors.

The functional requirements were created from the User Stories developed and presented.

The full list of User Stories that have been developed can be found in Appendix A. Next, some examples of User Stories are presented to explain their importance and structure, together with the corresponding functional requirements.

5.2.1 User Stories Examples

Epic Story 1: Authentication

As a user, i want to sign in to EPIWEB, to have access to its functionalities.

US-01: Landing Page

As an non-authenticated user, I want to see a welcoming and informative home page to provide me an overview of the platform's capabilities and features and the possibility to sign in.

US-02: Login

As an non-authenticated user of EPIWEB, I want to be able to log in to the platform securely so that I can access and use the platform's features. I want to enter the fields like username and password so that my identity is validated when accessing the platform.

US-03: Register

As an non-authenticated user, **I want** to recover my password by providing my registered email address. Upon submitting the email, I should receive a password reset email containing a secure link. Clicking the link will allow me **to** set a new password, ensuring that I can access my account. For that i need to introduce personal information like my email.

US-04: Create an Account

As an non-authenticated user, **I want** to create an EPIWEB account by providing my personal information and selecting a username and password. This will enable me **to** access the platform's features and contribute to epilepsy research. For that i need to introduce personal information like email, username and password.

US-05: Log out

As an authenticated user, **I want** the ability to log out of my EPIWEB account when I'm done using the platform **to** ensure the security of my account and data.

5.2.2 Functional Requirements

This section presents EPIWEB's functional requirements. These requirements are prioritised using the MoSCoW method, which stands for Must, Should, Could, Won't Have [Agile Business Consortium, 2022]. This prioritisation approach has been chosen to help determine which requirements are essential for implementation during the placement. Given the limited duration of this EPIWEB implementation phase, it may not be possible to address all the requirements identified. The MoSCoW methodology was chosen for its effectiveness in categorising and managing requirement priorities at four levels, as described below [Agile Business Consortium, 2022]:

- **Must Have (M):** considering the situation where a requirement isn't met. If implementing a solution without meeting this requirement would result in project cancellation, it qualifies as a Must Have requirement. However, if there is a workable solution, even if it's challenging and requires manual input, it falls under the Should Have or Could Have requirements.
- **Should Have (S):** requirements that are significant, but not essential, may cause inconvenience when omitted. However, the solution remains viable

by implementing workarounds, such as managing expectations, temporary inefficiencies or using existing solutions. Should Have requirements can be distinguished from Could Have requirements based on the level of inconvenience caused by their absence, considering factors like the number of affected individuals.

- **Could Have (C):** requirements desired but of lower priority. Their omission has a lesser impact compared to Should Have requirements. Could Have requirements often serve as contingency options and may only be fully implemented in an ideal situation. When confronted with difficulties or time constraints, they are typically postponed or eliminated.

- **Won't Have (W):** requirements that are deliberately excluded from the current project scope. They are documented to define the boundaries of the project and prevent their unintentional inclusion later. This ensures realistic expectations and focuses attention on the higher priority Could Have, Should Have and especially Must Have requirements.

The following table 5.1 lists the functional requirements that were derived from the User Stories. Each row (representing a requirement) contains various fields, each in a separate column. These fields include an identifier for the requirement, the associated actor, its priority level, and any dependencies on other requirements if relevant.

ID	Name	Priority	Dependencies
RF-01	Landing Page	Must	
RF-02	User Authentication	Must	
RF-03	Password Recovery	Won't	RF-02
RF-04	User Registration	Must	
RF-05	User Sign Out	Must	RF-02
RF-06	User Personal Home Page	Must	
RF-07	Studies Processing Queue Information	Must	
RF-08	Notifications for Study Updates	Must	
RF-09	Recent Studies Table	Could	
RF-10	Load Raw Data	Must	
RF-11	Access Raw Data	Must	RF-10
RF-12	Manage a Specific Raw Data	Must	RF-11
RF-13	Visualize Graphically Raw Data	Must	RF-11
RF-14	Filter Channels on Graphic Visualization	Must	RF-13
RF-15	Analyze Average Plot	Must	RF-13
RF-16	Analyze Subtraction Plots	Must	RF-13
RF-17	Change Window Size of Visualization	Must	RF-13
RF-18	Jump to Specific Time	Must	RF-13
RF-19	Jump to Specific Event	Must	RF-13
RF-20	Real-Time Filtering	Must	RF-13
RF-21	Share Raw Data	Must	RF-11
RF-22	Comment on Raw Data	Must	RF-11
RF-23	Access to Studies	Must	
RF-24	Manage a Specific Study	Must	RF-23
RF-25	Comment on a Study	Must	RF-23
RF-26	Share a Study	Must	RF-23
RF-27	Create a Feature Extraction Study	Must	
RF-28	Visualize Graphically in a Feature Extraction	Must	RF-23
RF-29	Filter Graphs by Features in a Feature Extraction	Could	RF-28
RF-30	Change Window Size in a Feature Extraction Study	Should	RF-28
RF-31	Jump to Specific Time in a Feature Extraction Study	Should	RF-28
RF-32	Jump to Specific Event in a Feature Extraction Study	Should	RF-28
RF-33	Edit a Feature Extraction Study	Should	RF-27
RF-34	Create a Feature Selection/Dimensionality Reduction Study	Must	
RF-35	Visualize Graphically in a Feature Selection/Dimensionality Reduction Study	Won't	RF-23
RF-36	Filter Graphs by Features in a Feature Selection/Dimensionality Reduction Study	Should	RF-35
RF-37	Change Window Size in a Feature Selection/Dimensionality Reduction Study	Won't	RF-35
RF-38	Jump to Specific Time in a Feature Selection/Dimensionality Reduction Study	Won't	RF-35
RF-39	Jump to Specific Event in a Feature Selection/Dimensionality Reduction Study	Won't	RF-35
RF-40	Edit Feature Selection/Dimensionality Reduction Study	Should	RF-34
RF-41	Create a Classification Study	Must	
RF-42	Visualize Classification Study Graphs	Won't	RF-23
RF-43	Change Window Size in a Classification Study	Won't	RF-42
RF-44	Jump to Specific Time in a Classification Study	Won't	RF-42

Table 5.1: List of functional requirements.

5.3 Non-functional Requirements

The non-functional requirements of EPIWEB comprise essential aspects that affect the user experience, performance, and accessibility of the platform. While the application is not yet in the production environment, focusing on these requirements establishes a foundation for a good transition to a fully operational state. Performance, usability, security are some examples of non functional requirements.

EPIWEB places a strong emphasis on **usability**, with a focus on providing an

intuitive and user-friendly interface. The design principles used intend to reduce the difficulty in learning for users and improve usability in general, serving both experienced and new users, in contrast to the previous tool, EPILAB. By adhering to established web design guidelines, EPIWEB ensures that users can navigate through its functionalities with ease.

Although EPIWEB is not yet deployed for public use, **performance** considerations have been an essential aspect of the platform's design. The system architecture is optimized to handle tasks, process and visualize data, and handle user interactions efficiently. These performance features contribute to a smooth user experience and facilitate effective platform scalability.

Even at this stage of development, ensuring **security** is extremely important. EPIWEB recognises the sensitivity of EEG data and is dedicated to maintaining rigorous security procedures. Access controls and user authentication mechanisms need to be implemented in order to protect user data and ensure that only authorised individuals have access to the platform. During the development environment, measures must be taken to maintain the privacy of EEG signals and related data.

5.4 User Interface

During the transition of EPILAB's functionality to a web platform, initial UI **sketches** were developed based on requirements extracted from User Stories. These sketches served as a starting point, using EPILAB's existing screens as a basis for further development. While these screens provided valuable insights, adaptations were made to suit the web environment and modern user expectations. Due to the solo development nature of the project and the lack of a formal design approval process, the designs were not subjected to traditional review and validation. However, they played an important role in discussions with advisors and helped to communicate the conceptual translation of EPILAB's functionalities into a web-based interface.

These sketches, although not as complete as formal mockups, provided a first look of the potential structure and functionality of the web platform. They acted as reference points during iterations with the advisors, facilitating constructive feedback and refinement discussions (development methodology will be detailed further in the chapter 8). While sketches lacked comprehensive detail, their purpose was to facilitate communication and collaboration by providing a tangible representation of the basic aspects of the UI. The aim was not to achieve pixel-perfect precision, but rather to establish a foundation for subsequent de-

sign iterations. The images of these initial sketches serve as valuable documentation resources [Nick Badich, 2020]. The developed sketches may not contain attributes that will be present in the final application, being subject to improvements. Therefore, the following subsections present figures that showcase the developed sketches.

5.4.1 Authentication

Upon accessing the application's landing page, users should encounter essential information such as platform description and features. This page also serves as the entry point for user authentication. Illustrated in figure 5.1 is a sketch for the user authentication interface. Once users attempt to log in, they are presented with this interface, requiring them to input their associated email and password. In cases of incorrect credentials, the user is promptly notified of the error.

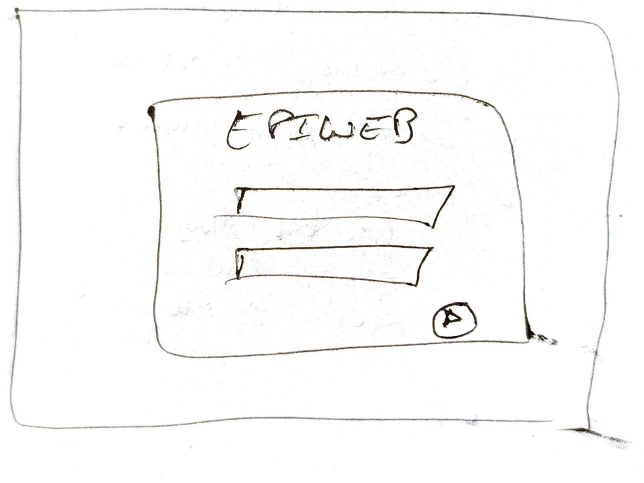


Figure 5.1: EPIWEB - Authentication page sketch.

5.4.2 Home Page

Upon logging in, users who have been authenticated are taken to their home page which replicates the functionalities of EPILAB (figure 5.2). This page provides different choices, such as the ability to access the list of raw data and completed studies. Additionally, users can create new studies or upload additional raw data by clicking the respective buttons located directly on this interface.

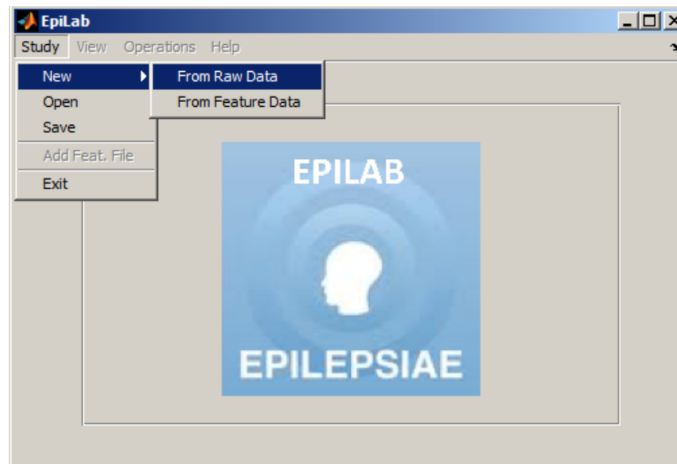


Figure 5.2: EPILAB - Main GUI.

5.4.3 Raw Data

On the patient listing page (raw data), users should be able to access key information from a selected patient, such as their ID, the number of related files, EEG channels, registration period, start and end dates, and sampling frequency, depicted in the sketch on figure 5.3. Nonetheless, users should also be able to investigate patient details further by accessing to a dedicated patient page, similar to the functionality observed in EPILAB, as demonstrated in the figure 5.4.

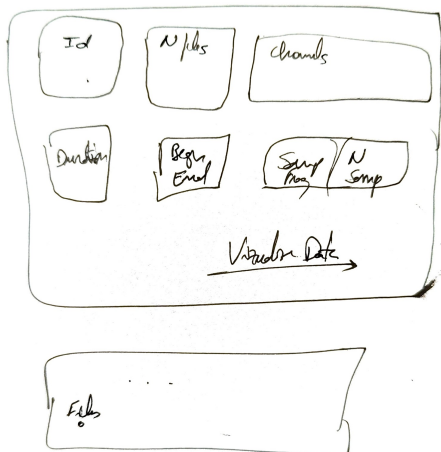


Figure 5.3: EPIWEB - Raw data fast details on list page sketch.

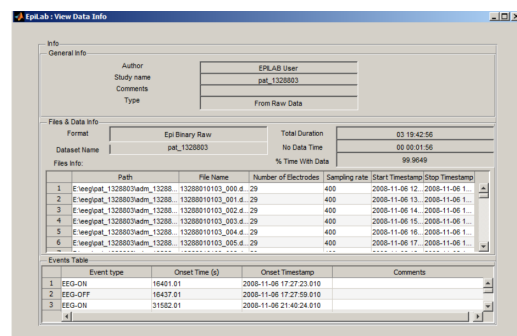


Figure 5.4: EPILAB - Information of the raw data GUI.

Within the same patient page, users should have access to the raw data, granting them with the ability to analyse specific channels of interest using real-time filtering capabilities. Within the same patient page, users should have access to the raw data, granting them with the ability to analyse specific channels of

interest using real-time filtering capabilities. The viewing page will mirror the functionality in EPILAB and enable the customisation of window dimensions.

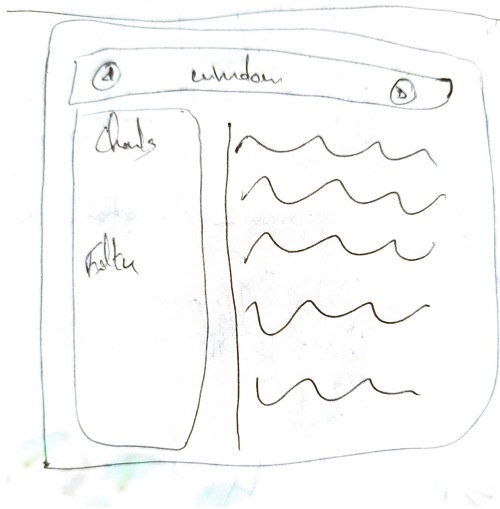


Figure 5.5: EPIWEB - Raw data visualization sketch.

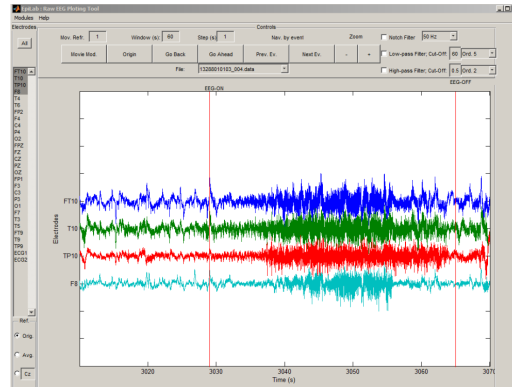


Figure 5.6: EPILAB - Raw data visualization.

5.4.4 Studies

Regarding studies, their presentation in the listing should mirror the structure of raw data, eliminating the need for a separate sketch.

For study creation, the provided sketches, represented by figures 5.7 and 5.9, draw inspiration from EPILAB's approach (figures 5.8 and 5.10), incorporating elements such as an informative dataset section relevant to the study and a graphical analysis preview.

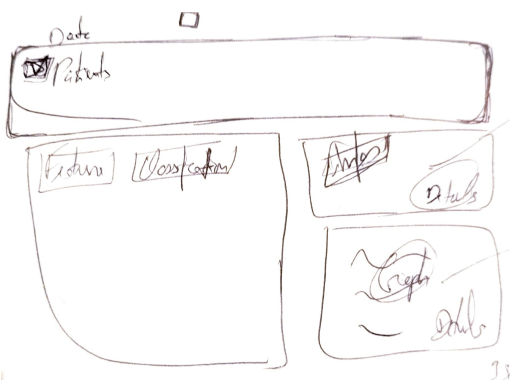


Figure 5.7: EPIWEB - Study creation page sketch.

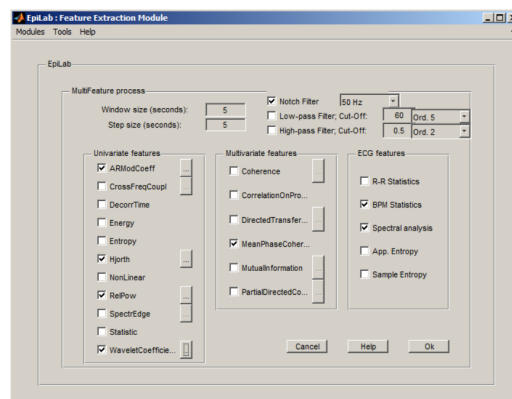


Figure 5.8: EPILAB - Feature extraction study creation page GUI.

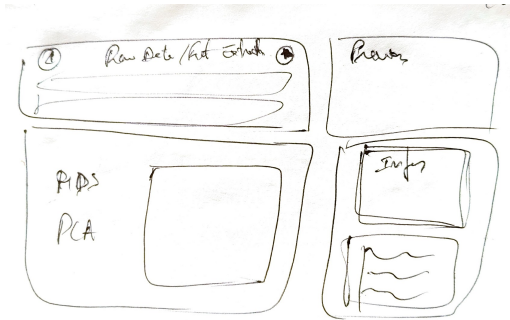


Figure 5.9: EPIWEB - Features selection study creation page sketch.

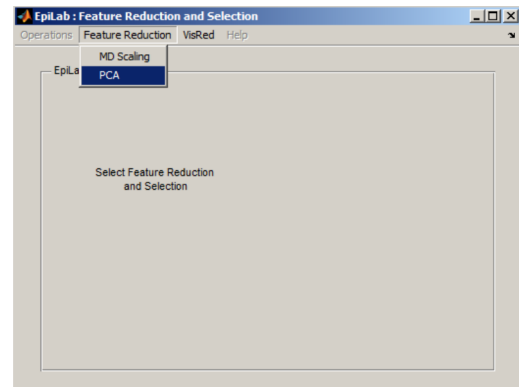


Figure 5.10: EPILAB - Feature selection study creation page GUI.

EPIWEB is tasked with the responsibility of providing dedicated information pages for each study, accessible through the listing page. In pursuit of this objective, the sketch from figure 5.11 was drawn. Drawing inspiration from the architecture of EPILAB, these study information pages should encompass various elements. These include segments dedicated to essential study information, such as creation date and the user who created it, alongside study-specific options like feature extraction settings. Additionally, the page design incorporates a comments section, allowing collaborative discussion among researchers, and an intuitive button for graphical visualization. Furthermore, users must also access information on other studies that have derived from this specific one.

It is also noteworthy that the visualization page mirrors the design of the raw data visualization page, thereby ensuring a consistent user experience.

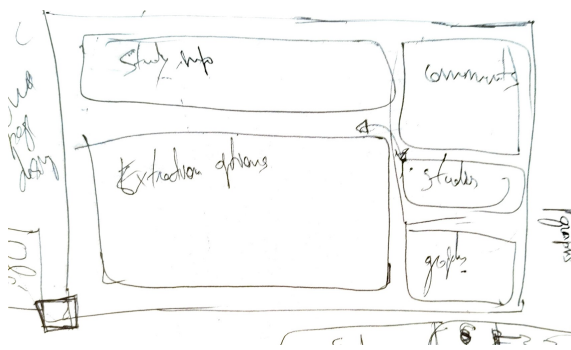


Figure 5.11: EPIWEB - Extraction study information page sketch.

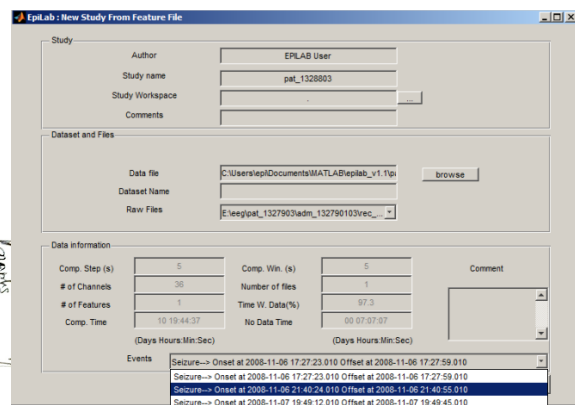


Figure 5.12: EPILAB - Feature extraction study information page GUI.

Chapter 6

Technologies

To successfully implement EPIWEB, it is crucial to choose appropriate backend and frontend technologies while considering both functional and non-functional requirements of the system. Although the system implementation technology can be chosen freely, it is crucial to make informed and justified selections. Therefore, a brief overview of the most common technologies in this field is valuable as it helps to make optimal decisions for the project's success.

In Section 6.1, it is presented some backend technologies, explaining briefly their advantages and disadvantages. Similarly, Section 6.2 thoroughly evaluates the frontend technologies, emphasizing their benefits and limitations. Taken together, these sections demonstrate the approach that was adopted to ensure that the chosen technologies are in complete alignment with the goals and requirements of EPIWEB.

6.1 Backend

The EPIWEB platform's backend is responsible for processing data, running algorithms and facilitating communication between the User Interface (UI) and the data infrastructure. By using a range of technologies, EPIWEB manages data effectively, seamlessly integrates systems and carries out EPILAB algorithms.

6.1.1 Integration of EPILAB Scripts

EPIWEB effortlessly translates EPILAB algorithms, previously only accessible through MATLAB. Unlike other project phases, this integration presents a fundamental data science problem. It comprises the complex task of combining data

from different datasets, analysing it, visually expressing its importance, and developing machine learning algorithms. The decision of the most appropriate programming language for this task is of utmost importance, given the project's specific objectives and the developer's level of expertise [Rice University].

Python

Python, known for its adaptability and extensive ecosystem, must assume a significant role in the EPIWEB backend. Emerging as the leader in programming language popularity rankings [TIOBE, 2023], Python is an open-source and versatile programming language, that extends its applicability not only to the data science field but also across other domains. The versatility of Python is truly remarkable, being widely used for data science tasks. This is predominantly attributed to its robust library ecosystem. Backed by a vast community of users, Python offers a wide range of reliable packages that allow various operations such as data pre-processing, visualization, statistical analysis, and intricate deployment of machine learning and deep learning models [Rice University].

Libraries like NumPy allow for efficient numerical computations, whereas SciPy offers resources for scientific and technical computing. Furthermore, scikit-learn improves the machine learning capabilities of EPIWEB with its provision of algorithms for regression and classification, among other things. Another crucial library within this field is TensorFlow [Javier Canales Luna, 2023].

6.1.2 Backend Frameworks

EPIWEB's back-end architecture is supported by robust frameworks that simplify development, facilitate structured organisation, and handle Hypertext Transfer Protocol (HTTP) requests with efficiency. Different frameworks catering to various preferences and needs are elucidated below.

Backend Framework	Repository Stars (GitHub archive)
Laravel	71,903
Django	67,941
Spring	64,405
Flask	59,137
Express JS	53,209
Nest JS	53,209
Ruby on Rails	52,913

Table 6.1: Backend frameworks usage on 2023 according to Statistics and Data. From [Statistics & Data, 2023].

Flask

Flask is a micro web framework for Python which provides a minimalistic foundation for creating web applications. It provides fine-grained control over components enabling developers to construct applications customized to suit their specific requirements. Flask's core offers essentials such as routing and request handling, while its modular approach facilitates smooth integration of third-party libraries for database connectivity, authentication, and other functionalities. The versatility of Flask renders it a prime option for projects that require bespoke solutions and value resource efficiency.

Django

Django is a Python web framework with a high-level design and a comprehensive suite of built-in functionalities. Its inclusive philosophy encompasses user authentication, an Object Relational Mapping (ORM) for managing databases, and a resilient administrative interface. Django's Model-View-Template (MVT) model is a helpful programming paradigm for creating applications, as it organises the various elements and their interactions. The framework enables swift development while maintaining best practices, rendering it appropriate for projects necessitating extensive features from the onset.

Express.js

Express.js, constructed on top of Node.js, is favoured by JavaScript aficionados. Its versatile and minimalistic design enables programmers to create web applications quickly. Express.js endorses middleware to deal with HTTP requests, which has made it a widely-used option for constructing APIs and single-page applica-

tions.

Spring

Spring, a Java-based framework, is well-suited for the enterprise environment. Its modular design and wide-ranging features support the creation of durable, scalable, and easily maintainable applications. The Spring ecosystem incorporates tools for multiple areas, such as data access, security, and integration.

Laravel

Laravel, a PHP web framework, prioritises developer convenience and elegant syntax. It offers features such as routing, authentication, and ORM, streamlining development. Laravel's expressive syntax and built-in tools enhance developer productivity.

6.1.3 Data Persistence

EPIWEB must use the PostgreSQL relational database management system to ensure strong data management and accuracy. PostgreSQL's support for complex data types, ACID-compliant transactions, and scalability make it an excellent option for storing Electroencephalogram (EEG) data, analytical results, and user-related data.

6.2 Frontend

The UI is how users access the EPIWEB platform, and it impacts their experience and interaction. This section explores the technologies that could be used in the EPIWEB frontend. It covers fundamental concepts, commonly used frameworks, and visualization libraries that users can engage with.

The frontend of EPIWEB encapsulates the visual and interactive components that users engage with. It facilitates the representation and manipulation of data, empowering users to interact seamlessly with the platform's functionalities.

6.2.1 Fundamental Concepts

At the core of modern web development, HyperText Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript collaborate to shape the structure, design, and interactivity of web pages [Lindley].

HTML

HTML forms the foundation of web development by structuring content. It includes a set of elements, each represented by opening and closing tags enclosed in angle brackets. HTML defines the structure and content of a web page, including headings, paragraphs, links, images, lists, forms, and more. It also allows for the inclusion of multimedia, hyperlinks, and metadata. Web browsers interpret HTML documents to render web content for users to see [MDN Web Docs, 2021b; W3C, 2021b].

CSS

CSS is essential for shaping a website's visual appeal and layout, complementing HTML by offering rules to control how HTML elements appear. It serves several roles, empowering designers and developers to define colors, fonts, spacing, and backgrounds for customization, reflecting brand identity and enhancing readability. CSS also arranges elements on web pages, ensuring responsive designs that adapt to diverse devices through techniques like Flexbox and CSS Grid, and maintains consistent styling across a website by propagating changes efficiently [MDN Web Docs, 2021a; W3C, 2021a].

JavaScript

JavaScript is a useful script language that helps to make the web more interactive and functional, when combined with HTML and CSS. It enables user interaction, such as detecting clicks and keyboard input, which leads to a better user experience. With JavaScript's ability to modify the Document Object Model (DOM), pages can be updated instantly without reloading. It helps talk to servers using AJAX, bringing smooth data updates and easy interactions. JavaScript makes animations, effects, and API integration possible, which creates numerous interactive features that direct user focus and enhance website attraction. Overall, JavaScript is a key component of contemporary web development, supporting developers to create immersive, responsive, and dynamic web platforms [MDN

Web Docs, 2021c; W3Schools, 2021].

6.2.2 Data Visualization Libraries

Effective data visualization plays a crucial role in enabling users to comprehend and analyze EEG data. EPIWEB could approach two prominent JavaScript libraries for interactive data visualization:

Plotly.js

The Plotly.js charting library was created by Plotly Inc. [PlotlyInc.]. This open-source tool is essential for data analysis suites and provides a diverse range of visualisations. It includes an extensive selection of chart types, from common options such as line and bar charts to more specific ones like histograms and box plots. Notably, Plotly extends its range beyond Plotly.js and offers charting libraries for Python and R, which are based on the underlying Plotly.js technology. This integrated suite increases its usefulness across various programming languages. If the requirements are in the field of data science, then Plotly.js is the recommended library to utilize [Derflinger, 2021].

Chart.js

Chart.js is an open-source JavaScript charting library, which has been widely adopted [Chart.js]. The library also offers diverse plugins, which can expand its capabilities, ranging from streaming data to financial charts and crosshairs. The adaptability of Chart.js is showcased by its inclusion of eight pre-configured chart types, spanning basic choices such as bar and line charts, as well as more complex ones like bubble charts, scatter charts, and polar charts.

6.2.3 Frontend Frameworks

To streamline front end development and enhance UI design, a variety of frameworks have emerged. React, Angular, and Vue.js are among the most widely used frameworks, each offering unique advantages [StateOfJs]. React emphasizes component reusability, Angular's declarative approach simplifies complex UIs, and Vue.js prioritizes simplicity and integration.

Frontend Framework	Usage (%)
React	82
Angular	49
Vue.js	46
Svelte	21
Preact	13
Ember	8
Alpine.js	6
etc.	

Table 6.2: Frontend frameworks usage on 2022 according to State of JS survey. From [StateOfJs].

React

React, a JavaScript library developed by Facebook, excels in component-based architecture. Components encapsulate specific UI elements and functionality, fostering reusability and separation of concerns. This modular approach enhances maintainability and facilitates collaboration among developers.

Angular

Angular, a full-fledged framework by Google, embraces a declarative style that reduces the imperative code, making UI development more straightforward. Its two-way data binding and dependency injection mechanisms enhance efficiency and facilitate real-time updates. Angular also supports TypeScript, enhancing code maintainability and minimizing errors.

Vue.js

Vue.js is a highly regarded frontend development framework known for its simplicity and adaptability. It integrates with existing projects, making it accessible for both smaller projects and teams with an uncomplicated learning curve. Vue.js leverages a virtual DOM, ensuring faster performance compared to other frameworks. Its strong emphasis on component reusability streamlines code development and enhances its comprehensibility. While Vue.js documentation is concise and well-structured, there's also a supportive online community available to provide assistance when needed.

Chapter 7

Architecture

Architecture plays a crucial role in all projects as it delineates the system's overall framework and behaviour, encompassing its various components. This chapter focuses on selecting the appropriate architectural approach for developing EPIWEB.

The initial stage is to select the project's development technologies, as outlined in section 7.1. Followed by this, section 7.2 covers the fundamental concepts relating to application architecture utilizing the chosen backend framework. Further elaboration is provided in section 7.3, where the specific architecture type utilized for EPIWEB's creation is established, based on the requirements presented. The chosen architecture is then comprehensively explained through C4 model diagrams. Finally, Section 7.4 presents the Entity-Relationship diagram devised for the project.

7.1 Technologies

As we mentioned in the Technologies chapter (chapter 6), the project can freely choose the technology for development. However, this choice depends on ensuring that the technologies chosen meet the project's requirements. It was carefully considered which options could best fulfill the specified requirements. Furthermore, it's crucial to understand and become skilled in the technologies as this knowledge greatly affects the effectiveness and efficiency of the implementation procedure.

Backend Technologies:

- **Python and libraries for data processing** - as already mentioned EPILAB's algorithms and scripts will be replicated in Python due to its great popularity and community on data science.
- **Django** - Python-based framework, is an ideal choice for building the back-end of the system. Its extensive range of third-party packages, well-documented resources and vibrant community make it a superior option. Choosing frameworks from different programming languages could potentially lead to complications in terms of compatibility and coordination. Such a choice could require additional resources to maintain and support different components of the system, making Django the preferred and more efficient choice (like it was done on the first prototype - see section 3.3).

Frontend Technologies:

- **HTML5** - language used to build the platform's web pages.
- **JavaScript** - high-level programming language, better known as a scripting language for web pages.
- **jQuery** - JavaScript library that allows you to execute AJAX requests and easily manipulate the Document Object Model (DOM) [jQuery].
- **Plotly.js** - open-source JavaScript library that allows the creation of interactive graphics. This technology was used to generate all the graphics in the platform.
- **Tailwind CSS** - open-source Cascading Style Sheets (CSS) framework. Unlike other technologies such as Bootstrap, it works by scanning all your HyperText Markup Language (HTML) files and then generates the corresponding styles and writes them to a static CSS file [Tailwind CSS].

7.2 Model-View-Template (MVT)

Django follows the software architectural pattern MVT (see figure 7.1), which aligns with the well-known Model-View-Controller (MVC) pattern. MVT provides a clear separation of concerns in web application development, making it easier to manage and maintain complex projects.

- The **Model** defines how the data is structured and interacts with the database schema. It represents the database schema and data and can be defined using Python classes through Django's Object Relational Mapping (ORM) layer. Working with databases without writing raw SQL queries is made easier with this layer, which defines the structure of the database tables and their relationships.
- The **View** is responsible for rendering data to the user. The View in Django oversees the presentation of data to the user and controls the business logic. Views handle user requests, process data from the Model, and determine how to present that data. In Django, a view comprises a Python function that accepts a web request and provides a web response. The views may generate HTML templates, or they can return JSON data, or execute various actions based on the request.
- The **Template** in Django is accountable for producing HTML and other output formats. Templates permit defining the outline and arrangement of the user interface. The template language of Django facilitates the dynamic addition of data from the Model into the HTML document, thereby rendering the generation of dynamic and data-controlled web pages effortless. Templates are not relevant in the context of a frontend framework like React. As this project is not using any frontend framework, templates have their due significance.

In the context of the MVC pattern, the View determines the data shown to users, not its visual style. In Django, a View is a Python function linked to a specific URL, defining what data to display. Templates handle how data appears visually. The controller role is primarily managed by Django's framework, directing requests to the appropriate view based on URL configuration [Django Documentation, 2021].

The MVT procedure [Sara Bastian, 2021] can be demonstrated through the following step-by-step analysis:

1. user sends a URL request for a particular resource through Django;
2. Django framework searches for the appropriate URL resource;
3. if the URL path matches a View, that View is triggered;
4. the View interacts with the Model, retrieving the required data from the database;

- subsequently, the View generates an appropriate template, integrating the obtained data, and displays it to the user.

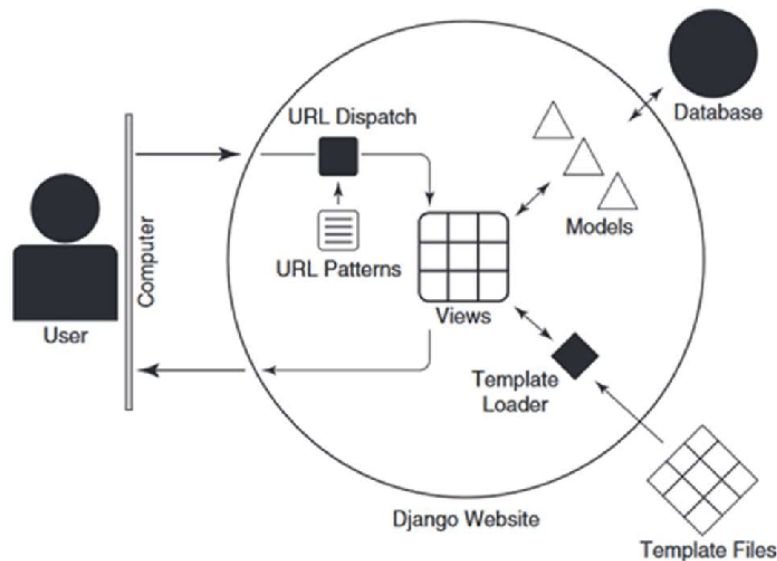


Figure 7.1: Django Architecture seen as MVT. From [Pinkham, 2016].

7.3 C4 Model

The C4 model [C4 Model] presents a hierarchical series of diagrams that display different levels of abstraction in a software system. These diagrams provide various viewpoints and aid comprehension of the system's architecture, allowing the development team to view different levels of detail. The model comprises four levels. Figure 7.2 presents examples of these different C4 model levels, highlighting the increasing depth of information provided for each part of the system as you move from top to bottom.

The **Context** diagram (Level 1) reveals the integration of the software system with its environment, highlighting its interaction with users and external systems.

The **Container** diagram (Level 2) expands the system to display its constituent containers, such as applications, microservices, and databases.

The **Component** diagram (Level 3) explores individual containers in detail, exposing their components and interactions both internally and externally.

Finally, the **Code** diagram (Level 4) delves into the code level specifics. Examine the code implementation of each component in great detail, with a focus on the finest level.

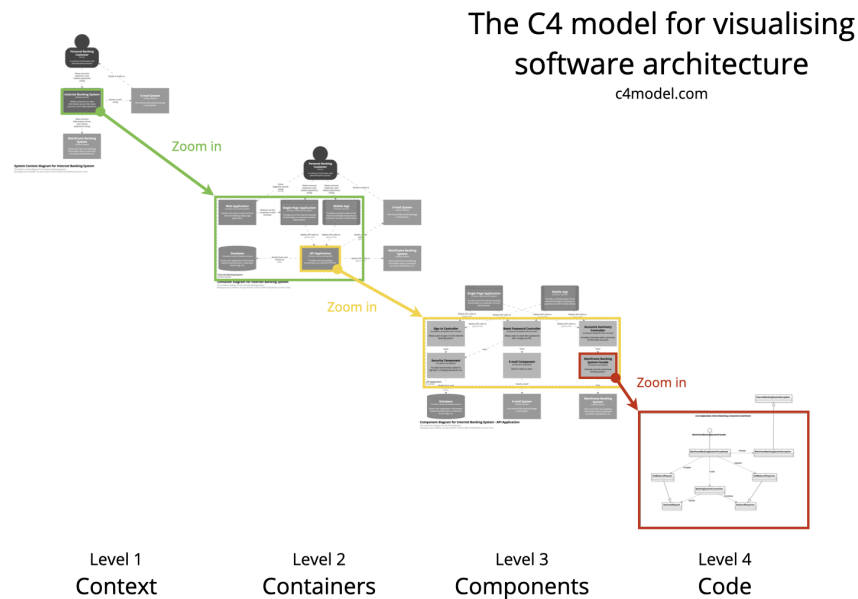


Figure 7.2: Enter Caption. From [C4 Model].

Subsequently, the EPIWEB architecture will be depicted using the various C4 model diagram levels.

Context Diagram

In figure 7.4, the context diagram illustrates EPIWEB's interactions with several external systems, all of which are libraries designed for specific technological functions. Two frontend libraries, Plotly.js and Tailwind CSS, contribute to improving the user experience by providing dynamic data visualisation and styling, respectively. As explained in chapter 1 Introduction, EPIWEB functions as a web platform for developing classification models that predict and detect epileptic seizures. This requires connection to several Python libraries that play essential roles. Numpy permits algebraic calculations, whereas pandas enables efficient dataset manipulation. For feature extraction like wavelet coefficients, the pywavelets library is used. Moreover, sci-kit learn offers tools for constructing machine learning models, and TensorFlow facilitates the creation of deep learning models.

EPIWEB's interactions extend beyond technological libraries to include two primary actors. The first is an unauthenticated user primarily engaged in accessing platform presentations and information without contributing content. The second actor is an authenticated user who has greater capabilities. This user can not only analyze information, but also create, modify and delete content within the system.

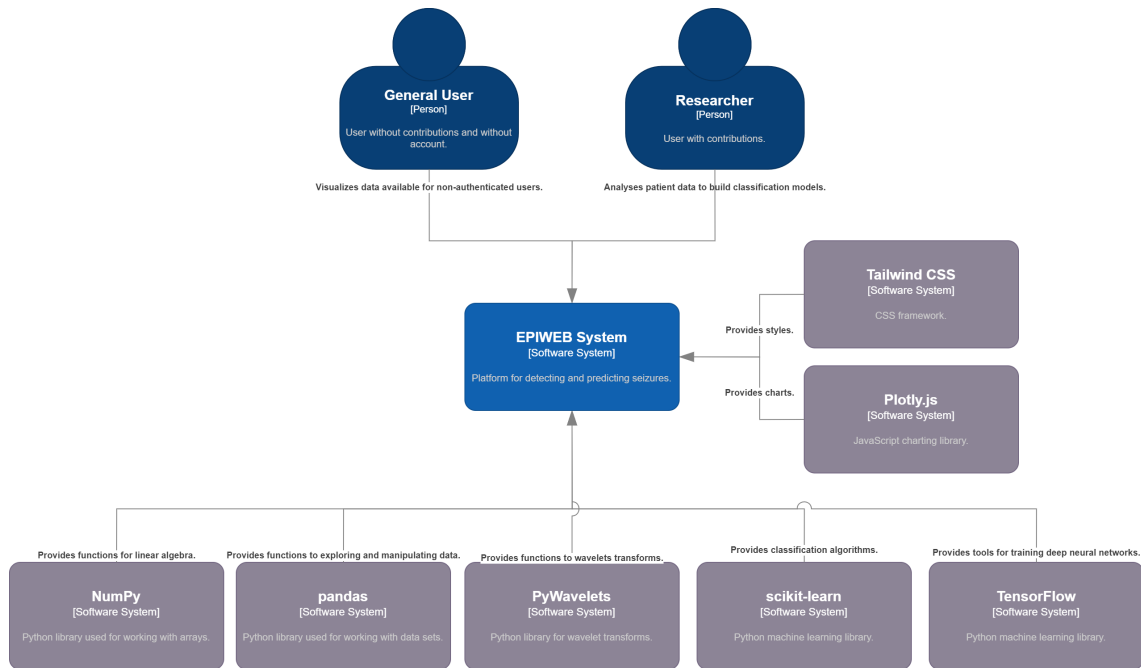


Figure 7.3: Context diagram.

Container Diagram

In figure 7.4, the container diagram presents a comprehensive breakdown of the software system’s architecture.

Communication within the system centres around HTTP requests, facilitated by request and response objects which propagate state throughout the architecture. This exchange occurs between the frontend, responsible for the user interface (HTML/CSS/JS), and the backend structured according to the Django MVT architecture. In this construct, the backend acts as the core container, expertly managing incoming requests from users. To successfully get this job done, the backend container is fitted with a range of functionalities, detailed shown in figure 7.6. These functionalities manage the handling of data that is stored in a PostgreSQL relational database, in order to obtain the necessary information.

Component Diagram

Figure 7.5 presents the component diagram, providing a comprehensive understanding of how the backend container operates, while figure 7.6 explores the details of the components which make up the container’s features.

One more time, it is worth noting that the diagram highlights two distinct user categories: the "Researcher" user, who has full access to the set array of platform functionalities, and the end user, who is granted access to a web page to create

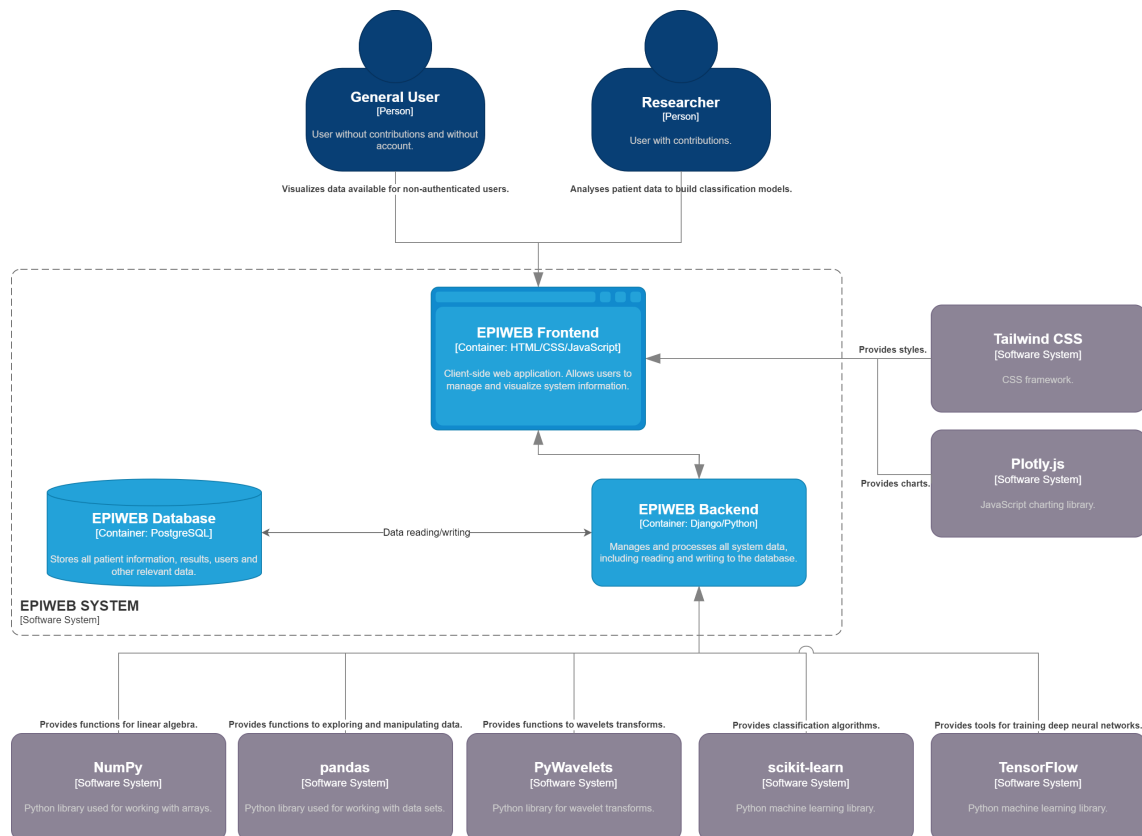


Figure 7.4: Containers diagram.

a personal account. Therefore, Authentication component is crucial for the proper functioning of the system. It guarantees that requests are accepted and correctly processed within the application. Using Django's built-in features, a specific Authentication component is created. This integral component effectively manages user authentication by overseeing user accounts, groups, permissions, and user sessions that are cookie-based. The Authentication component facilitates user authentication and verification, utilizing tokens within each request. This process ensures that authenticated users can proceed with their Hypertext Transfer Protocol (HTTP) requests. Upon successful authentication, user HTTP requests are sent to Core component that comply with the requirements set out for the project.

The Data Management component takes on a pivotal role, serving as an intermediary through which other components interact with the database. This interaction includes both reading and writing operations. With Django's database- abstraction API, the Data Management component enables the creation, retrieval, updating, and deletion of objects within the database, effectively encapsulating the interaction with data storage.

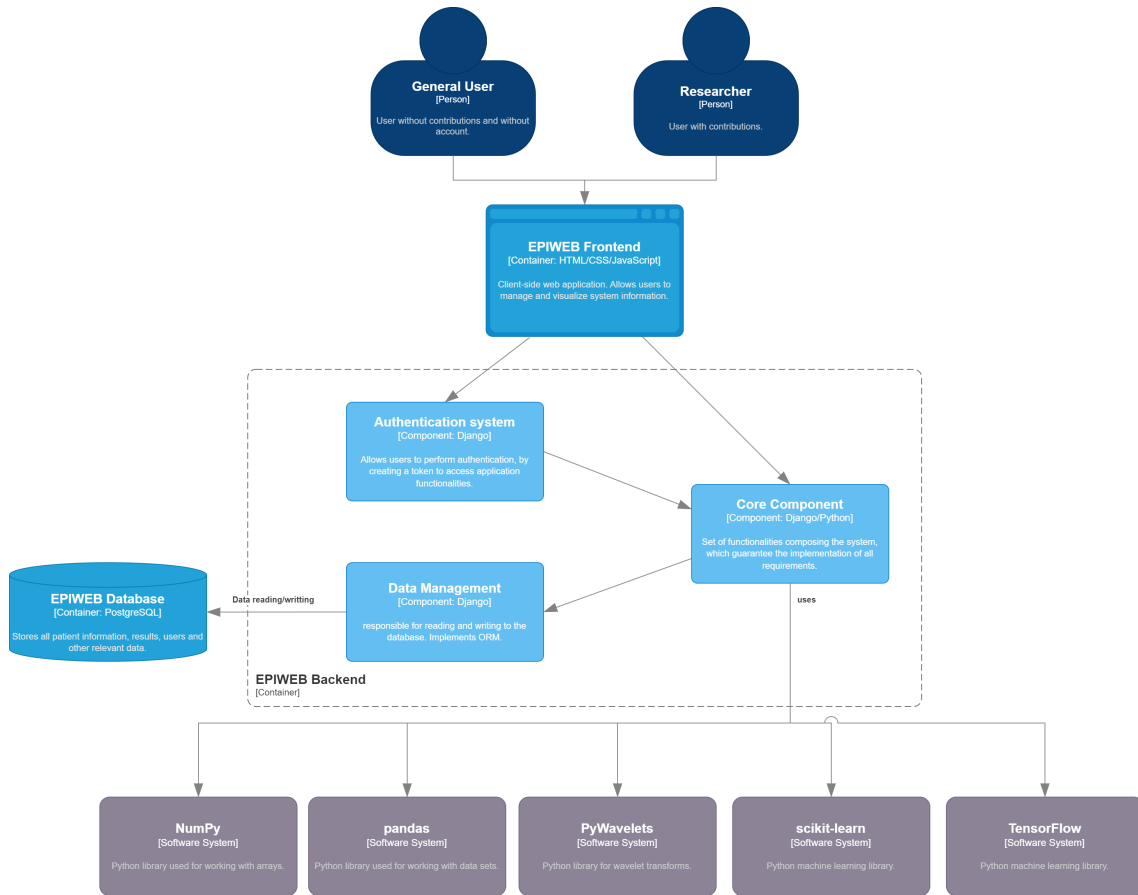


Figure 7.5: Components diagram.

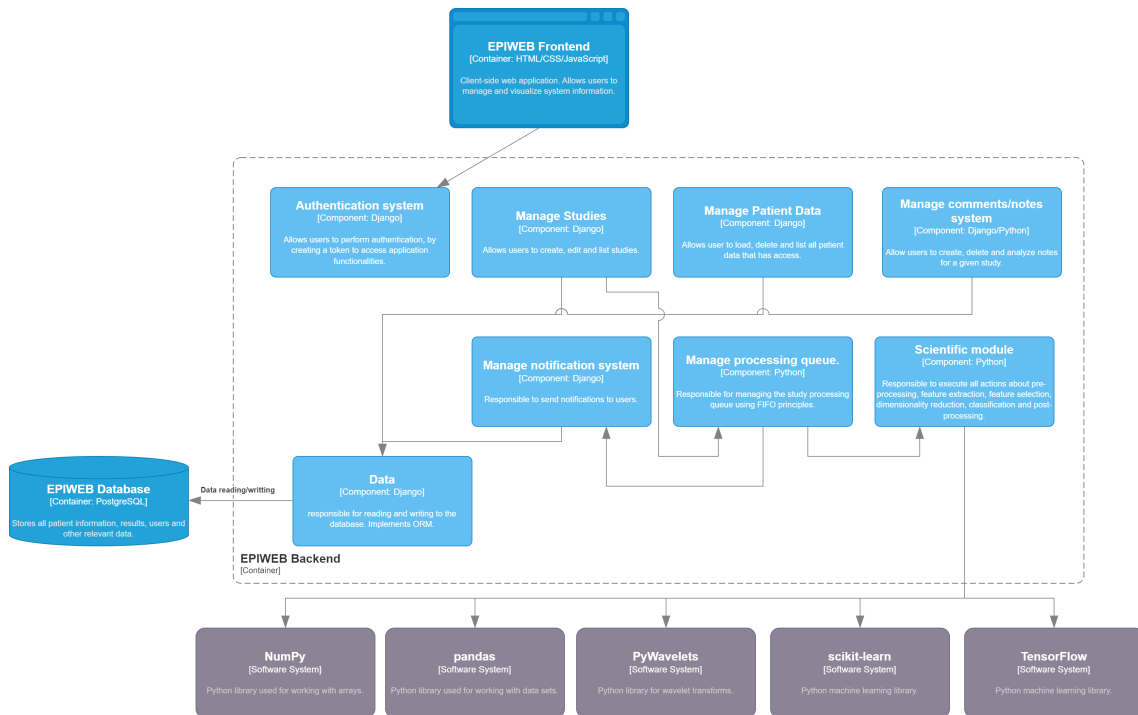


Figure 7.6: Components diagram for Core component.

7.4 Entity-Relationship Diagram

The Entity-Relationship (E-R) Diagram provides a visual representation of the database structure that provides EPIWEB's functionalities. This section offers an overview of the key entities, their significant attributes, and the relationships that link them together.

Developing an E-R diagram is extremely useful for gaining insights into the different entities within the problem domain and their interconnections. Additionally, it provides a comprehensive view of each entity's attributes, which clarifies the data that should be stored in the system's databases.

The E-R diagrams in this study were created using the ONDA tool (Online Database Architect) [ONDA - Online Database Architect, 2022] from Department of Informatics Engineering (DEI). This tool facilitates the creation of the database's conceptual model and automatically generates the physical diagram. The generated physical diagram offers insights into foreign keys and auxiliary tables required for a comprehensive database structure (see figure 7.8). The E-R model from the figure 7.7 delineates distinct entities and clarifies their relationships within the system, contributing to a deeper understanding of the system's data architecture.

With the E-R diagram we can distinguish the following entities and how they relate to each other in the system:

- **Patient:** This entity represents a patient (raw data) uploaded by users into the system. It relates to other entities such as:
 - Information: responsible for storing information from each set of data relating to a patient.
 - Event: responsible for signaling in time the occurrence of a certain event, such as epileptic seizures, for the given patient.
 - Comment: responsible for saving notes/extra information on the patient entered by users.
- **Client:** This entity stores users who are registered on EPIWEB. It relates to other entities such as:
 - Comment: is the user who enters comments on studies or patients.
 - Patient: the user is responsible for uploading the data set for a specific patient to the system. Study: it is the user who creates a study.

- **Hasaccess:** This entity intends to manage access to studies or patients for users other than those who uploaded the data to the system or who created the study. In this way, it relates to other entities as follows:
 - Study: the study that wants to be shared.
 - Client: identification of the user who will have access.
 - Patient: the patient and their data that wish to be shared with other users.
- **Study:** This entity represents a study created in EPIWEB and stores basic information about it. It relates to other entities such as:
 - Client: is the user responsible for creating the study.
 - Patient: the study focuses on data from a single patient.
 - Comment: responsible for saving notes/extra information in the study introduced by users.
 - Notification: responsible for saving study updates.
 - Extraction: the study can be of the feature extraction type.
 - Selection_Reduction: the study can be of the feature selection or dimensionality reduction type.
 - Classification: the study can be a classification study.
- **Selection_Reduction:** intends to save feature selection or dimensionality reduction studies carried out by EPIWEB users, with all parameters entered. It relates to other entities such as:
 - Extraction: feature selection is performed on the results obtained from a feature extraction study.
- **Classification:** intends to save classification studies carried out by EPIWEB users, with all parameters entered. It relates to other entities as follows:
 - Extraction: classification can be based on features extracted from raw data (patient).
 - Selection_Reduction: classification can be based on a set of data with previously selected features or with reduced dimensionality.
 - Patient: classification can be based directly on raw data (e.g. Deep Learning).

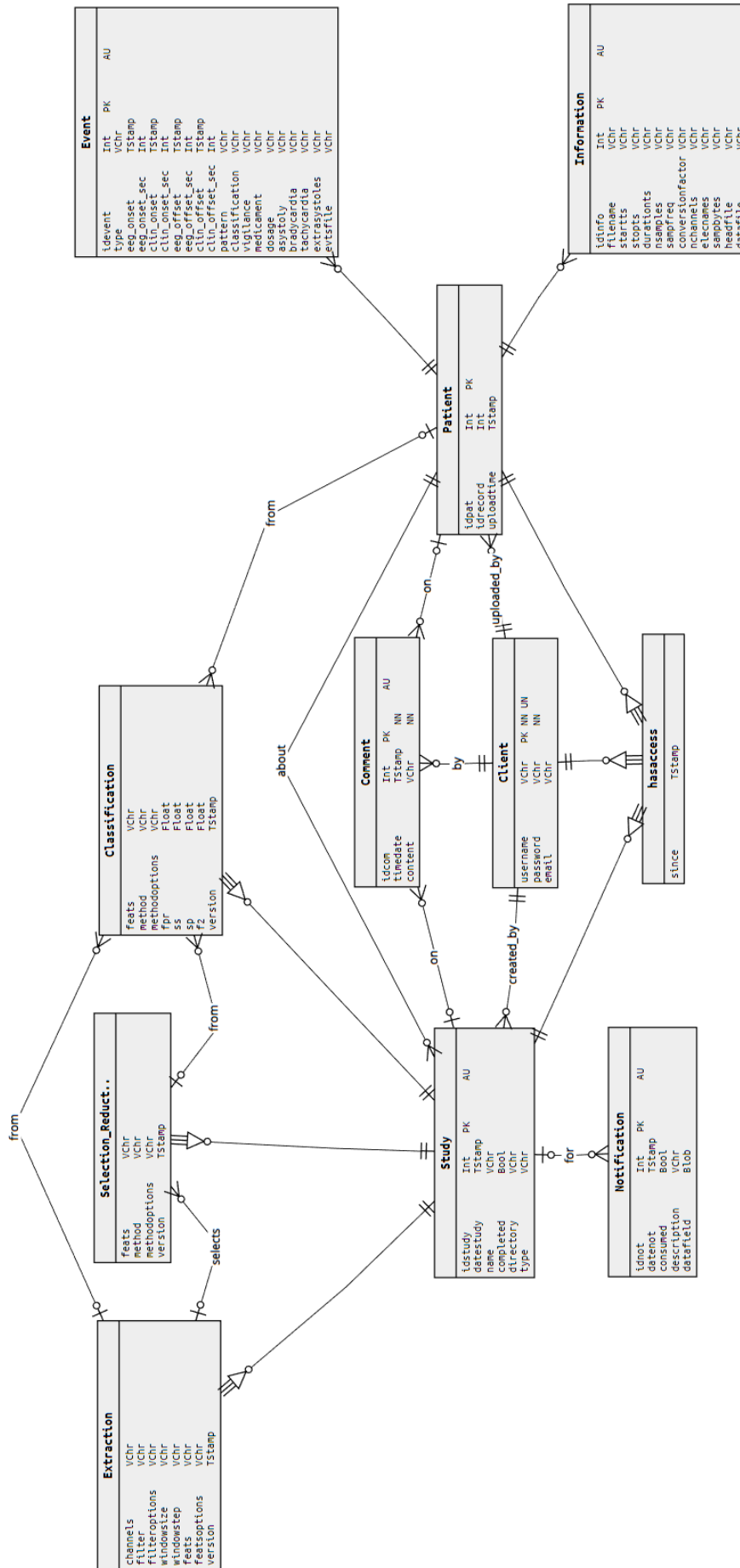


Figure 7.7: Entity-Relationship model diagram.

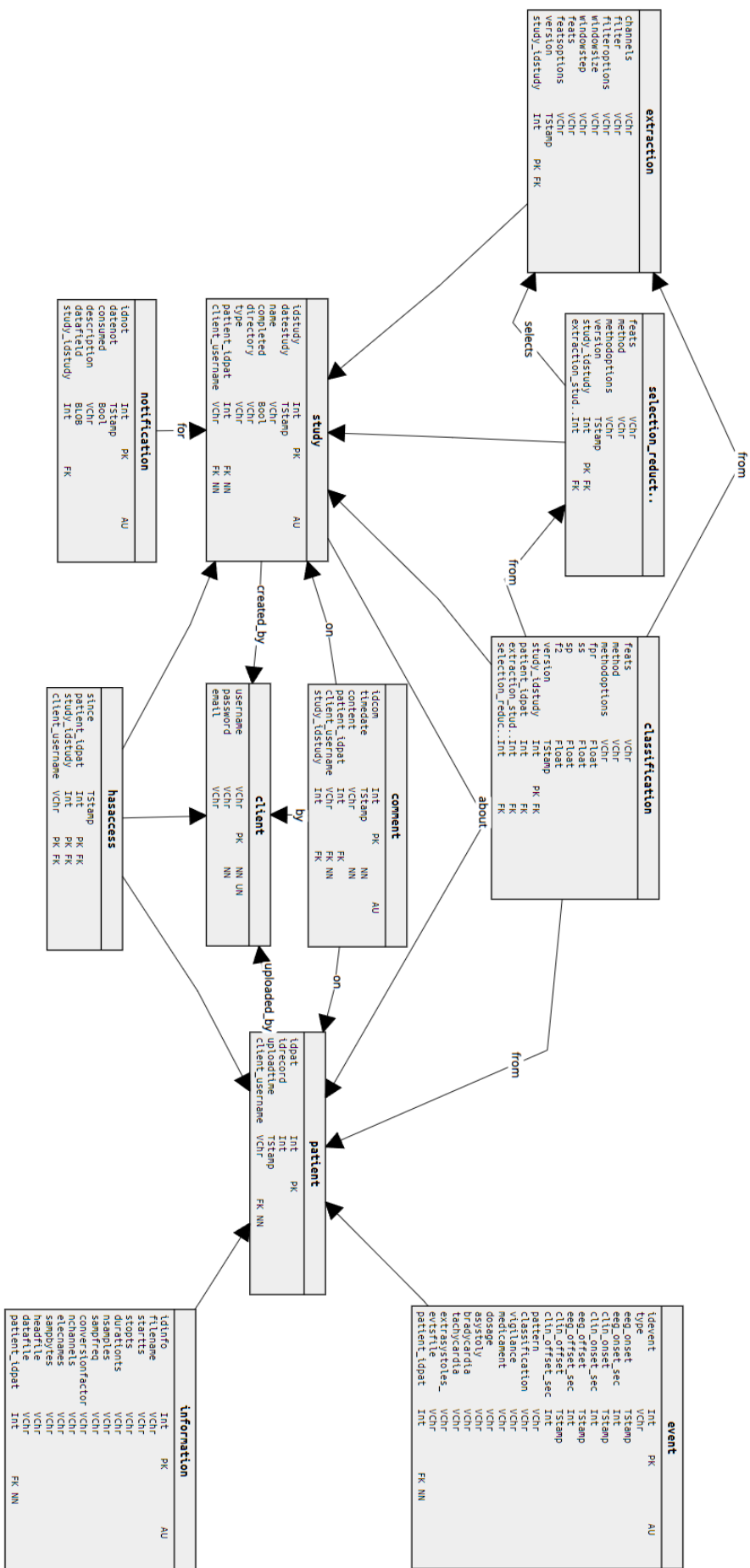


Figure 7.8: Physical model diagram.

Chapter 8

Development

The implementation phase of the EPIWEB project encompasses a range of steps, activities and challenges. Firstly, section 8.1 will delve into an analysis of select implementations and practices, with a particular emphasis on the Django framework's role in ensuring robust security and privacy measures throughout the development process. Section 8.2 will explore the challenges encountered during the development journey, contextualizing them within the initial planning. Lastly, a presentation and demonstration of the fully developed platform, EPIWEB, in its final state, in section 8.3. It provides an overview of the main features, functionalities and user interface of the platform

8.1 Important Practices

To ensure **secure access** to the server and protect **data privacy**, certain effective practices were employed using Python and Django. As discussed in section 7.2, the system follows an Model-View-Template (MVT) model where Views mediate between Templates and Models, responding to Template events and processing requests by interacting with the Model component. Basically, the View handles the system's logic for processing server requests. To maintain data confidentiality and privacy within these views, specifically making sure that only authorized users can access the server and its data, decorators were utilized [Django]. This means that if an unauthenticated user attempts a request, they will be redirected to the landing page for authentication. This approach effectively prevents non-authenticated users from accessing information related to existing studies or registered patients in the database. An example where a decorator is used is shown in the code block 8.1 line 1.

```
1 @login_required(login_url='landing')
2 def index(request):
3     userstudies = Study.objects.filter(user=request.user)
4     patients = Patient.objects.filter(user=request.user)
5     return render(request, 'app/home.html', {
6         'userstudies': userstudies,
7         'patients': patients,
8     })
```

Listing 8.1: A EPIWEB View portion with a login required decorator - authentication verification.

Below is another example of a decorator, code block 8.2 line 1. This decorator verifies whether there is an loggedin user in the current session. Only in this scenario the access to the login page be granted. Allowing a user who is already logged in to access this page would be illogical. Similar decorators are employed, often used to verify IDs when accessing specific information. This approach ensures both security and reliability, while also avoiding unnecessary code repetition.

```
1 @unauthenticatedUser
2 def loginUser(request):
3     form = CustomUserCreationForm()
4     context = {'form': form}
5     if request.method == 'POST':
6         ...
7     return render(request, 'app/login.html', context)
```

Listing 8.2: A EPIWEB View with a unauthenticated user is required decorator - nonauthentication verification.

In order to authenticate each request made to the server, Django employs sessions and middleware to integrate the authentication system into request objects. This enables the availability of a `request.user` attribute within every request, representing the current user's identity.

To enhance overall security, additional middleware components are configured within the project. A noteworthy middleware, the `CsrfViewMiddleware`, plays a pivotal role in protecting against Cross-Site Request Forgery (CSRF) attacks. Such attacks occur when a malicious website attempts to carry out actions on your website using the credentials of an authenticated user who visits the malicious site. To prevent this, it is crucial to set up the `X-CSRFToken` header for all forms in Django templates and AJAX requests. Examples of this practice are illustrated in code blocks 8.3 (line 2) and 8.4 (line 7). By following this approach, the project adheres to the principle of ensuring secure methods, such as GET re-

quests, are protected against any side effects, while enhancing the protection of 'unsafe' methods, including POST, PUT, and DELETE requests, against potential CSRF vulnerabilities.

```

1 <form method="post" name="register">
2   {% csrf_token %}
3   {% for field in form %}
4     <label class="text-white text-sm font-bold mb-2" for="{{ field.
      auto_id }}">{{ field.label }}</label>
5     <br />
6     {{ field }}
7     {% if field.label == "Password" %}<p class="text-white text-5xl
      ">{{ field.help_text }}</p>{% endif %}
8     <br />
9   {% endfor %}
10  <button type="submit"
11      formmethod="post"
12      class="mt-6 bg-zinc-700 hover:bg-zinc-800 text-white font
      -bold py-2 px-4 rounded focus:outline-none focus:shadow-outline"
13      >
14    Submit
15  </button>
</form>

```

Listing 8.3: A EPIWEB Template form using CSRF tag.

```

1 const url = "{{ request.path }}";
2 $.ajax({
3   url: url,
4   type: "POST",
5   data:{
6     'patid': patient,
7     'csrfmiddlewaretoken': $('input[name=csrfmiddlewaretoken]')
      .val()
8   },
9   datatype:'json',
10  success: function(data) {
11    ...
12  },
13 });

```

Listing 8.4: A EPIWEB Template script calling an AJAX request using the CSRF token cookie.

Incorporating Django forms is recommended. Within Django's Form class, one of its core function involves scrutinizing and validating user-submitted data. This approach enhances code organization. Code block 8.2, line 3, showcases a declaration of one Django form class that is passed to the template as context,

employed to scrutinize user authentication. Code block 8.3 depicts in line 3 a declaration on the template.

Furthermore, great care was taken during development to prevent potential vulnerabilities caused by SQL injection attacks. This was accomplished mainly through using Django's Object Relational Mapping (ORM), instead of raw SQL queries. By following this approach, the project gains from the natural protection against SQL injection afforded by Django's ORM.

8.2 Challenges

Some challenges were faced throughout the implementation phase of the project. The initial work plan and any resultant delays are important to explore them. The complexity and goals, as well as the risks identified earlier, led to several challenges that required careful management and adaptation.

In the section 4.3 (Methodology chapter), dedicated to the risk analysis carried out before the development of the EPIWEB platform, several risks associated with its implementation were identified. The most prevalent risks are now reviewed taking in the account the experience and challenges encountered during the actual implementation process.

- **R1 - Insufficient Familiarity with Tools and Technologies:** This risk manifested as the developer needed to become proficient in various web development tools and technologies, including web frameworks and libraries. The learning curve associated with these technologies led to slower development progress. Efforts were invested in training and self-study to address this challenge.
- **R2 - Changes in project requirements:** As the project progressed, there were instances where project requirements evolved due to feedback from advisors, new insights into epilepsy prediction, or changing user needs, as expected. These changes impacted the project's timeline and required adjustments in the development approach. Regular meetings with project advisors allowed for ongoing discussions and adaptations to changing requirements. Prioritization of requirements and maintaining a flexible development plan helped accommodate these changes without causing significant delays.
- **R3 - Complexity of Electroencephalogram (EEG) Data Processing:** The processing of EEG data, a crucial aspect of the project, proved to be more in-

tricate than initially envisioned. Developing accurate algorithms for seizure prediction required extensive testing, optimization, and fine-tuning. This complexity resulted in delays in achieving the desired level of accuracy, and a lot of EPILAB scripts were not developed.

- **R5 - Size of Work Too Ambitious:** The ambitious scope of the project, which aimed to transition from the EPILAB toolbox to a comprehensive web platform, presented challenges. Balancing the size of the work with the resources available, particularly as a solo developer, was a recurring issue. The vastness of the project's scope contributed to overall delays.

It is worth noting that the solo development aspect of this work played a significant role in the challenges encountered. As the sole developer, the responsibility for researching, designing, developing, testing and maintaining the entire system rested with a single individual. This meant a greater learning curve, increased workload and a potential bottleneck in addressing challenges in a timely manner.

In addition, the lack of a team to share tasks and responsibilities could have contributed to delays in certain areas, particularly in tackling complex tasks such as EEG data processing and algorithm development, resulting in longer development times.

8.3 Outcomes

In this section, a concise overview of the EPIWEB platform's development will be provided. All features can be accessed via the web browser and require a login. To authenticate, user must click the Sign In button on the landing page as displayed in the figure 8.1. Additionally, this page outlines the platform's purpose and functionalities for non-authenticated users. The figure 8.2 illustrates the login page, which allows users to access their account by inputting their email and password, provided they have already registered. Those who wish to create a new EPIWEB account can do so by completing the required fields in the registration section depicted in the figure 8.3.

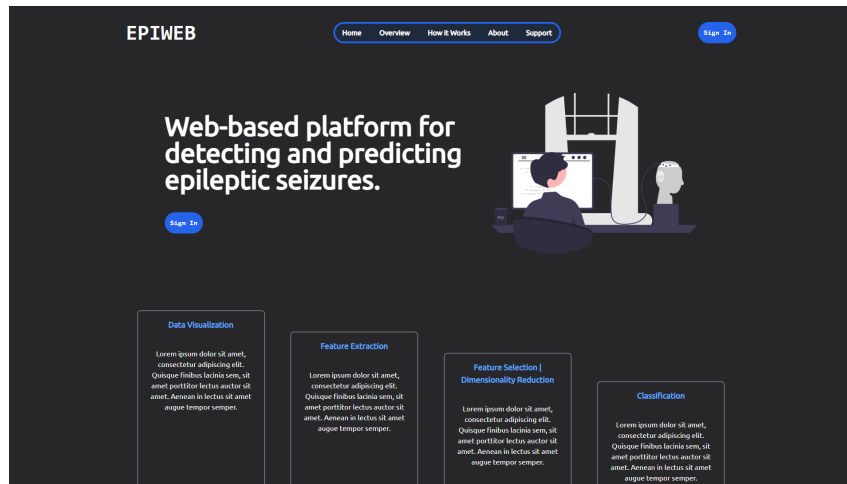


Figure 8.1: EPIWEB - Landing page.

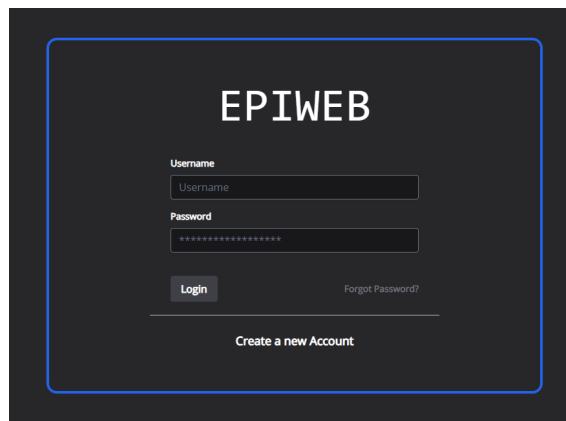


Figure 8.2: EPIWEB - Login page.

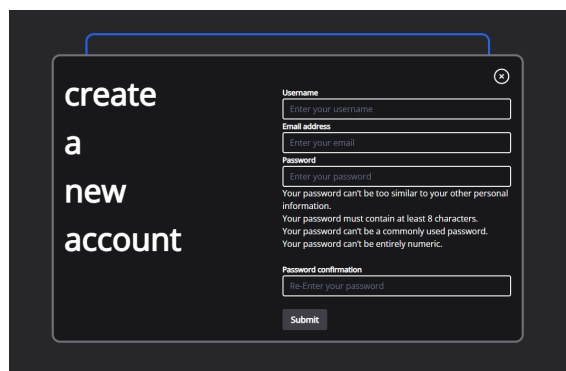


Figure 8.3: EPIWEB - Create a new account page.

As shown in the figure 8.4, after the session authentication, the user can access their homepage. Initially, they can observe the current status of their submitted studies. This allows them to determine whether any are still awaiting processing, being processed, or have already been completed. In the same figure is presented

a table that is available to display the user's recent activity related to studies created for easier access to each one, if desired. On this page, the user can access the pages containing the data loaded into the system through dedicated buttons. These pages list both raw data, that is, patient data, and studies that have already been conducted.

The screenshot shows the EPIWEB user authenticated home page. The header is dark blue with the EPIWEB logo on the left, a navigation menu (Data, Studies, About, Help) below it, and a 'Logout' button on the right. The main content area is white and contains a welcome message 'Hello admin, Welcome to EPIWEB'. Below the welcome message are two panels. The first panel, 'Your Studies Current Situation', shows a summary of study statuses: 5 Total, 0 On Queue, 0 Processing, and 5 Completed. The second panel, 'Your Recent Activity', is a table with columns for STUDY NAME, PATIENT, TYPE, and DATE, and a 'more details' link for each row.

STUDY NAME	PATIENT	TYPE	DATE	
2023-09-01_17-32-19	8902	Classification	Sept. 1, 2023, 5:32 p.m.	more details
2023-09-01_17-31-42	8902	Classification	Sept. 1, 2023, 5:31 p.m.	more details
2023-09-01_17-30-27	8902	Selection/Reduction	Sept. 1, 2023, 5:30 p.m.	more details
test_study	8902	Extraction	Aug. 4, 2023, 10:29 p.m.	more details
2023-08-02_18-23-24	8902	Extraction	Aug. 2, 2023, 6:23 p.m.	more details

Figure 8.4: EPIWEB - User authenticated home page.

On the listing page, it is also possible to upload new data to the system using the dedicated button. A form will appear, allowing you to import files (see figure 8.6). To manage a patient, select one from the list. When selected, a brief overview will be displayed, including the number of channels that collected the signals, the duration of the recording, and the sampling frequency, amongst other details. By clicking on the See Details button, the user will be directed to the specific page of that patient.

The screenshot shows the EPIWEB interface for patient 8902. At the top, there is a navigation bar with 'admin | Logout' and a breadcrumb 'Home > Data'. Below this are three main navigation buttons: 'Raw Data | Patients', 'Studies', and 'Classification Models Results'. The patient profile is displayed in a grid format:

Patient ID 8902	Number of Files 174	Channels -FP1 -FP2 -F3 -F4 -C3 -C4 -P3 -P4 -O1 -O2 -F7 -F8 -T3 -T4 -T5 -T6 -Fz -Cz -Pz -T1 -T2 -R5 -EOG1 -EOG2 -EMG -ECG -PHO	Start Time 2003-04-09 14:37:18
End Time 2003-04-16 14:34:47	Total Duration 6 days, 23:57:29	Sample Rate 256	Number of Samples 154636032

Buttons for 'See Details' and 'Upload new Raw Data' are also visible.

Figure 8.5: EPIWEB - Patient raw data management page.

This screenshot shows the same patient profile as Figure 8.5, but with an 'Upload new Raw Data' modal dialog box open. The dialog contains the following text:

Upload new Raw Data
You can import one dataset - folder - to your profile, at a time.
Make sure to import all the data and the respective information files on a individual folder:
(HEAD) (DATA) (EVENTS)

Files:
Escolher Ficheiros Nenhum Ficheiro selecionado

Buttons for 'Close' and 'Submit' are located at the bottom right of the dialog.

Figure 8.6: EPIWEB - Patient raw data management page.

On the patient's page, depicted in the figure 8.7, further information is featured in addition to the former details, including information about the events that could take place during the specified timeframe, e.g. epileptic seizures. Additionally, situated at the bottom of the page, a table is presented highlighting the studies that have been conducted using this dataset, for fast access. Still on this page, the user has the option of publishing comments, serving as notes for the user or sharing experiences with other users who also have access to this dataset. Additionally, the user can opt to conduct a feature extraction study.

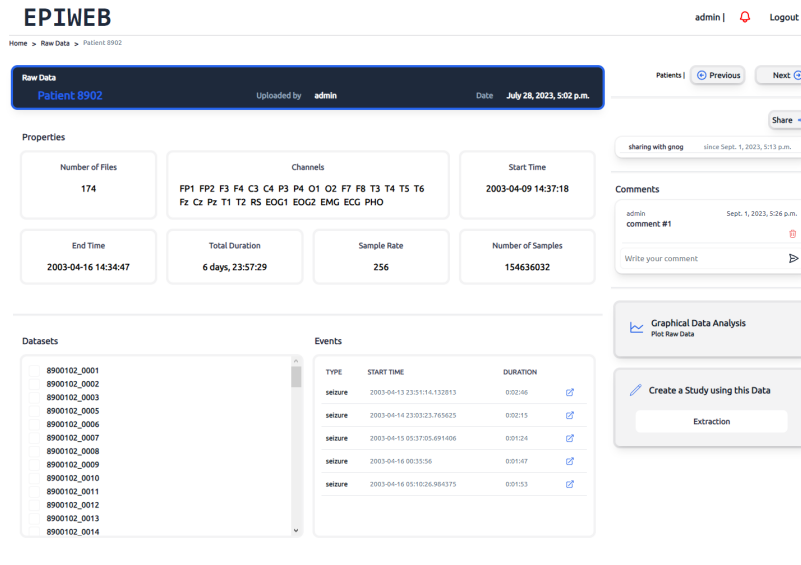
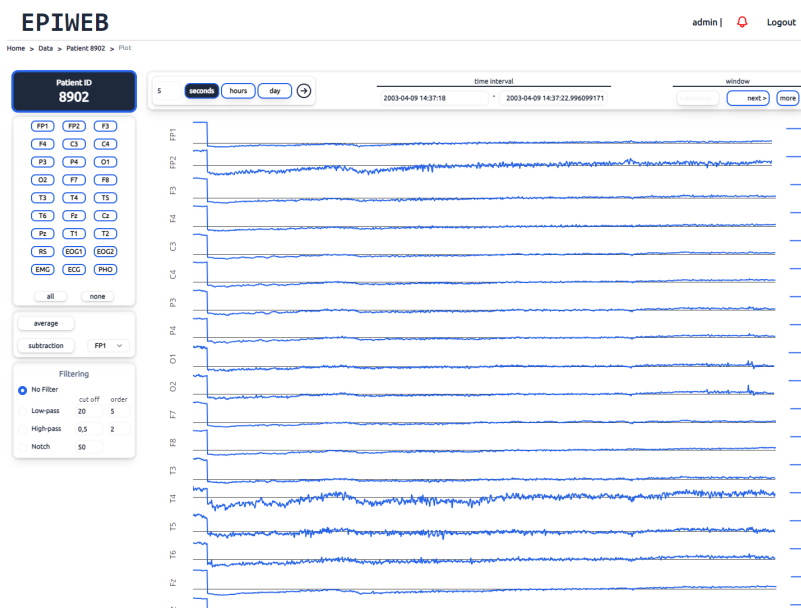
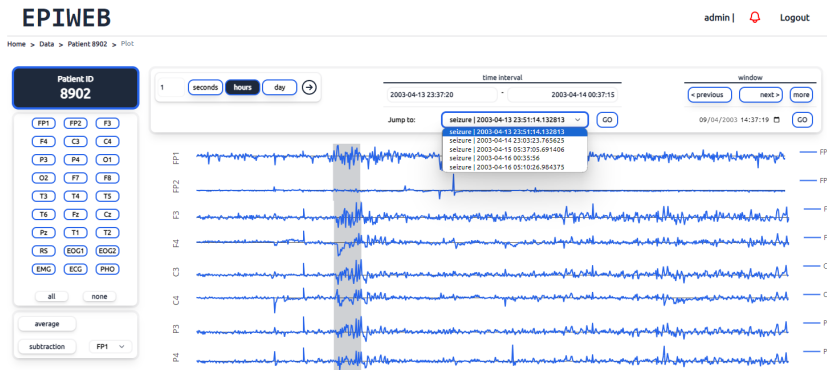


Figure 8.7: EPIWEB - Patient raw data information page.

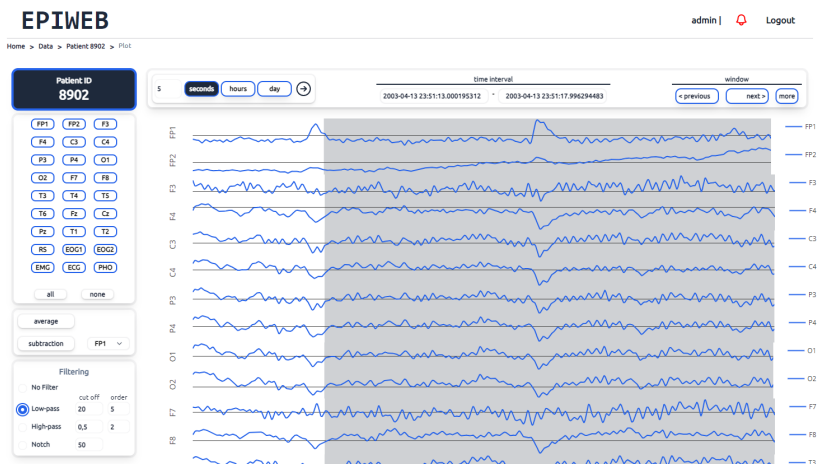
Clicking the button to display the patient's raw data graphically leads to the page depicted in the figure 8.8a. Numerous options are available on this page that the user can utilize to analyze the signals in the most optimal manner. The window size can be adjusted to seconds, hours, or days, and it is possible to navigate to a specific date and time of the signals. In real-time, the user can perform filtering by defining parameters such as the filter type, cutoff frequency, and order of the filter. Real-time calculation of the average signal for the current window and comparison with other channels allows for objective conclusions. Shading in the graphs highlights significant events.



(a)



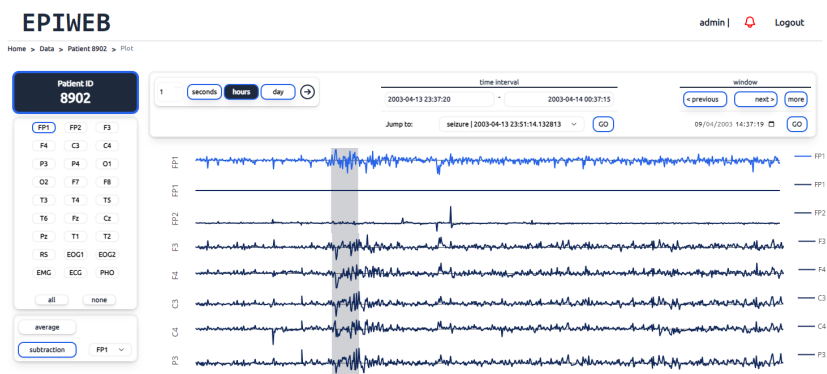
(b)



(c)



(d)



(e)

Figure 8.8: EPIWEB - Patient raw data graphical visualization.

As demonstrated in the figure 8.9, the list of conducted studies can be filtered by either text contained in each table row or study type.

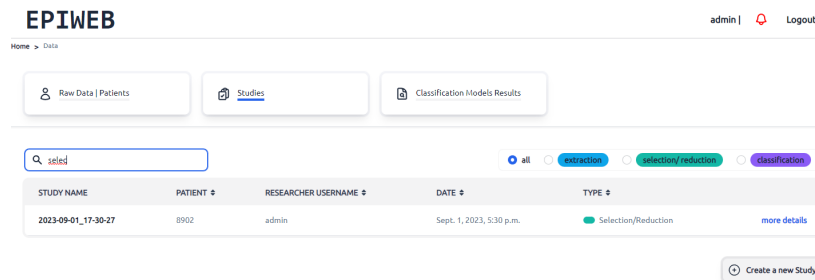


Figure 8.9: EPIWEB - Studies management page.

By clicking the New Study button, the user can access the page displayed in the figure 8.10. To generate a feature extraction study, the user must first select the raw data required to produce the features. When selecting the data set, the user will have access to studies that have already been conducted on it, relevant information about it that may be crucial when selecting parameters, and a graphical overview of it. From this point, the user may select options for extraction, such as dataset channels, desired features, pre-processing techniques - such as filtering - and the size and step of the windows used to calculate the feats. If the user hovers the cursor over help icons or specific features, a help panel will appear, providing explanations for the associated feature or procedure, as demonstrated in the same figure 8.10.

Once the study parameters have been selected, the user should click on the Create Study button. The parameter confirmation panel will appear, summarising the chosen parameters and allowing the user to name the study (see figure 8.11).

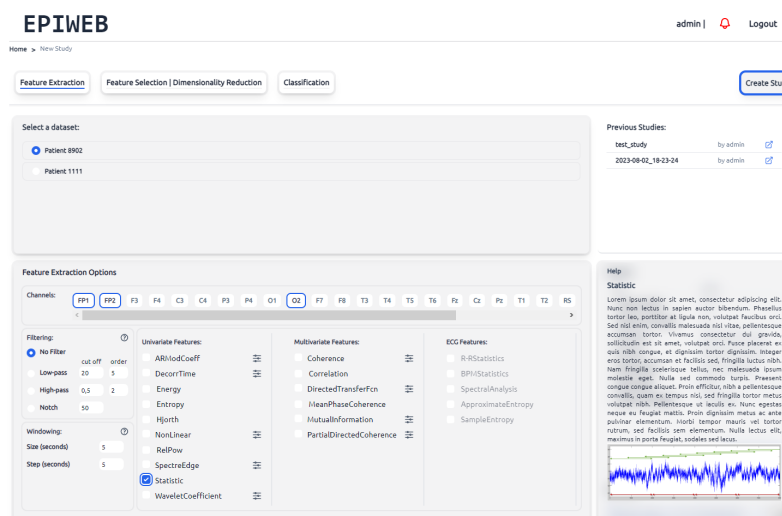


Figure 8.10: EPIWEB - Extraction study creation page.

Figure 8.11: EPIWEB - Study creation confirmation form.

On the study listing page (figure 8.9), if the user selects a feature extraction study, they will be directed to its information page (shown in the following figure 8.12). Here, user can access various details about the study including its creation date, author and selected parameters, as well as the results obtained. As seen on the patient page, the information is presented objectively with a logical flow. There is also a system for comments, details on the users that have access to it, and a table of subsequent studies based on this research.

If the user clicks Edit, the figure 8.13 panel will be displayed with the current parameters already selected. Then, through choices, the results of the study will be updated or the study itself will be carried out again now with other parameters.

STUDY NAME	RESEARCHER USERNAME	DATE
2023-09-01_17-30-27	admin	Sept. 1, 2023, 5:30 p.m.

Figure 8.12: EPIWEB - Extraction study information page.

Figure 8.13: EPIWEB - Extraction study edit form.

To visually display the results of the study, as demonstrated in the figure 8.8a, the user should click the designated button on the study page. The page functions similarly to the visualization of raw data.

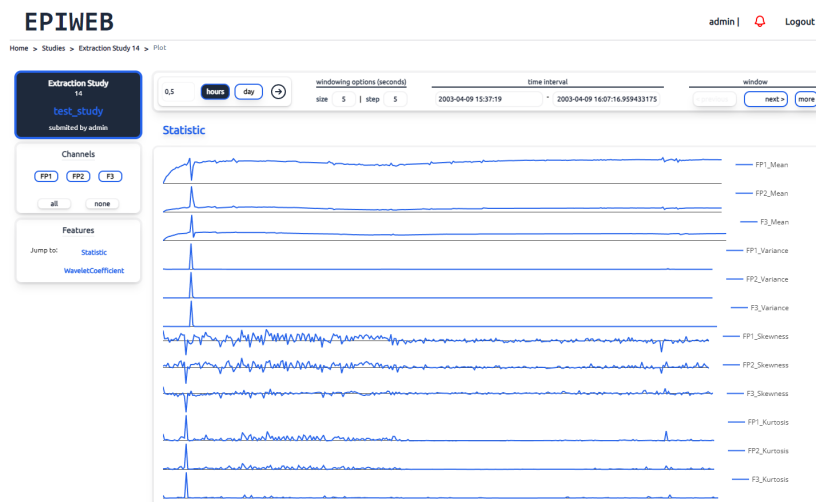


Figure 8.14: EPIWEB - Extraction study graphical visualization.

For studies involving Feature Selection/Dimensionality Reduction and Classification, the creation and information pages exhibit similar dynamics to those for feature extraction. The following figures demonstrate these functionalities.

Figure 8.15: EPIWEB - Selection/Reduction study creation page.

STUDY NAME	RESEARCHER USERNAME #	DATE	
2023-09-01_17-32-19	admin	Sept. 1, 2023, 5:32 p.m.	more details

Figure 8.16: EPIWEB - Selection/Reduction study information page.

The screenshot shows the EPIWEB interface for creating a classification study. At the top, there are navigation links for 'admin' and 'Logout'. Below the header, there are tabs for 'Feature Extraction', 'Feature Selection | Dimensionality Reduction', and 'Classification', with a 'Create Study' button on the right. The main content area is divided into several sections:

- Select a dataset:** A list of three datasets: #15 (Feature Selection | Dimensionality Reduction), #14 (test_study), and #13 (2023-08-02_18-23-24).
- Classification Options:** A section with tabs for 'SVM' and 'ANN' (selected). Below this, there are options for 'Number of Classes' (2 and 4), 'Network Type' (Feed-forward backprop and Recurrent), and 'Network Properties' (Training Function, Epochs, Learning Function, Goal, Performance Function, Pre-ical, Threshold Level, Early Detection Time).
- Layer Properties:** A section for configuring the neural network layers, including 'Layer 1' and 'Layer 2' with options for 'Number of Neurons' and 'Transfer Function'.
- Feature Extraction Study Summary:** A table showing key metrics: ID, Channels, Features Group, Filter, Window Size, Window Step, Number of Samples, and Number of Features.
- Extracted Feature data:** A section with a 'Statistic' plot showing waveforms for FPI_Mean and FPI_Min.

Figure 8.17: EPIWEB - Classification study creation page.

The screenshot shows the EPIWEB interface for viewing a classification study. At the top, there are navigation links for 'admin' and 'Logout'. Below the header, there are tabs for 'Classification Studies | Previous' and 'Next'. The main content area is divided into several sections:

- Classification Study Summary:** A dark blue box containing the study ID '2023-09-01_17-32-19', the creator 'admin', the date 'Sept. 1, 2023, 5:32 p.m.', and the status 'Completed? True'.
- Study Options:** A table showing key metrics: 'From Features FPI_Statistic |', 'Method ANN', 'Number of Samples 3600', 'Number of Features 1', and 'Number of Classes 2'.
- Model Performance Evaluation:** A section with four buttons for 'Sensitivity', 'Specificity', 'F2 Score', and 'False Positive Rate', each with a '-%' placeholder.
- Comments:** A section with a 'Write your comment' input field and a 'Share' button.
- Graphical Data Analysis:** A button labeled 'Graphical Data Analysis Plot Results'.

Figure 8.18: EPIWEB - Classification study information page.

When creating a new study, the status of the processing queue's live update panels is updated every few seconds on both the homepage and in the notification system, as illustrated in the figures 8.19 and 8.20 below. Clicking on a notification will redirect the user to the study that it is about to.

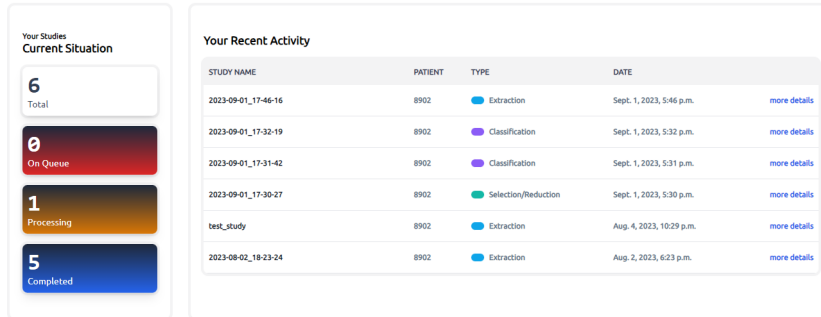


Figure 8.19: EPIWEB - Real time study queue update system on home page.

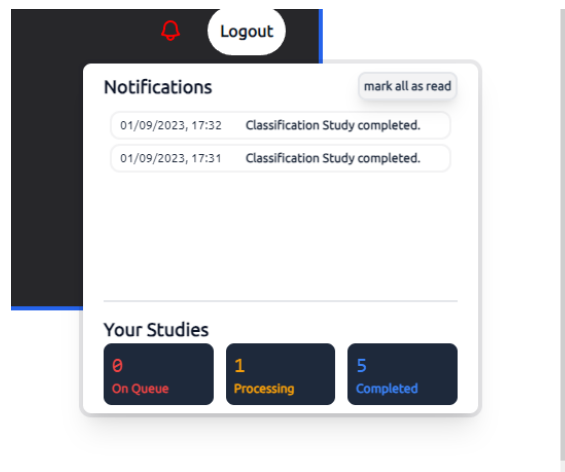


Figure 8.20: EPIWEB - Real time study queue update system on notifications element.

The user can access a page via the list of registered data, where the developed classification models are organized by patient. Upon selecting a particular model, users can view details such as the features utilized and the studies employed to construct the model. The following figure 8.21 illustrates that.

PREDICTORS	MODEL TYPE	PARAMETERS	TRANSFERFUNCTIONS	PREICTAL (MIN)	FPR	SS	SP	F2
2023-09-01_17-31-42	ANN	13 25	tansig tansig	30	0.01	43	90	80
2023-09-01_17-32-19	ANN	13 25	tansig tansig	30	0.01	43	90	80

Figure 8.21: EPIWEB - Classification models page.

As previously mentioned, patients or studies registered by a particular user can be shared with others via the Share button present on all relevant pages. The figure 8.22 illustrates the panel where the user will need to enter the username of the user with whom they wish to share the study or patient. Following confirmation of the sharing request, the user is provided with details of the individual they are sharing with and the duration of time since sharing commenced.

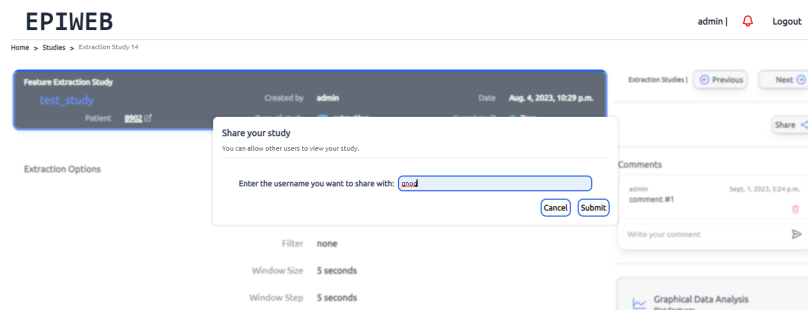


Figure 8.22: EPIWEB - Sharing raw data/studies system.

Chapter 9

Validation and Verification

The validation process played a crucial role in guiding the development of EPIWEB. The weekly meetings were conducted to review progress and ensure that the project was on track. These meetings served as checkpoints to assess whether the development efforts were aligning with the goals and objectives.

This chapter also explains the verification process the application showing how EPIWEB meets the defined functional and non-functional requirements, taking in account the objectives outlined in previous chapters (chapter 5).

To contribute to the evaluation of the product, it is crucial to verify and comprehend its alignment with the functional requirements. In this regard, a table is presented below that outlines the functional requirements and their implementation status during the work period.

For the authentication requirements, all aspects were successfully implemented, with the exception of the password recovery feature. The omission of this feature is attributed to the need for integration with an external API, which was not within the scope of this project.

In the domain of raw data management, including patient data, all functional requirements were not only met but also executed. The ability to visualize primary data emerges as a fundamental feature for future studies, enhancing the platform's utility and user experience.

The necessary functions, such as generating and modifying studies, and some graphical display requirements, have been integrated on the platform. However, the scientific module of feature selection/dimensionality reduction or classification studies has not been developed. The platform's prioritisation was discussed with the advisors, with the replication of EPILAB scripts only being considered if time permitted. As time was limited, certain functionalities that depend on the

replicated algorithms were suspended, including graph visualisation for selection or classification studies.

In contrast, feature extraction studies, comprising all 'Must' priority requirements, were thoroughly implemented and integrated into the platform, providing a robust foundation for future enhancements.

ID	Name	Priority	Dependencies	Developed
RF-01	Landing Page	Must		T
RF-02	User Authentication	Must		T
RF-03	Password Recovery	Won't	RF-02	F
RF-04	User Registration	Must		T
RF-05	User Sign Out	Must	RF-02	T
RF-06	User Personal Home Page	Must		T
RF-07	Studies Processing Queue	Must		T
RF-08	Notification System	Must		T
RF-09	Recent Studies Table	Could		T
RF-10	Load Raw Data	Must		T
RF-11	Access Raw Data	Must	RF-10	T
RF-12	Manage a Specific Raw Data	Must	RF-11	T
RF-13	Visualize Graphically Raw Data	Must	RF-11	T
RF-14	Filter Channels on Graphic Visualization	Must	RF-13	T
RF-15	Analyze Average Plot	Must	RF-13	T
RF-16	Analyze Subtraction Plots	Must	RF-13	T
RF-17	Change Window Size of Visualization	Must	RF-13	T
RF-18	Jump to Specific Time	Must	RF-13	T
RF-19	Jump to Specific Event	Must	RF-13	T
RF-20	Real-Time Filtering	Must	RF-13	T
RF-21	Share Raw Data	Must	RF-11	T
RF-22	Comment on Raw Data	Must	RF-11	T
RF-23	Access to Studies	Must		T
RF-24	Manage a Specific Study	Must	RF-23	T
RF-25	Comment on a Study	Must	RF-23	T
RF-26	Share a Study	Must	RF-23	T
RF-27	Create a Feat. Extract. Study	Must		T
RF-28	Visualize Graphically in a Feat. Extract.	Must	RF-23	T
RF-29	Filter Graphs by Features in a Feat. Extract.	Could	RF-28	T
RF-30	Change Window Size in a Feat. Extract. Study	Should	RF-28	T
RF-31	Jump to Specific Time in a Feat. Extract. Study	Should	RF-28	F
RF-32	Jump to Specific Event in a Feat. Extract. Study	Should	RF-28	F
RF-33	Edit a Feat. Extract. Study	Should	RF-27	T
RF-34	Create a Sel. Red. Study	Must		T
RF-35	Visualize Graphically in a Sel. Red. Study	Must	RF-23	F
RF-36	Filter Graphs by Features in a Sel. Red. Study	Should	RF-35	F
RF-37	Change Window Size in a Sel. Red. Study	Won't	RF-35	F
RF-38	Jump to Specific Time in a Sel. Red. Study	Won't	RF-35	F
RF-39	Jump to Specific Event in a Sel. Red. Study	Won't	RF-35	F
RF-40	Edit Sel. Red. Study	Should	RF-34	T
RF-41	Create a Classification Study	Must		T
RF-42	Visualize Classification Study Graphs	Must	RF-23	F
RF-43	Change Window Size in a Classification Study	Won't	RF-42	F
RF-44	Jump to Specific Time in a Classification Study	Won't	RF-42	F

Table 9.1: List of functional requirements with the indication if they were developed.

In this section, the alignment with non-functional requirements outlined in chapter 5.1, encompassing usability, security, privacy, and performance is explained. While the platform is currently in the developmental phase, preliminary assessments have been conducted to ensure these aspects are being effectively addressed.

Usability testing aims to evaluate the user-friendliness and intuitiveness of the platform. Despite being in the development phase, initial usability tests were carried out by soliciting feedback from project advisors, yielding positive results.

Security and privacy measures are paramount for safeguarding user data and preventing unauthorized access. Throughout the development phase, a range of security best practices were implemented, as detailed in chapter 8, including the integration of Cross-Site Request Forgery (CSRF) tokens and the secure utilization of Django Object-Relational Mapping (ORM).

Preliminary **performance** tests have been undertaken to evaluate platform responsiveness and efficiency. While the platform has yet to be deployed in its final environment, assessments were made to measure the duration of key functions (e.g., data loading, graphical processing) to completion.

Chapter 10

Conclusion

In this dissertation, it is explained the planning and development of EPIWEB, a web-based tool designed to aid in the development and evaluation of classification models for predicting and detecting epileptic seizures. The project was initiated to replace the current MATLAB-based EPILAB toolbox with a more widely accessible and adaptable web platform.

The development process followed an agile-based approach, providing room for iterative and continuous improvement based on feedback. Weekly meetings with advisers provided useful insights and ensured alignment with project objectives. The selected technologies, such as Django for the backend and plain HTML/CSS/JavaScript for the frontend, were carefully chosen to fulfil the project's specific needs.

The validation of EPIWEB comprised a comprehensive evaluation of functional and non-functional requirements. Usability testing offered useful feedback on the user interface. Security measures, such as the implementation of CSRF tokens and the use of secure ORMs, were implemented to safeguard user data. Preliminary performance testing provided insights into the platform's responsiveness and efficiency, assisting future optimizations. About the functional requirements, the most important were executed.

Despite the achievements, certain challenges were encountered during implementation, comprising delays in project milestones and the ambitious nature of the project scope. Risk analysis and mitigation strategies were employed to address these challenges.

In summary, EPIWEB represents a significant development in the field of epilepsy research, offering a user-friendly and accessible platform for creating and evaluating classification models. While the platform is currently in the de-

velopment phase, it has established the basis for a robust, secure and efficient system. Future work will focus on replicating EPILAB algorithms and improving the platform based on user feedback and further testing in a production environment.

Finally, the development of EPIWEB was driven by a dedication and a comprehension of both Biomedical Engineering and Software development, during the course of this project. As the platform's developer, I, a Biomedical Engineering student, aimed to bridge the gap between advanced EEG analysis and accessible web-based tools. This project has not only enhanced my technical skills but has also strengthened my dedication to progress healthcare solutions through innovative technology. EPIWEB is evidence of the potential achieved through interdisciplinary collaborations and the influence that technology can have on altering the field of epilepsy research and diagnosis.

10.1 Future Work

The development of EPIWEB has provided a robust basis for future enhancements and research in the area of epilepsy prediction. The development of the remaining functional requirements outlined could be the first thing to do in the future.

One of the key aims for the future development of EPIWEB is the replication of EPILAB algorithms. This involves modifying existing MATLAB-based algorithms for Python implementation, and aligning them with the platform's architecture. Integration of these algorithms will considerably improve EPIWEB's functionality and analytical capabilities.

To transform EPIWEB from the development environment to a production one, will require careful consideration of scalability and deployment strategies. This involves choosing suitable cloud infrastructure, enhancing database efficiency, and introducing load balancing to ensure seamless user experiences.

Expanding EPIWEB's capabilities to integrate with external data sources, like EPILEPSIAE database, presents a promising opportunity to enrich the platform's data set and provide a more inclusive perspective of patient health.

Automating the entire machine learning pipeline, from data pre-processing to model deployment, could simplify the process for researchers and healthcare professionals. This includes implementing robust data management, automated feature selection, and model hyperparameter tuning.

References

- Epilepsiae - epilepsy project. Available at <http://epilepsiae.eu/>. Accessed on August 01, 2023.
- U. Rajendra Acharya, S. Vinitha Sree, G. Swapna, Roshan Joy Martis, and Jasjit S. Suri. Automated eeg analysis of epilepsy: A review. *Knowledge-Based Systems*, 45:147–165, 2013.
- Agile Business Consortium. Chapter 10: Moscow prioritisation, 2022. URL <https://www.agilebusiness.org/dsdm-project-framework/moscow-prioritisation.html>. Accessed on August 01, 2023.
- American Epilepsy Society. Seizure prediction challenge (2014), 2014. URL <https://www.kaggle.com/competitions/seizure-prediction/data>. Accessed on August 01, 2023.
- Ralph G. Andrzejak, Klaus Lehnertz, Florian Mormann, Christoph Rieke, Peter David, and Christian E. Elger. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Phys. Rev. E*, 64:061907, Nov 2001.
- Baeldung. Mvc with servlets and jsp - baeldung. URL <https://www.baeldung.com/mvc-servlet-jsp>. Accessed on August 01, 2023.
- Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, ..., and Jeff Sutherland. Manifesto for agile software development, 2001. URL <http://agilemanifesto.org/>. Accessed on August 01, 2023.
- A. T. Berg, S. F. Berkovic, M. J. Brodie, J. Buchhalter, J. H. Cross, W. van Emde Boas, J. Engel, J. A. French, T. A. Glauser, G. W. Mathern, S. L. Moshé, D. R. Nordli, P. Plouin, and I. E. Scheffer. Revised terminology and concepts for organization of seizures and epilepsies: Report of the ilae commission on classification and terminology, 2005-2009. *Epilepsia*, 51(4):676–685, 2010.

Elie Bou Assi, Dang K. Nguyen, Sandy Rihana, and Mohamad Sawan. Towards accurate prediction of epileptic seizures: A review. *Biomedical Signal Processing and Control*, 34:144–157, 2017. ISSN 1746-8094.

György Buzsáki, Costas A. Anastassiou, and Christof Koch. The origin of extracellular fields and currents—EEG, ECoG, LFP and spikes. *Nature Reviews Neuroscience*, 13(6):407–420, 2012.

C4 Model. C4 model. URL <https://c4model.com/>. Accessed on August 01, 2023.

Chart.js. Chart.js | open source html5 charts for your website. URL <https://www.chartjs.org/>. Accessed on August 01, 2023.

Luigi Chisci, Andrea Mavino, Giancarlo Perferi, Marco Sciandrone, Carmelo Anile, Gabriella Colicchio, and Fabio Fuggetta. Real-time epileptic seizure prediction using ar models and support vector machines. *IEEE Transactions on Biomedical Engineering*, 57:1124–1132, 2010.

Song Cui, Lijuan Duan, Yuanhua Qiao, and Ying Xiao. Learning eeg synchronization patterns for epileptic seizure prediction using bag-of-wave features. *Journal of Ambient Intelligence and Humanized Computing*, 2018. doi: 10.1007/s12652-018-1000-3.

Haneen Daoud and Magdy A. Bayoumi. Efficient epileptic seizure prediction based on deep learning. *IEEE Transactions on Biomedical Circuits and Systems*, 13(5):804–813, Oct. 2019. doi: 10.1109/TBCAS.2019.2929053.

Thomas Derflinger. Comparing 3 popular javascript charting libraries, 2021. URL <https://www.tderflinger.com/en/comparing-3-popular-javascript-charting-libraries>. Accessed on August 01, 2023.

Django. Decorators. URL <https://book.pythontips.com/en/latest/decorators.html>. Accessed on August 01, 2023.

Django Documentation. The model-view-template pattern, 2021. URL <https://docs.djangoproject.com/en/3.2/faq/general/#django-looks-like-an-mvc-framework-but-you-call-the-controller-the-view-and-the-view>. Accessed on August 01, 2023.

R. S. Fisher, C. Acevedo, A. Arzimanoglou, A. Bogacz, J. H. Cross, C. E. Elger, J. Engel, L. Forsgren, J. A. French, M. Glynn, D. C. Hesdorffer, B. I. Lee, G. W. Mathern, S. L. Moshé, E. Perucca, I. E. Scheffer, T. Tomson, and M. Watanabe. Ilae official report: A practical clinical definition of epilepsy. *Epilepsia*, 55(4): 475–482, 2014.

- Ary L. Goldberger, Luís A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger Mark, ..., and H. Eugene Stanley. Physiobank, physiokit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000. URL <https://doi.org/10.1161/01.cir.101.23.e215>. Online.
- J Gotman. Automatic recognition of epileptic seizures in the eeg. *Electroencephalography and Clinical Neurophysiology*, 54(5):530–540, 1982.
- Matthias Ihle, Hinnerk Feldwisch-Drentrup, César A. Teixeira, Adrien Witon, Björn Schelter, Jens Timmer, and Andreas Schulze-Bonhage. Epilepsiae – a european epilepsy database. *Computer Methods and Programs in Biomedicine*, 106(3):127–138, 2012a. ISSN 0169-2607. doi: <https://doi.org/10.1016/j.cmpb.2010.08.011>. URL <https://www.sciencedirect.com/science/article/pii/S0169260710002221>.
- Matthias Ihle, Hinnerk Feldwisch-Drentrup, César A. Teixeira, Adrien Witon, Björn Schelter, Jens Timmer, and Andreas Schulze-Bonhage. Epilepsiae – a european epilepsy database. *Computer Methods and Programs in Biomedicine*, 106(3):127–138, 2012b. ISSN 0169-2607. doi: <https://doi.org/10.1016/j.cmpb.2010.08.011>. URL <https://www.sciencedirect.com/science/article/pii/S0169260710002221>.
- A. Jacoby and G. A. Baker. Quality-of-life trajectories in epilepsy: A review of the literature. *Epilepsy & Behavior*, 12(4):557–571, 2008.
- javatpoint. Mvc in jsp - javatpoint. URL <https://www.javatpoint.com/MVC-in-jsp>. Accessed on August 01, 2023.
- Javier Canales Luna. Top programming languages for data scientists in 2022, 2023. URL <https://www.datacamp.com/blog/top-programming-languages-for-data-scientists-in-2022>. Accessed on August 01, 2023.
- jQuery. jquery. URL <https://jquery.com/>. Accessed on August 01, 2023.
- Jan Klatt, Hinnerk Feldwisch-Drentrup, Michael Ihle, Vincent Navarro, Martin Neufang, Caio Teixeira, Christian Adam, Martin Valderrama, Carolina Alvarado-Rojas, Agnieszka Witon, Michel Le Van Quyen, Francisco Sales, Antonio Dourado, Jens Timmer, Andreas Schulze-Bonhage, and Björn Schelter. The epilepsiae database: An extensive electroencephalography database of epilepsy patients. *Epilepsia*, 53:1669–1676, 2012. doi: 10.1111/j.1528-1167.2012.03564.x.

- R. Kobau, H. S. Zahran, D. J. Thurman, and M. M. Zack. Epilepsy surveillance among adults—19 states, behavioral risk factor surveillance system, 2005. *MMWR. Surveillance Summaries*, 64(10):1–35, 2015.
- Cody Lindley. What is a front-end developer? · front-end developer handbook 2018. URL <https://frontendmasters.com/guides/front-end-handbook/2018/what-is-a-FD.html>. Accessed on August 01, 2023.
- MDN Web Docs. Css (cascading style sheets), 2021a. URL <https://developer.mozilla.org/en-US/docs/Web/CSS>. Accessed on August 01, 2023.
- MDN Web Docs. Html (hypertext markup language), 2021b. URL <https://developer.mozilla.org/en-US/docs/Web/HTML>. Accessed on August 01, 2023.
- MDN Web Docs. Javascript, 2021c. URL <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. Accessed on August 01, 2023.
- Florian Mormann, Ralph G. Andrzejak, Christian E. Elger, and Klaus Lehnertz. Seizure prediction: The long and winding road. *Brain*, 130(Pt 2):314–33, 2007.
- Nick Badich. Sketch, wireframe, mockup, and prototype: Why, when and how, 2020. URL <https://uxplanet.org/sketch-wireframe-mockup-and-prototype-why-when-and-how-29a25b3157c4>. Accessed on August 01, 2023.
- Ernst Niedermeyer and Fernando Lopes da Silva. *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. Lippincott Williams & Wilkins, 5th edition, 2005.
- ONDA - Online Database Architect. Onda - online database architect, 2022. URL <http://onda.dei.uc.pt/v4/>. Accessed on August 01, 2023.
- Oracle Corporation. Class processbuilder - java platform se 7. URL <https://docs.oracle.com/javase/7/docs/api/java/lang/ProcessBuilder.html>. Accessed on August 01, 2023.
- Andrew Pinkham. *Django Unleashed*. Sams, Indianapolis, IN, 2016.
- PlotlyInc. Plotly javascript graphing library. URL <https://plotly.com/javascript/>. Accessed on August 01, 2023.
- Sriram Ramgopal et al. Seizure detection, seizure prediction, and closed-loop warning systems in epilepsy. *Epilepsy & Behavior*, 37:291–307, 2014.
- Max Rehkopf. User stories, 2022. URL <https://www.atlassian.com/agile/project-management/user-stories>. Accessed on August 01, 2023.

- Zhe Ren, Xiong Han, and Bin Wang. The performance evaluation of the state-of-the-art eeg-based seizure prediction models. *Frontiers in Neurology*, 13, 2022.
- Rice University. Programming languages for data science. URL <https://csweb.rice.edu/academics/graduate-programs/online-mds/blog/programming-languages-for-data-science>. Accessed on August 01, 2023.
- Sara Bastian. Learning django rest framework and mvt architecture, 2021. URL <https://medium.com/geekculture/learning-django-rest-framework-and-mvt-architecture-1ca173163f1c>. Accessed on August 01, 2023.
- Mahsa Sazgar and Martin G. Young. Overview of eeg, electrode placement, and montages. pages 117–125, 2019.
- StateOfJs. State of javascript 2022: Front-end frameworks. URL <https://2022.stateofjs.com/en-US/libraries/front-end-frameworks/>. Accessed on August 01, 2023.
- Statistics & Data. Most popular backend frameworks: 2012-2023, 2023. URL <https://statisticsanddata.org/data/most-popular-backend-frameworks-2012-2023/>. Accessed on August 01, 2023.
- Tailwind CSS. Tailwind css. URL <https://tailwindcss.com/>. Accessed on August 01, 2023.
- C.A. Teixeira, B. Direito, H. Feldwisch-Drentrup, M. Valderrama, R.P. Costa, C. Alvarado-Rojas, S. Nikolopoulos, M. Le Van Quyen, J. Timmer, B. Schelter, and A. Dourado. Epilab: A software package for studies on the prediction of epileptic seizures. *Journal of Neuroscience Methods*, 200(2):257–271, 2011. ISSN 0165-0270. doi: <https://doi.org/10.1016/j.jneumeth.2011.07.002>. URL <https://www.sciencedirect.com/science/article/pii/S0165027011003888>.
- The Apache Software Foundation. Apache maven. URL <https://maven.apache.org/>. Accessed on August 01, 2023.
- TheMathWorksInc. Matlab version r2023a, 2022. URL <https://www.mathworks.com>.
- TheMathWorksInc. App building, 2023. URL <https://www.mathworks.com/help/matlab/gui-development.html>.
- TIOBE. Tiobe index, 2023. URL <https://www.tiobe.com/tiobe-index/>. Accessed on August 01, 2023.

W3C. Css - cascading style sheets, 2021a. URL <https://www.w3.org/Style/CSS/>. Accessed on August 01, 2023.

W3C. Html - hypertext markup language, 2021b. URL <https://www.w3.org/html/>. Accessed on August 01, 2023.

W3Schools. Javascript tutorial, 2021. URL <https://www.w3schools.com/js/>. Accessed on August 01, 2023.

World Health Organization. Epilepsy, 2022. URL <https://www.who.int/news-room/fact-sheets/detail/epilepsy>. Accessed on August 11, 2023.

Appendices

Appendix A

List of User Stories

A.1 User Stories

A.1.1 Epic Story 1: Authentication

As a user, i want to sign in to EPIWEB, to have access to its functionalities.

US-01: Landing Page

As an non-authenticated user, I want to see a welcoming and informative home page to provide me an overview of the platform's capabilities and features and the possibility to sign in.

US-02: Login

As an non-authenticated user of EPIWEB, I want to be able to log in to the platform securely so that I can access and use the platform's features. I want to enter the fields like username and password so that my identity is validated when accessing the platform.

US-03: Register

As an non-authenticated user, I want to recover my password by providing my registered email address. Upon submitting the email, I should receive a password reset email containing a secure link. Clicking the link will allow me to set a new password, ensuring that I can access my account. For that i need to introduce

personal information like my email.

US-04: Create an Account

As an non-authenticated user, **I want** to create an EPIWEB account by providing my personal information and selecting a username and password. This will enable me **to** access the platform's features and contribute to epilepsy research. For that i need to introduce personal information like email, username and password.

US-05: Log out

As an authenticated user, **I want** the ability to log out of my EPIWEB account when I'm done using the platform **to** ensure the security of my account and data.

A.1.2 Epic Story 2: Enhacing User Management

As an authenticated user, I want to stay engaged and informed while using the platform. I want to receive notifications about updates, track recent activity, and have an interactive user experience to enhance my overall engagement and satisfaction.

US-06: User Personal Home Page

As an authenticated user, **I want** to have a personal home page **where** I can access important information, view my saved EEG data, and access tools for analysis and prediction.

US-07: Studies Processing Queue Information

As an authenticated user, **I want** to have access to information about my studies processing queue **to** monitor the progress of EEG signal processing and ensure efficient usage of the system.

US-08: Receive Notifications for Study Updates

As a registered user of Epiweb, **I want** to receive notifications whenever there are updates or changes related to the studies I am following or contributing **to** be always informed and can stay engaged with the platform.

US-09: Recent Studies Table

As an authenticated user, **I want** to view a table displaying information about my recent studies **to** access and review the results of my EEG signal processing.

A.1.3 Epic Story 3: Raw Data Management

As an authenticated user, I want efficient tools to manage and work with raw EEG data. I want to be able to upload, store, view, and analyze EEG data so that I can perform research effectively.

US-10: Access to Raw Data

As an authenticated user, **I want** the ability to access and view the raw EEG data from my patients' studies **to** perform in-depth analysis or further processing if needed.

US-11: Load Raw Data

As an authenticated user, **I want** to be able to upload and load raw EEG data files into the system **for** further processing and analysis.

US-12: Manage a Specific Raw Data

As an authenticated user, **I want** to be able to manage and perform actions on a specific raw EEG data file that I have uploaded into the system.

US-13: Visualize Graphically Raw Data

As an authenticated user, **I want** to be able to visualize the raw EEG data graphically **to** gain insights into the patient's brain activity.

US-14: Filter Channels on Graphic Visualization

As an authenticated user, **I want** to have the ability to filter and display specific EEG channels on the graphical visualization of raw EEG data **to** focus on relevant data for analysis.

US-15: Analyze Average Plot

As an authenticated user, **I want** the ability to generate and analyze average plots from EEG data **to** identify patterns and trends.

US-16: Analyze Subtraction Plots

As an authenticated user, **I want** the ability to create subtraction plots by subtracting one EEG channel from another **for** comparative analysis.

US-17: Change Window Size of Visualization

As an authenticated user, **I want** the flexibility to adjust the window size of graphical EEG data visualization **to** focus on specific time intervals.

US-18: Jump to Specific Time

As an authenticated user, **I want** the ability to quickly jump to a specific time point within the EEG data **for** detailed analysis.

US-19: Jump to Specific Event

As an authenticated user, **I want** to easily navigate to specific events, such as seizures, within the EEG data **for** in-depth analysis.

US-20: Real-Time Filtering

As an authenticated user, **I want** the ability to apply real-time filtering to EEG data **to** enhance data visualization and analysis.

US-21: Share Raw Data

As an authenticated user, **I want** to be able to share specific raw EEG data files with other users or collaborators **for** discussion and analysis.

US-22: Comment on Raw Data

As an authenticated user, **I want** to be able to add comments and annotations to specific points within raw EEG data files **for** collaborative analysis and discussion.

A.1.4 Epic Story 4: Studies Management

As an authenticated user, I want a system for managing my studies. I want to create, edit, view, filter, and analyze different types of studies (feature extraction, feature selection/dimensionality reduction, classification) to advance my research goals efficiently.

US-23: Access to Studies

As an authenticated user, **I want** to have easy access to my studies and associated EEG data **for** analysis and review.

US-24: Manage a Specific Study

As an authenticated user, **I want** to have the ability to manage and edit the details of a specific study that I have created.

US-25: Comment on a Study

As an authenticated user, **I want** to be able to leave comments on specific studies to provide feedback, notes, or collaborate with other users.

US-26: Share a Study

As an authenticated user, **I want** to be able to share a specific study with other registered users, allowing **for** collaboration and information sharing.

US-27: Create a Feature Extraction Study

As an authenticated user, **I want** to create a Feature Extraction study to analyze EEG data and extract relevant features **for** further analysis.

US-28: Visualize Graphically in a Feature Extraction

As an authenticated user working on a Feature Extraction study, **I want** to visualize EEG data graphically **to** understand the raw signals and extracted features.

US-29: Filter Graphs by Features in a Feature Extraction

As an authenticated user conducting a Feature Extraction study, **I want** to filter the displayed graphs based on specific extracted features.

US-30: Change Window Size in a Feature Extraction Study

As an authenticated user working on a Feature Extraction study, **I want** to adjust the window size for data visualization **to** focus on specific time intervals.

US-31: Jump to Specific Time in a Feature Extraction Study

As an authenticated user conducting a Feature Extraction study, **I want** the ability to quickly jump to a specific time point in the EEG data **for** detailed analysis.

US-32: Jump to Specific Event in a Feature Extraction Study

As an authenticated user conducting a Feature Extraction study, **I want** the capability to jump to specific events, such as seizures, in the EEG data **for** detailed analysis.

US-33: Edit a Feature Extraction Study

As an authenticated user, **I want** to have the ability to edit the details and settings of a Feature Extraction study after its initial creation.

US-34: Create a Feature Selection/Dimensionality Reduction Study

As an authenticated user, **I want** to create a Feature Selection/Dimensionality Reduction study **to** analyze and reduce the dimensionality of EEG data for more efficient processing and analysis.

US-35: Visualize Graphically in a Feature Selection/Dimensionality Reduction Study

As an authenticated user working on a Feature Selection/Dimensionality Reduction study, **I want** to visualize EEG data graphically **to** understand the raw signals and analyze dimensionality-reduced data.

US-36: Filter Graphs by Features in a Feature Selection/Dimensionality Reduction Study

As an authenticated user conducting a Feature Selection/Dimensionality Reduction study, **I want** to filter the displayed graphs based on specific features after dimensionality reduction.

US-37: Change Window Size in a Feature Selection/Dimensionality Reduction Study

As an authenticated user working on a Feature Selection/Dimensionality Reduction study, **I want** to adjust the window size for data visualization **to** focus on specific time intervals in dimensionality-reduced data.

US-38: Jump to Specific Time in a Feature Selection/Dimensionality Reduction Study

As an authenticated user working on a Feature Selection/Dimensionality Reduction study, **I want** the ability to jump to a specific time point within the dimensionality-reduced data for more precise analysis.

US-39: Jump to Specific Event (Seizures) in a Feature Selection/Dimensionality Reduction Study

As an authenticated user conducting a Feature Selection/Dimensionality Reduction study, **I want** to quickly navigate to specific events, such as seizures, within the dimensionality-reduced data **to** analyze their impact.

US-40: Edit Feature Selection/Dimensionality Reduction Study

As an authenticated user, **I want** the ability to edit the details and configuration of an existing Feature Selection/Dimensionality Reduction study.

US-41: Create a Classification Study

As an authenticated user, **I want** the capability to create a Classification study **to** train machine learning models for predicting epileptic seizures based on EEG data.

US-42: Visualize Classification Study Graphs

As an authenticated user working on a Classification study, **I want** the ability to visualize classification-related graphs and metrics **to** assess the performance of my machine learning models.

US-43: Change Window Size in a Classification Study

As an authenticated user working with a Classification study, **I want** the ability to change the window size for data analysis **to** explore different temporal aspects of the EEG data.

US-44: Jump to Specific Time in a Classification Study

As an authenticated user conducting a Classification study, **I want** the ability to navigate to a specific time point within the EEG data **to** analyze model predictions and performance.