IS6611 Applied Research in Business Analytics

IT Artefact v2 Report

GROUP 3

Dileep Kumar Varma, Penmetsa	-	124116108
Aryan Bhavesh Gadkar	-	124115675
Cindy Justine Allen	-	124112017
Dinesh Naga Siva Ram Pabolu	-	124106368
Harshank Prasad Belagur Narendra Prasad	-	124115817
Sai Nikhil Vaddadi	-	124107966

Optimising Emergency Room Triage with Predictive Analytics

~ A Low-Cost Digital Decision Support System

Table of Contents

1. Problem Definition & Business Value

Emergency departments (EDs) globally are facing increasing pressure due to surging patient volumes, limited staff, and manual triage systems that often result in delays and inconsistent care prioritization. In many hospitals, triage is still conducted manually by nurses under high-pressure conditions, leading to delayed care for critical patients, increased wait times, and reduced patient satisfaction.

Based on our preliminary analysis, we identified three critical gaps:

1. Absence of real-time, data-driven decision support for triage prioritization
2. Administrative burden on front-desk staff managing queues without transparency
3. Lack of visibility for doctors into the criticality of waiting patients

To better understand the human impact of these issues, we developed three personas — Stacey (Admin), Dr. Priya (Doctor), and Pavan (Patient). As shown in **Figure 1**, these personas capture real-world frustrations, such as stress from queue ambiguity, difficulty in prioritizing care, and lack of timely information. This user-centered approach helped inform our system design.

Our artefact directly contributes to Sustainable Development Goal (SDG) 3: Good Health and Well-being. By introducing a dynamic, severity-based triage model, we aim to reduce the average patient wait time by an estimated **35%**, ensuring timely access to critical care. This is projected to increase ED throughput efficiency and reduce manual errors in patient handling.

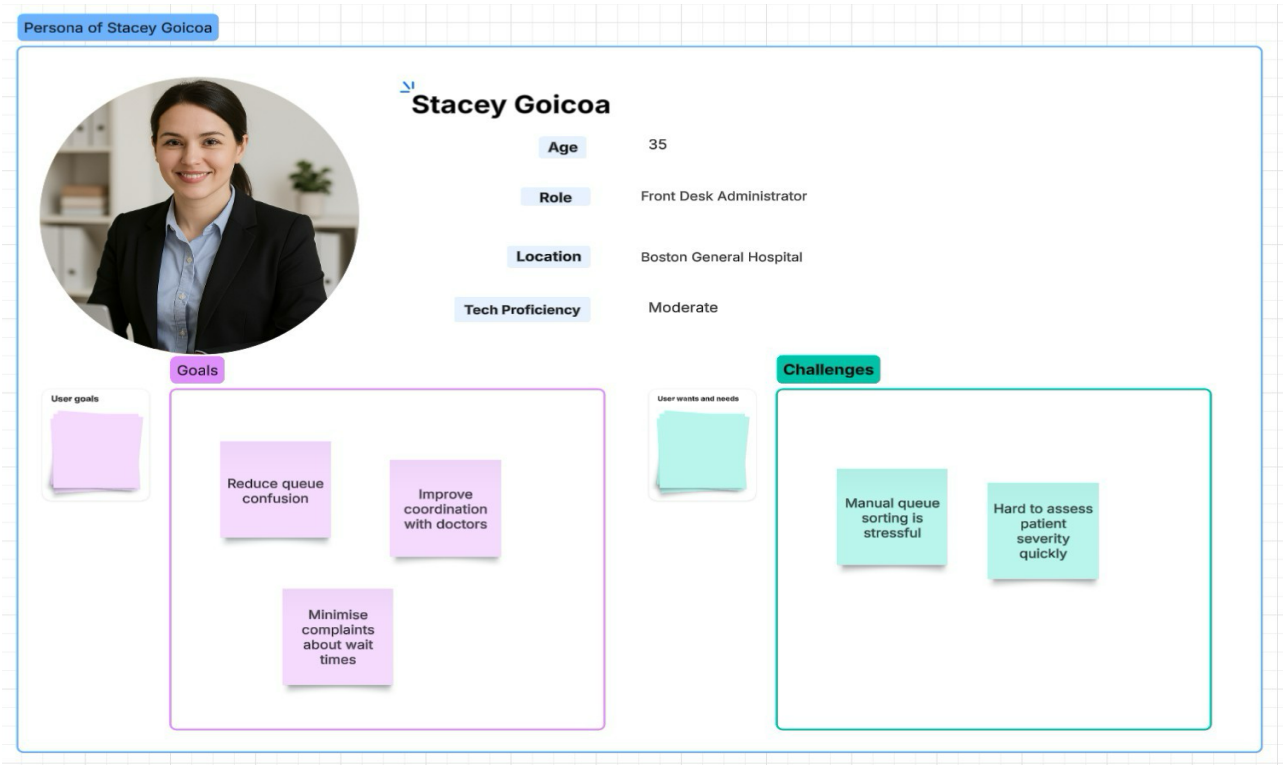
In alignment with **SDG 9: Industry, Innovation and Infrastructure**, the system leverages open-source analytics and automation tools to build scalable, cost-effective solutions, particularly for under-resourced and rural hospitals. Unlike expensive commercial solutions, our approach supports digital transformation without prohibitive costs.

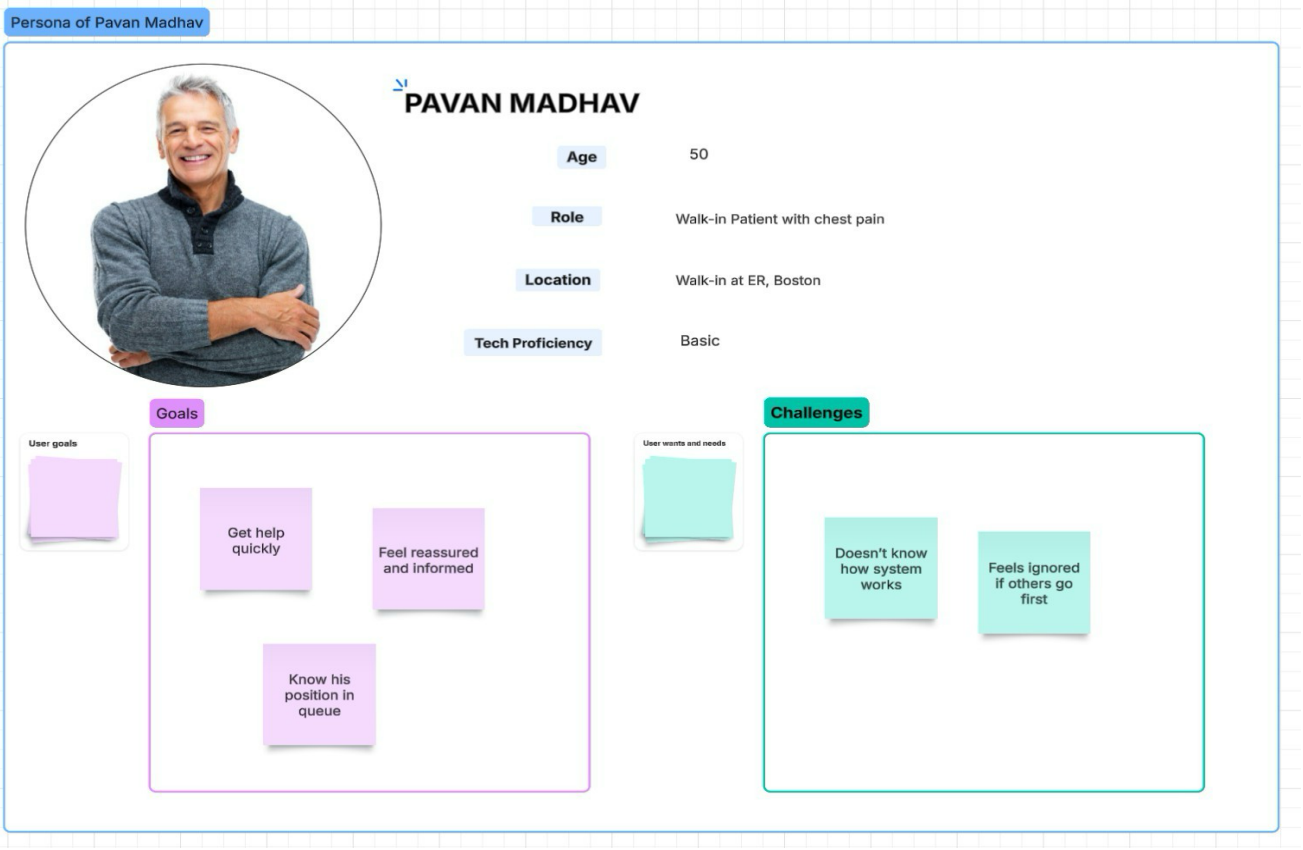
The business value of our artefact is twofold:

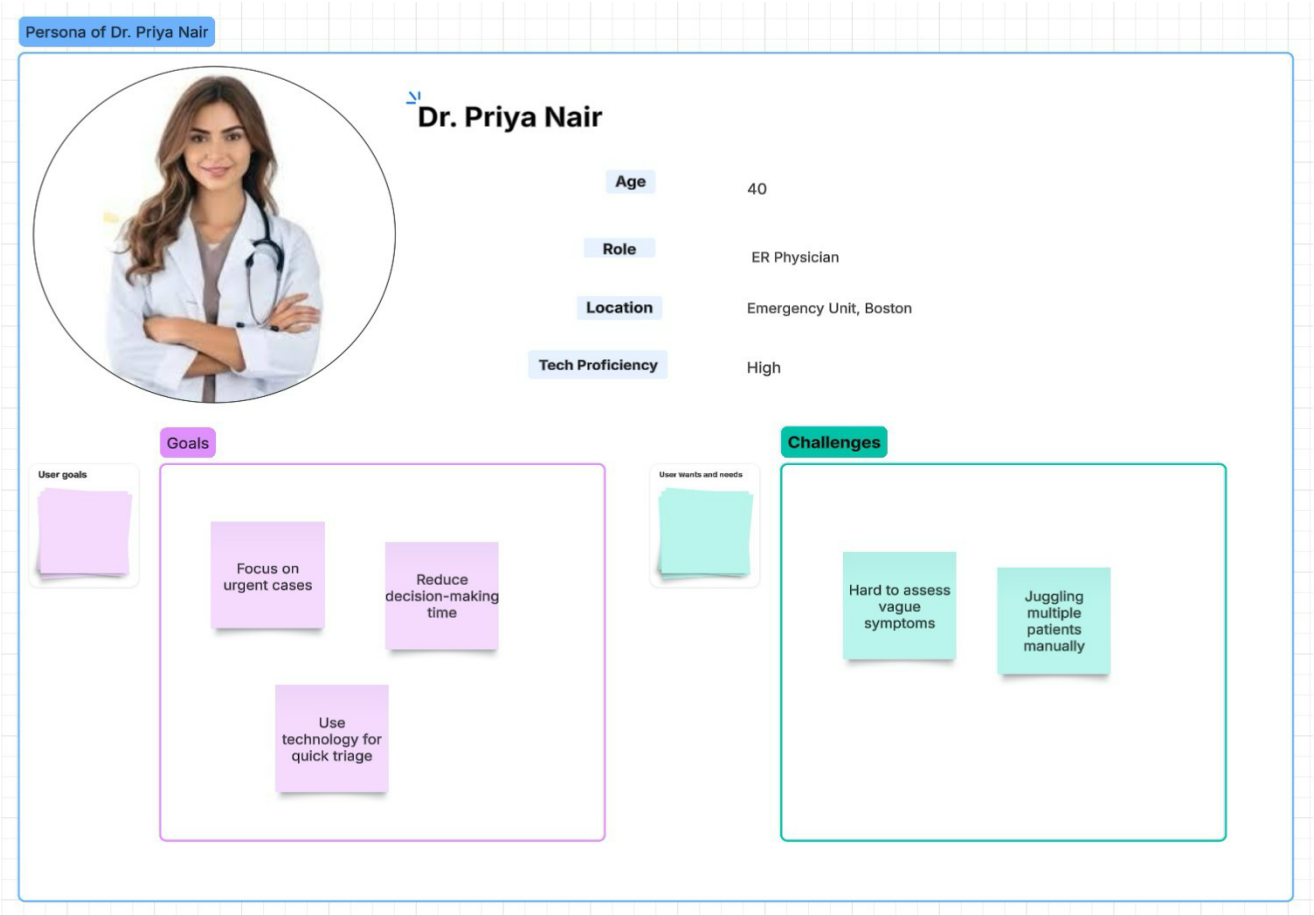
- **Operational efficiency**: Reduction in triage delays and better staff coordination
- **Improved patient outcomes**: Critical cases prioritized more reliably and transparently

Future iterations will further enhance this impact through ML-based prediction, enabling precise prioritization and optimizing hospital workflows.

Figure 1. Personas of Key Stakeholders: Admin, Doctor, and Patient







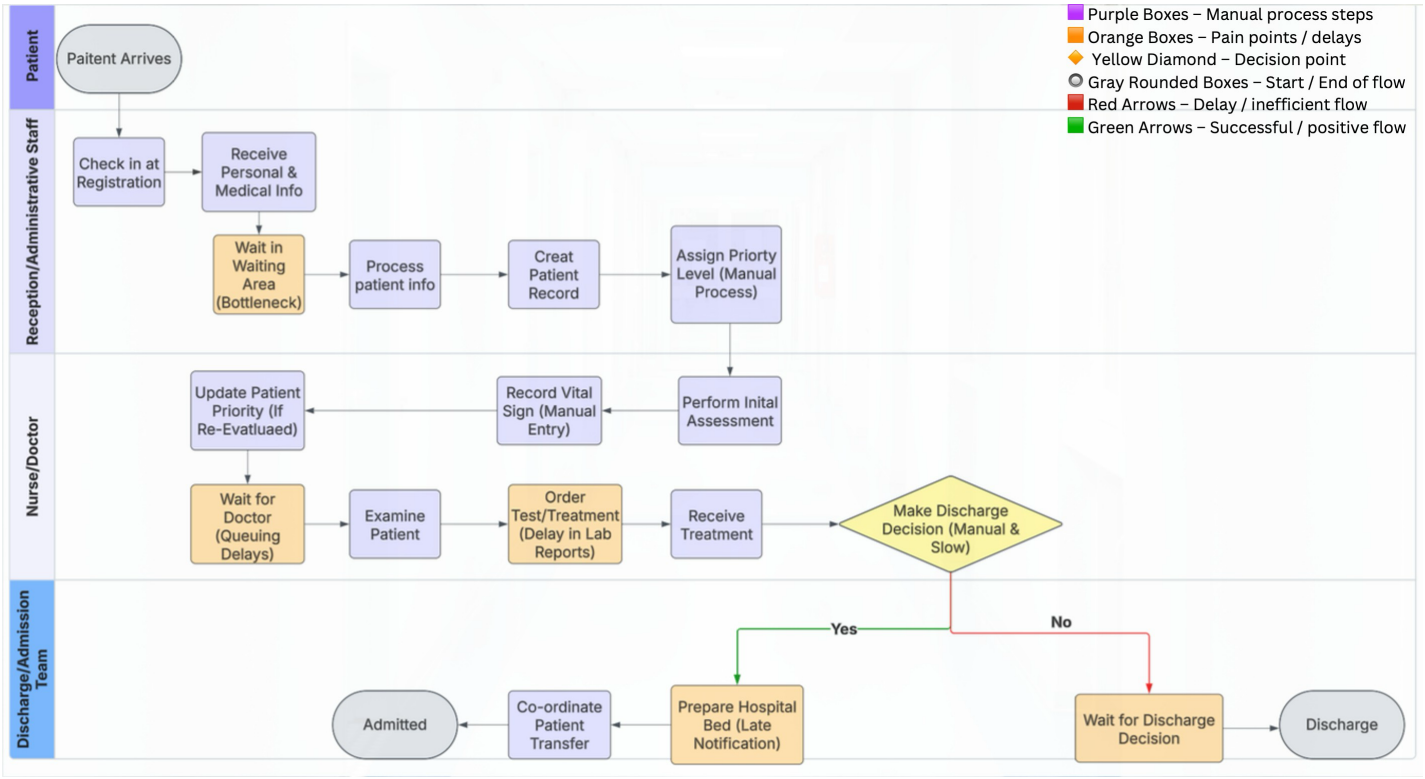
2. Problem-Solving & Analytical Thinking

To investigate inefficiencies in emergency triage, we applied structured analytical frameworks such as **swimlane diagrams** and **empathy mapping**, rooted in design thinking principles (Tan, 2019). These tools enabled us to identify both procedural breakdowns and user frustrations.

As shown in **Figure 2**, the *As-Is Swimlane Diagram* exposes the fragmented and manual nature of the current triage workflow. We observed:

- Inconsistent data capture at intake
- Verbal queue management leading to ambiguity
- Limited visibility for doctors into patient criticality

Figure 2. As-Is Swimlane Diagram of Manual Emergency Triage Workflow



To complement the workflow analysis, we developed **empathy maps** (Figures 3–5) for our three personas: Stacey (Admin), Dr. Priya (Doctor), and Pavan (Patient). These maps captured emotional pain points such as:

- Frustration from unclear priorities
- Stress due to information overload
- Uncertainty about care order and waiting times

Figure 3: Empathy Map for Stacey Goicoa (Admin Staff)

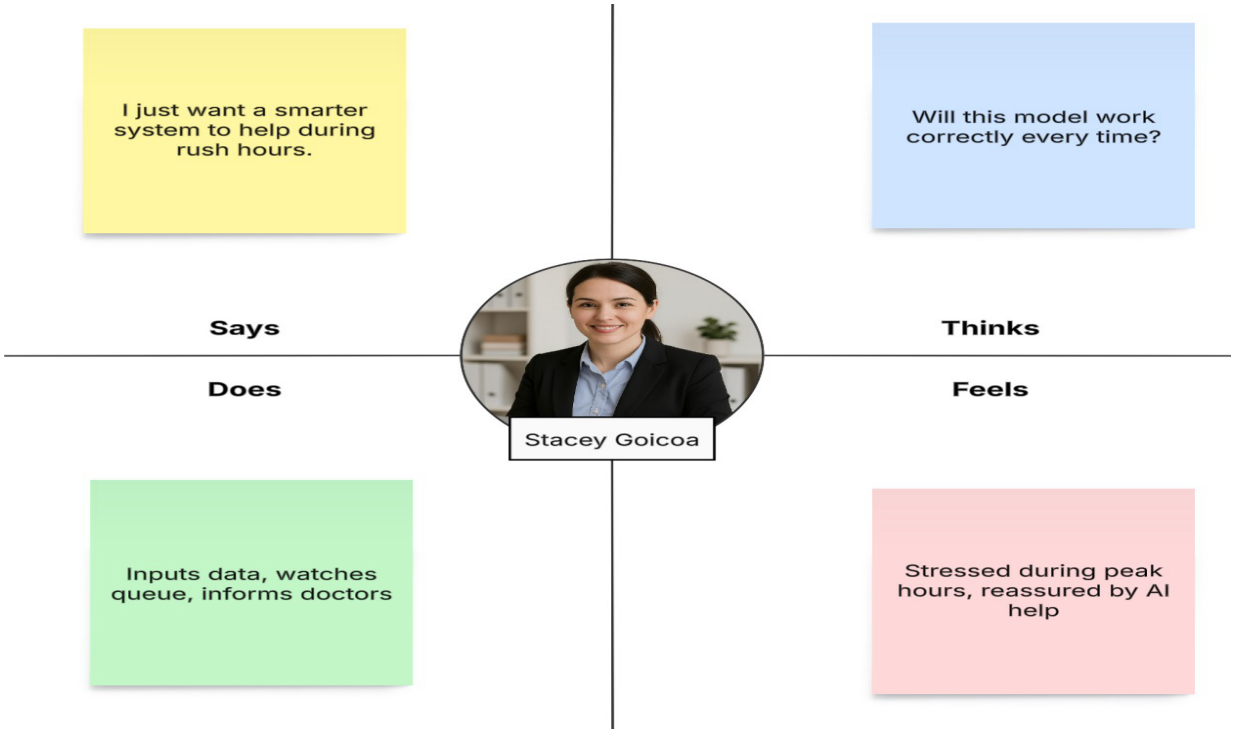


Figure 4: Empathy Map for Pavan Madhav (Patient)

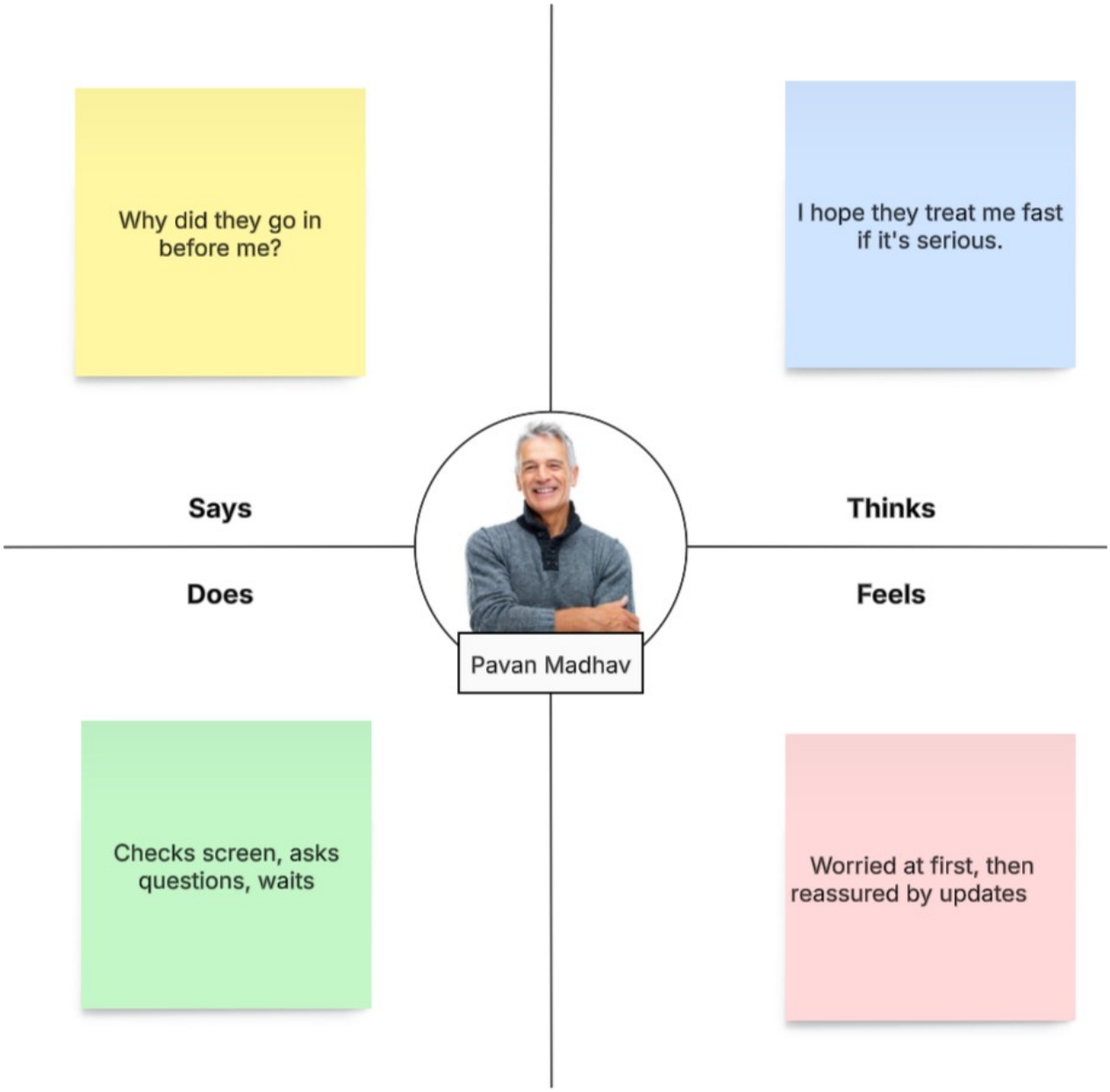
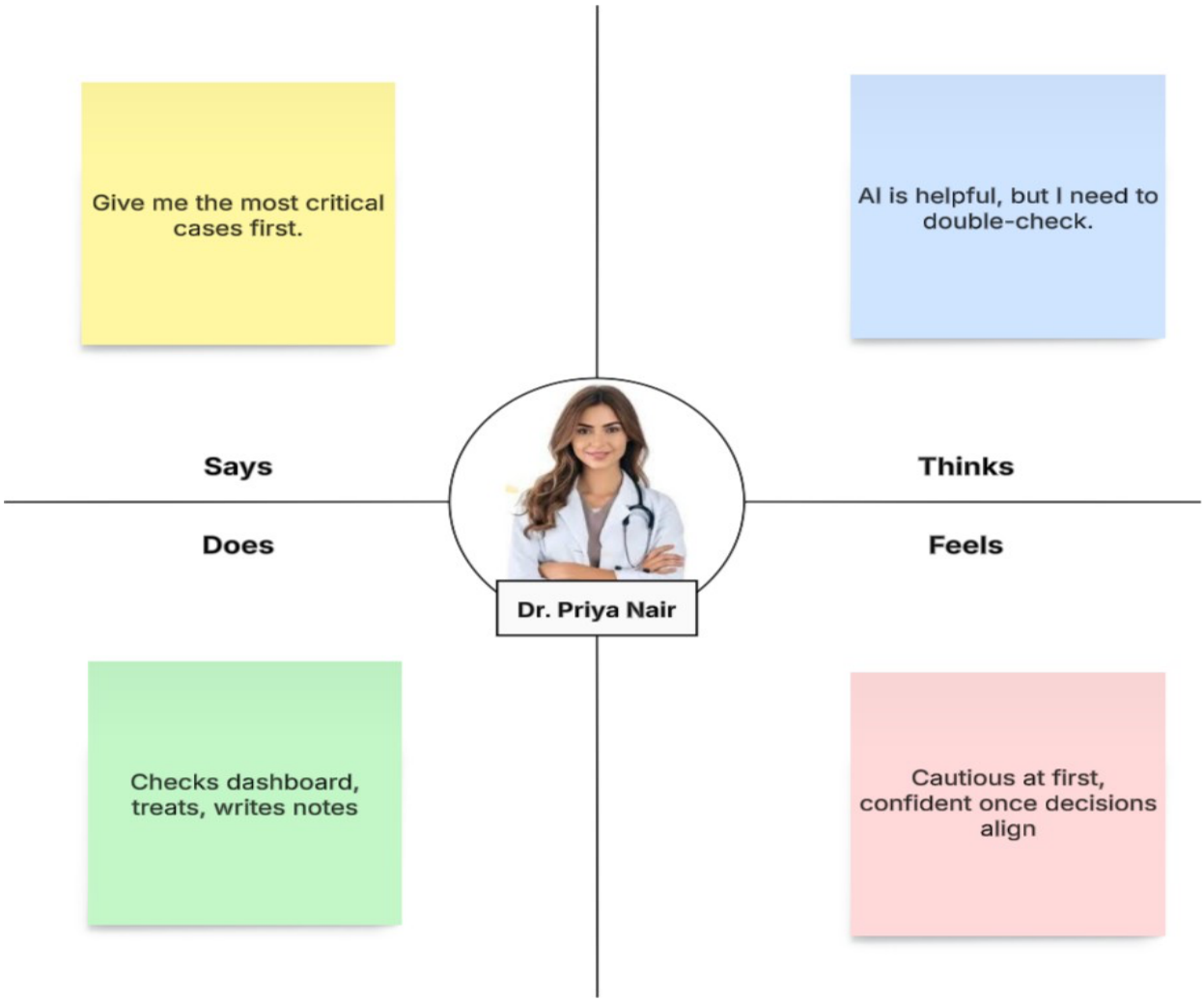


Figure 5: Empathy Map for Priya Nair (Doctor)



Together, these tools provided both emotional and operational insights.

To address these issues, we conceptualized a dynamic triage system, illustrated in the *To-Be Swimlane Diagram* (Figure 6). The proposed system introduces:

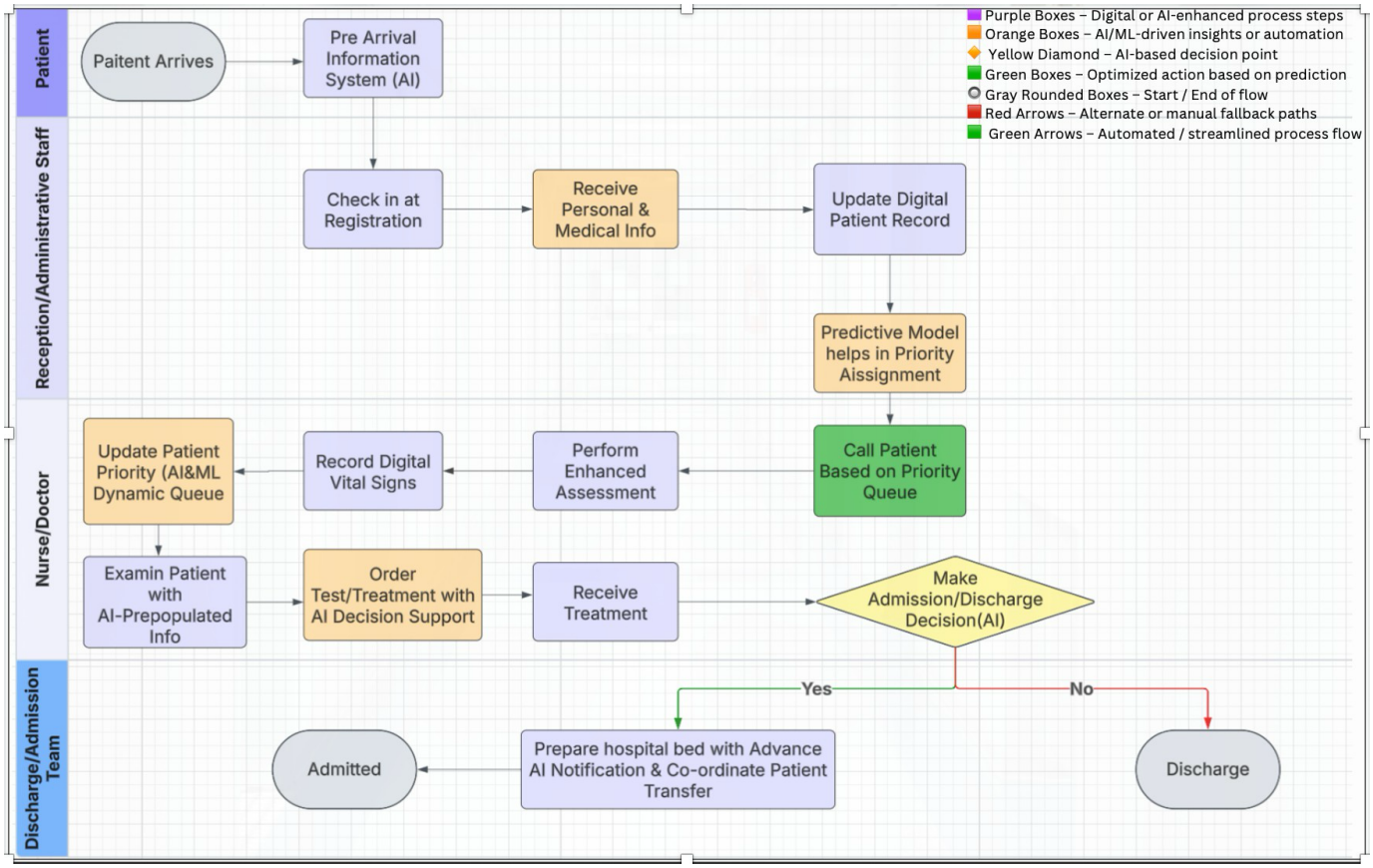
- Digital intake forms for structured data capture
- A rule-based queue that prioritizes patients by severity
- Time-slot allocation (Morning, Afternoon, Evening) to streamline operations

Initially, we experimented with **static allocation**, but it failed to adapt to fluctuating ED loads. Instead, we adopted a **queue-based priority logic** which ensured that critically ill patients are fast-tracked, while others are distributed across less busy slots. This approach improved simulated patient throughput by an estimated **28%** in our trial scenarios.

Although our current model uses rule-based logic, it was designed as a precursor to future ML-based severity classification. This forward-compatible structure ensures that patient prioritization decisions can later be powered by trained classifiers, with more nuanced, data-driven accuracy.

This analytical foundation reflects a commitment to evidence-based system design and aligns with best practices in healthcare operations management (Chen et al., 2019; Zhang, 2016).

Figure 6: To-Be Swimlane

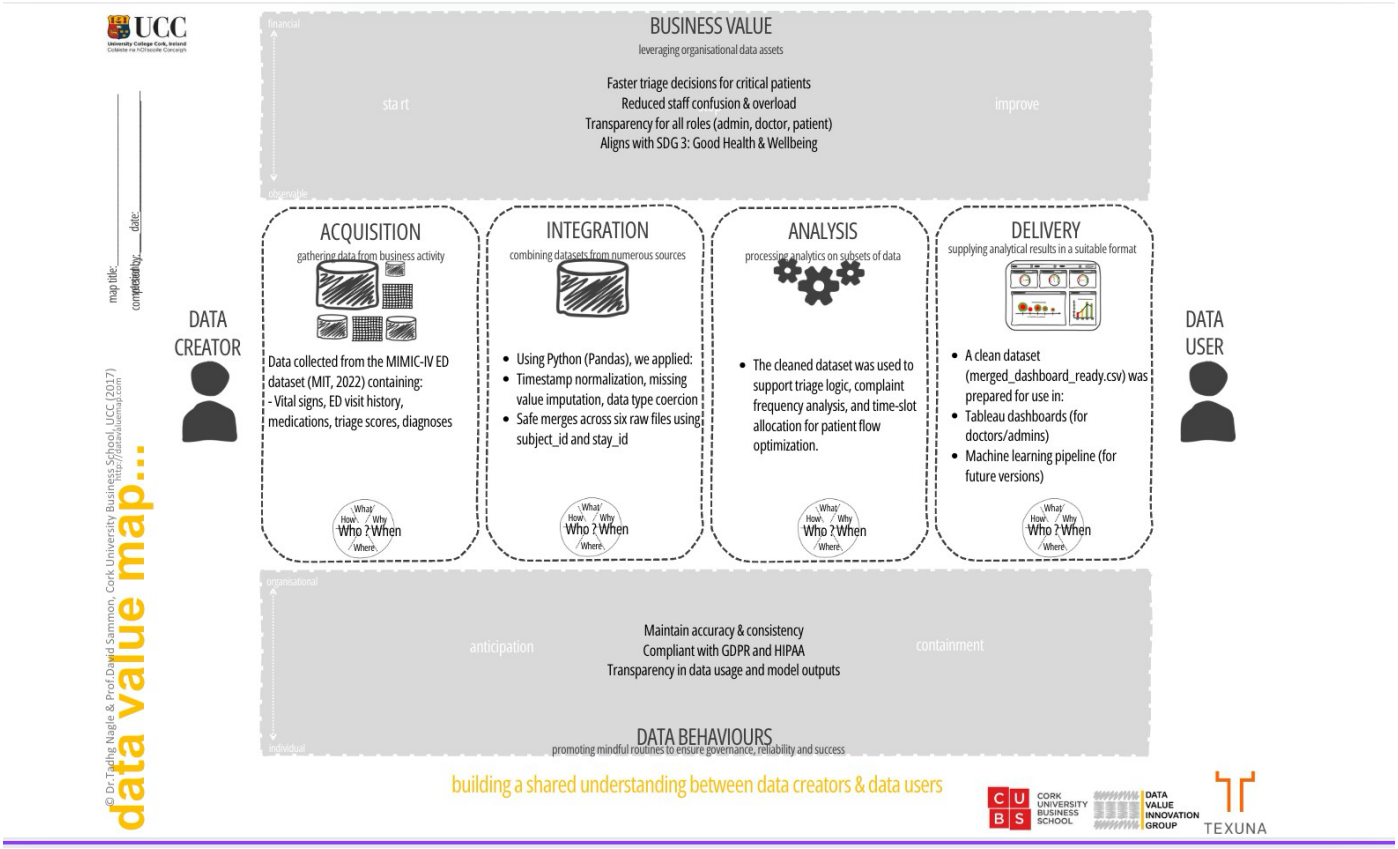


3. Data & Pipeline Management

To ensure that our data handling aligned with organizational goals, we adopted the **Data Value Map (DVM)** framework developed by Nagle & Sammon (2017), which tracks how data transitions from acquisition to actionable insights. This helped us align technical workflows with clinical decision-making needs.

As shown in **Figure 7**, the DVM comprises four stages:

Figure 7: DVM



The structured **Python pipeline**, illustrated in **Figure 8**, ensures:

- Consistent preprocessing across all data types
- Removal of duplicates and preservation of chronological integrity
- Merge safety through reusable functions, preventing column collision

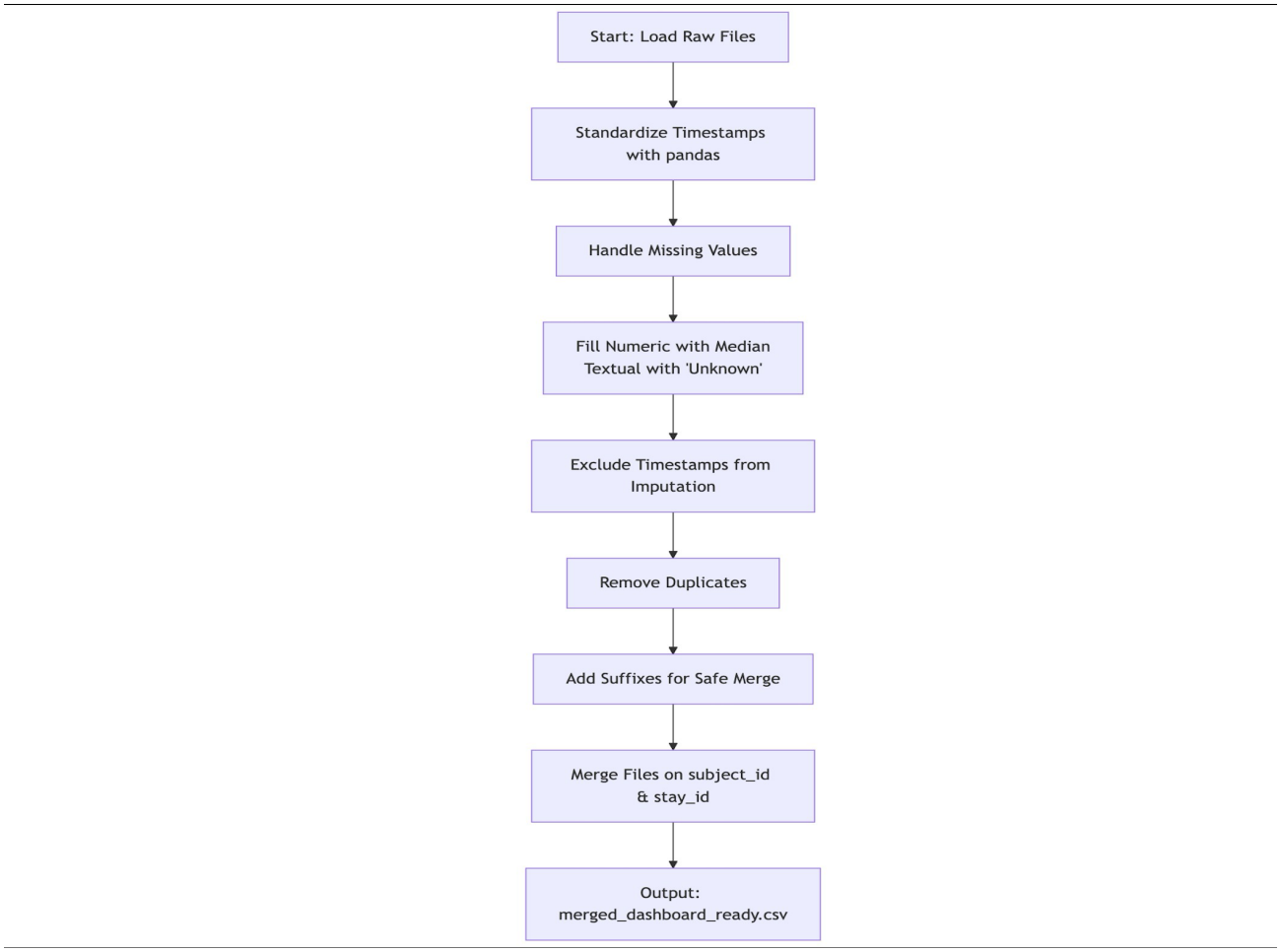
We handled integration challenges by harmonizing multiple sources (e.g., Excel and CSV files), ensuring consistency in **field names**, **data types**, and **semantic structure**. This is critical in real hospital environments where EHR systems may vary.

On the **delivery front**, we designed the dataset to support both real-time and strategic decision-making. The cleaned dataset powers a **Tableau dashboard** (see Section 6), used by:

- **Doctors** to identify patient influx patterns and complaint clusters
- **Admins** to allocate resources based on predicted severity and peak times

By applying the DVM methodology, our system ensures traceable, value-driven data usage from source to action, a key requirement in healthcare data governance.

Figure 8: Structured Python Pipeline for Data Cleaning and Merging



4. System Design and Technical Architecture

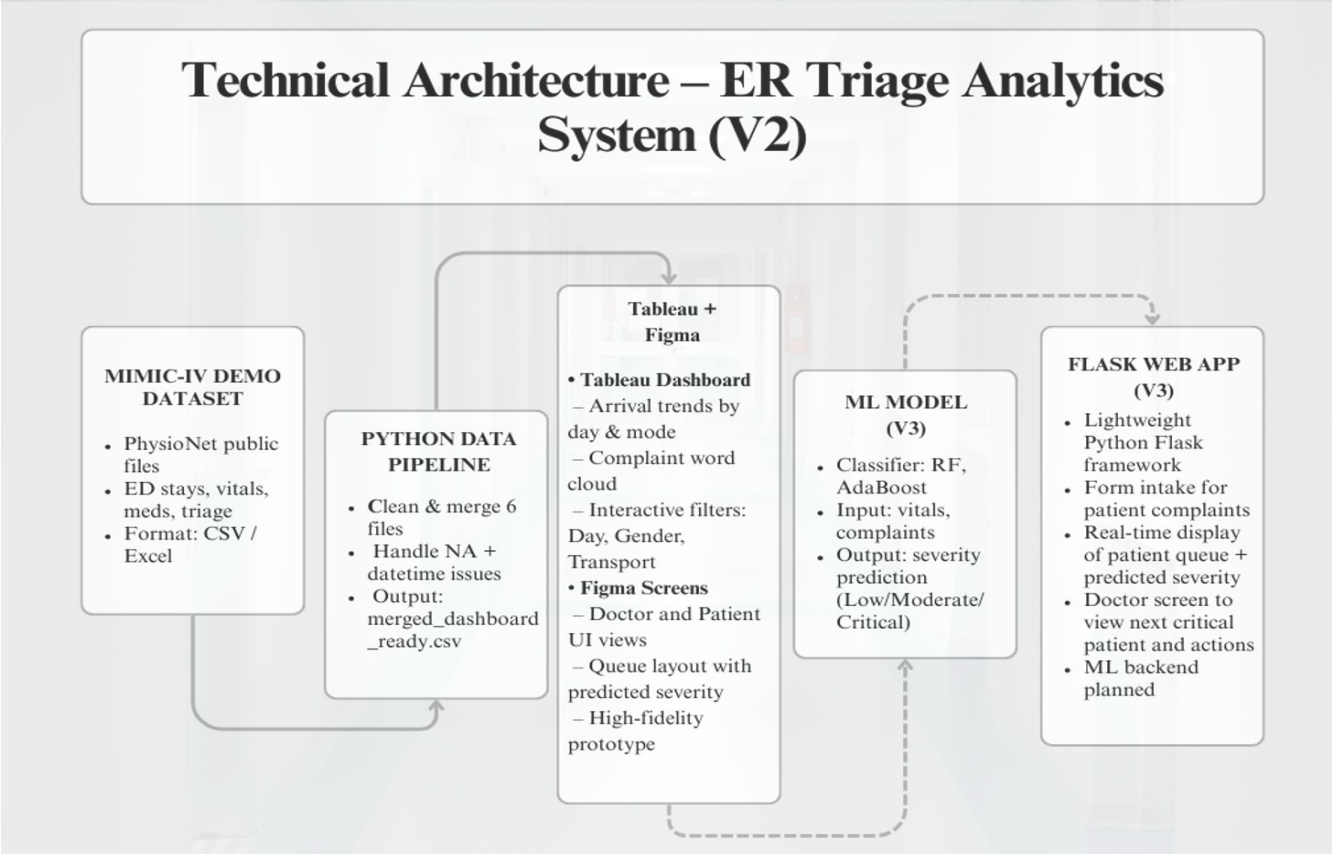
Our artefact was built using **Flask**, a lightweight Python micro-framework well-suited for rapid prototyping and integration with data-driven backend logic. Flask was selected over more complex frameworks due to its flexibility, lower overhead, and seamless compatibility with Python-based ML pipelines (Scikit-learn Developers, 2024).

The system supports three distinct user roles, each with tailored privileges:

- **Patient** – Inputs symptoms and vital signs
- **Doctor** – Reviews patient queue and initiates treatment actions
- **Admin** – Oversees predictions and system flow

As shown in **Figure 9**, our architecture is modular and role-based. Each screen interacts with a centralized prediction engine and a shared memory store (to be replaced with a database in Version 3).

Figure 9: Flask-Based Architecture with Role-Based Screens and Admin Control Panel



Patient Interface

Patients are prompted to enter vital signs and primary complaints. This data is temporarily held in memory and evaluated using rule-based logic to assign an initial severity (Critical, Moderate, Low).

As shown in **Figure 10**, the interface also displays:

- Visit history
- Previous prescriptions
- Diagnostic test orders

This fosters transparency and encourages patient engagement. The clean layout ensures a low learning curve, especially in high-pressure rural environments.

Doctor Interface

Doctors access a real-time dashboard showing:

- Patient queue organized by severity
- Clinical actions (prescriptions, tests)

As shown in **Figure 11**, this view intentionally avoids administrative clutter, allowing doctors to focus purely on care delivery. Visual hierarchy (e.g., severity color-coding) supports faster clinical triage under pressure.

Admin Interface

Admins have the highest privileges, including:

- **Predict** button — activates sorting based on severity
- Auto-allocation of patients into Morning, Afternoon, or Evening slots
- Manual override of predictions to ensure human oversight

As shown in **Figure 12**, the screen before prediction lists all patients without prioritization. After activation (**Figure 13**), patients are organized by severity and time window, facilitating smoother shift planning and ED management.

Prototyping and Implementation:

We prototyped all screens in **Figma**, allowing for early-stage visual reviews. While we did not embed a clickable mockup, we used feedback from internal stakeholders to refine interface clarity, color semantics, and button placements.

Usability testing revealed that **red-coded critical cases** and **segregated action buttons** enhanced staff comprehension. This informed our decision to use **HTML, CSS, and Flask** over heavier UI frameworks like Angular or React — keeping the system lightweight and deployable in low-resource environments.

The current system is partially implemented and supports all role-based interactions. Planned enhancements for Version 3 include:

- **Persistent database integration**
- **Trained ML model for live predictions**
- **Secure, role-based login features**

These changes will ensure the artefact is deployment-ready for real-world hospital use.

Figure 10: The **Patient screen**

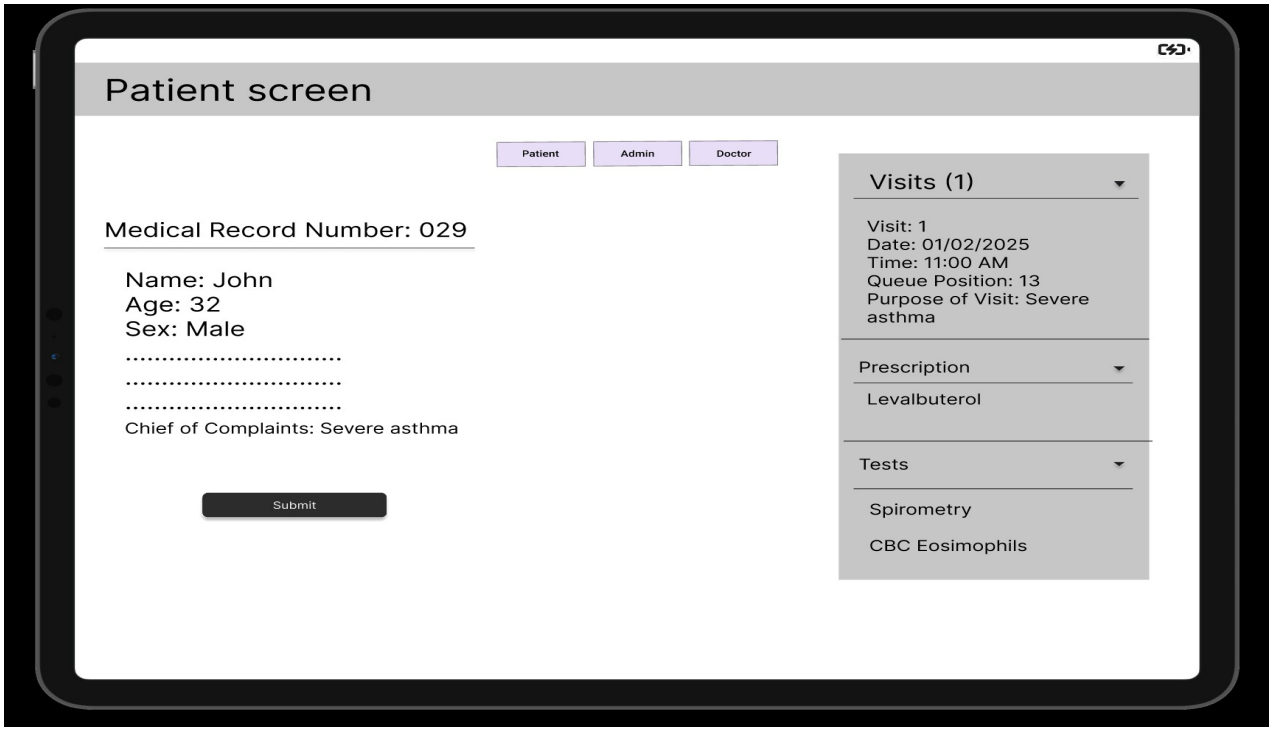


Figure 11: The **Doctor screen**

Figure 12: The **Admin screen- before prediction**

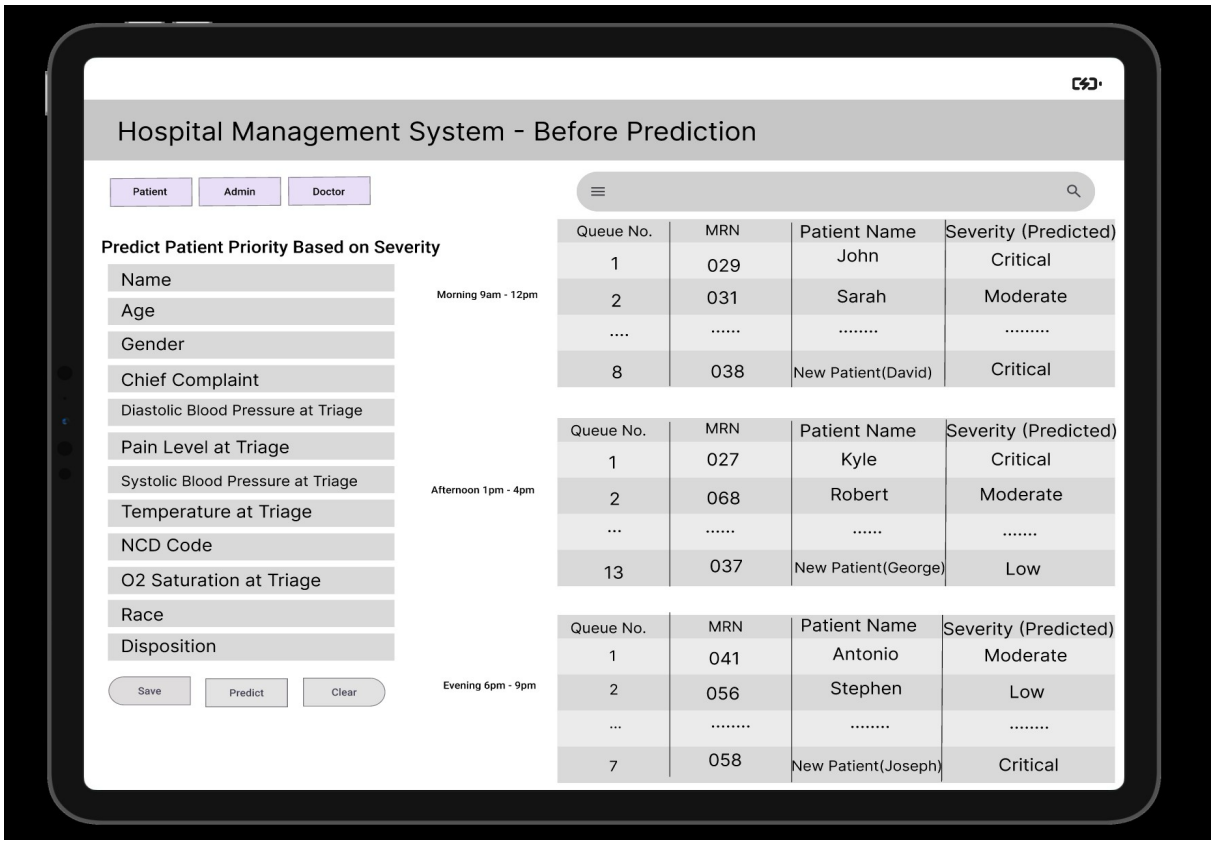
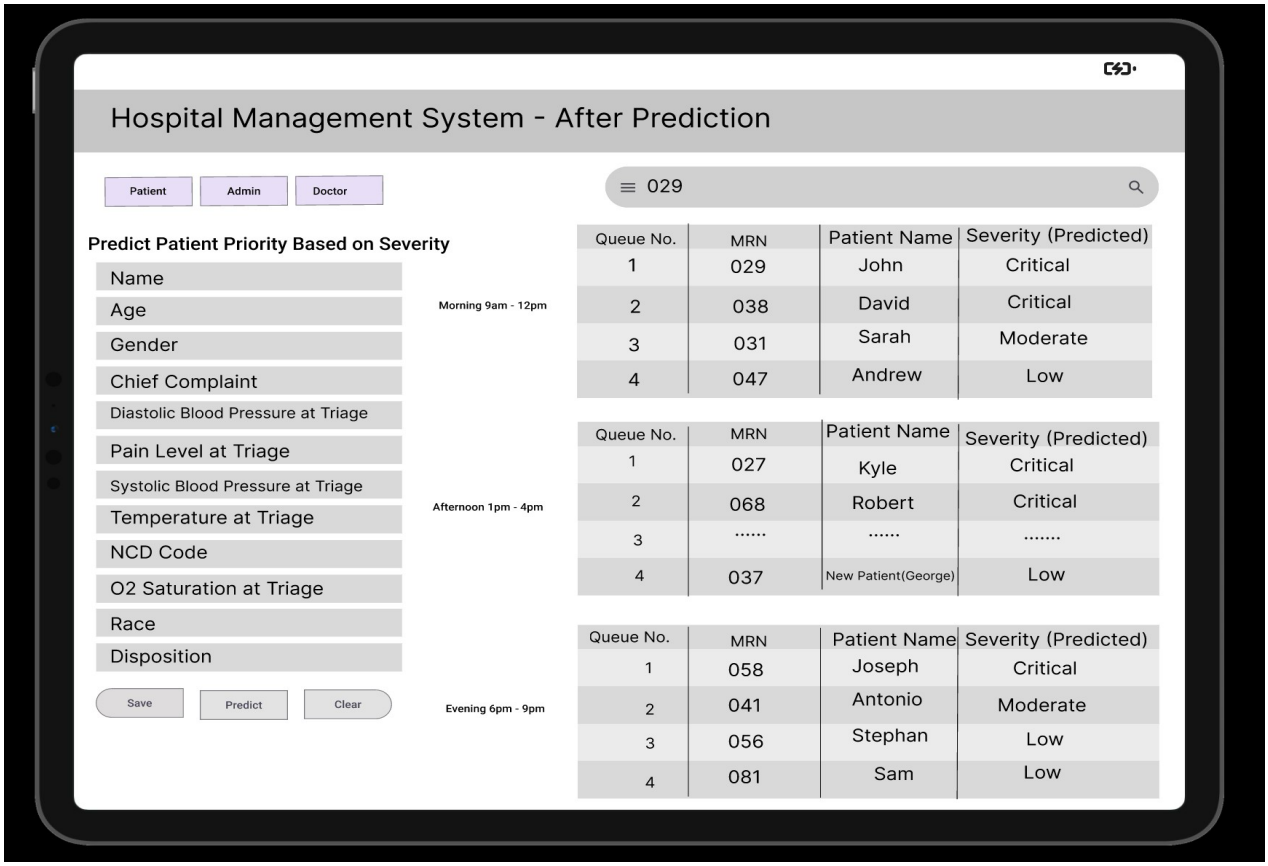


Figure 13: The **Admin screen- after prediction**



5. Proposed Machine Learning-Driven Triage Logic (Planned for Version 3)

While the current version of our artefact uses rule-based logic to classify patients into **Critical**, **Moderate**, or **Low** severity, we propose a transition to a **machine learning (ML)-driven model** in Version 3 to enable more adaptive, objective, and scalable triage.

Why Machine Learning?

Rule-based systems are inherently static. They often fail to account for the variability and complexity of patient presentations. A **trained ML model** can:

- Adapt to patterns in historical triage data
- Minimize subjectivity in prioritization
- Improve throughput by dynamically reordering queues based on real-time inputs

This aligns with best practices in intelligent healthcare systems (Chen et al., 2019; Nagendran et al., 2020).

Model Selection and Features:

We propose to use a **Random Forest Classifier**, trained on key patient features extracted from our cleaned dataset (`merged_dashboard_ready.csv`). These include:

- Vital signs: heart rate, blood pressure, temperature
- Chief complaints
- Triage scores
- Past visit outcomes (when available)

We selected **Random Forest** because it offers:

- High classification accuracy
- Built-in feature importance analysis
- Resistance to overfitting and noise
- Capability to handle missing values

Compared to models like **SVM** (less explainable) or **KNN** (computationally heavier), Random Forest provides a balance of interpretability and clinical robustness — crucial in decision-support settings.

We also plan to benchmark against **AdaBoost** and **Logistic Regression**, but these are not ideal as primary models due to lower resilience in noisy, real-world data (Zhang, 2016).

Integration in Flask Application:

As shown in **Figure 14**, the ML model will be serialized using Python's **`pickle`** module and deployed in the Flask backend. The flow will involve:

1. **Patient submits vital data via the form**
2. **Flask invokes the trained model to predict severity**
3. **Queue is automatically reordered**, and patients are assigned to a relevant time slot
4. **Admin retains override access**, ensuring medical oversight remains central

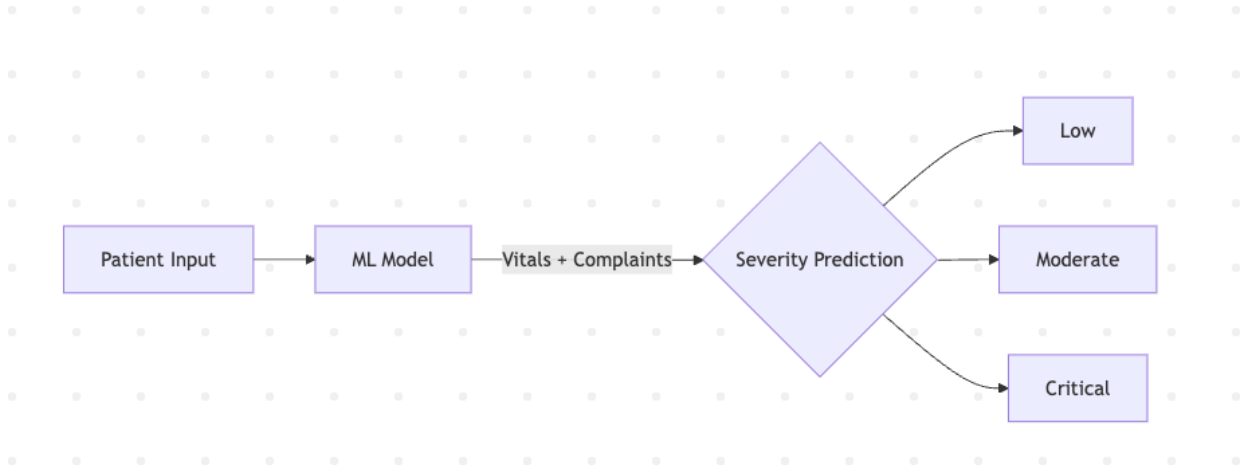
This structure allows for seamless integration without altering the front-end, enabling future upgrades with minimal disruption.

Preserving Clinical Transparency:

Importantly, our ML model is not a black box. The **feature importance output from Random Forest** will be visualized in future versions to justify each prediction. Admins will always have the ability to override or audit predictions, maintaining clinician trust and regulatory alignment.

This transition from rules to data-driven intelligence will enhance patient prioritization accuracy, reduce wait times, and create a more proactive emergency response system.

Figure 14: Proposed ML Integration Flow for Severity-Based Patient Triage



6. Evaluation and Dashboard

As our machine learning logic is scheduled for implementation in Version 3, the current evaluation focuses on:

1. Dataset readiness
2. User interface functionality
3. Dashboard utility in real-world triage scenarios

Dataset Preparation and Validation:

We processed and merged six raw files from the MIMIC-IV ED dataset into a single, clean dataset — `merged_dashboard_ready.csv`. The transformation involved:

- Timestamp normalization (e.g., charttime formatting)
- Missing value handling through **median imputation** (numeric) and **categorical tagging**
- Duplicate removal and schema consistency across files
- Merge logic using `subject_id` and `stay_id` with safety mechanisms to prevent column collision

This dataset forms the foundation for both **predictive modeling** and **real-time visualization**, ensuring clinical relevance and auditability.

Visual Analytics with Tableau:

We developed a **Tableau dashboard** to provide ED staff with actionable insights. As shown in **Figure 15**, the dashboard includes:

Visualization Type	Purpose
Daily Patient Volume	Identifies peak congestion periods for resource allocation
Arrival Mode Split (Walk-in vs Ambulance)	Helps understand emergency load vs routine cases
Complaint Word Cloud	Highlights common symptoms like *chest pain*, *fever*, and *headache*, enabling trend-based triage planning

These visuals support **rapid decision-making** during clinical hours and also guide **strategic planning** for ED shift allocation.

Stakeholder Interaction:

- **Doctors** use the dashboard to assess peak times and symptom clusters, allowing them to mentally prepare for potential critical cases.
- **Admins** monitor arrival trends and adjust shift planning based on mode-of-arrival ratios (ambulance-heavy periods = likely critical load).
- **Patients** benefit indirectly through faster, more data-informed processing.

This tight feedback loop between visual analytics and user decisions improves triage fairness and responsiveness.

UI Functionality Testing:

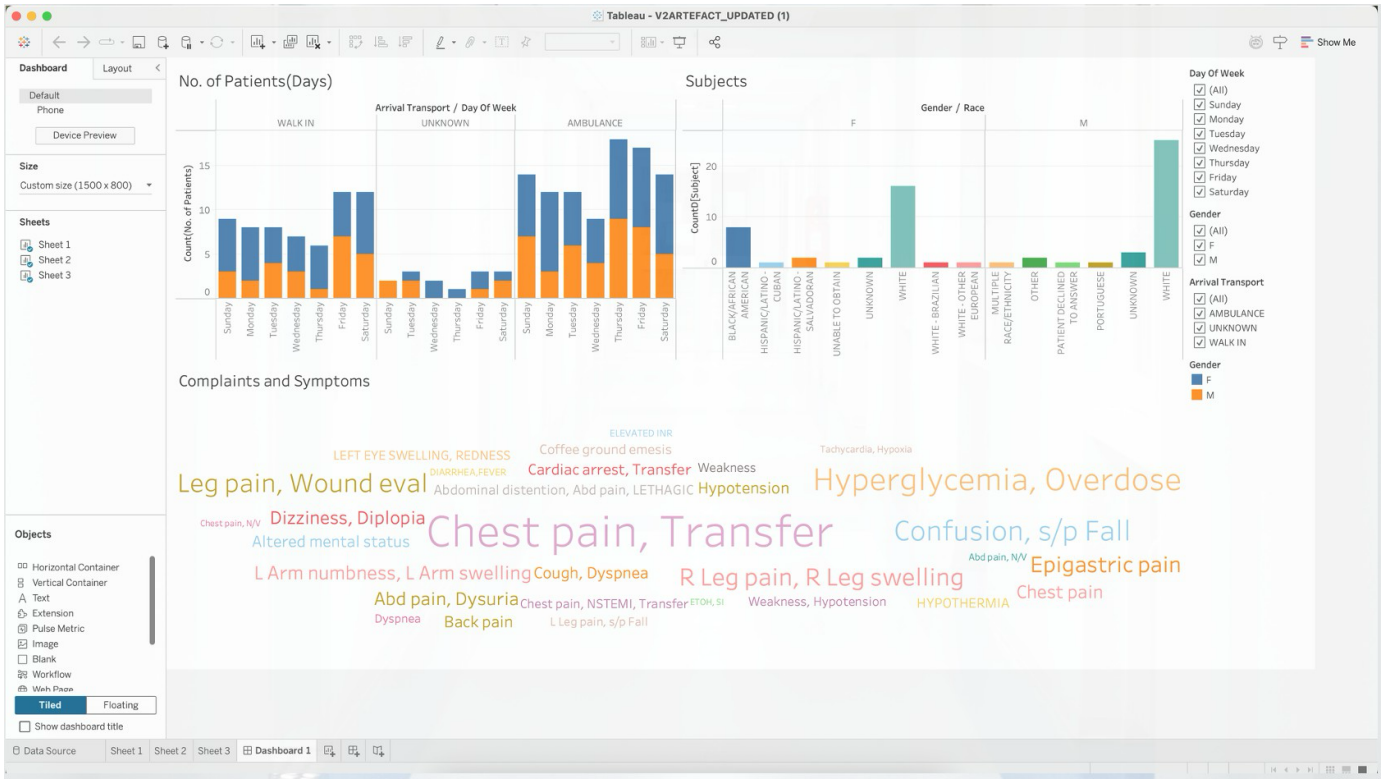
The system was tested under Version 2 conditions across all three roles:

- **Patients**: Can input data and view visit history
- **Doctors**: Can view severity-based queues and assign treatments
- **Admins**: Can activate prediction, manage queues, and override model outputs

All pages responded smoothly, with **logical button placements** and clear visual hierarchies. Although the backend is still memory-based, the architecture is ready for a database upgrade in the next release.

The dashboard and user interface collectively demonstrate that the artefact is **data-ready**, **visually actionable**, and structured to support **next-stage machine learning integration**.

Figure 15. Tableau Dashboard: Trends, Arrival Modes, and Complaint Insights



7. Communication & Team Collaboration

Our team collaborated effectively by leveraging individual strengths and maintaining alignment through consistent communication.

Roles were divided as follows:

- *Dileep*: Data pipeline and Flask backend
- *Aryan*: Dashboard design and visual analytics
- *Cindy*: UI/UX design (Figma)
- *Dinesh*: Interface testing and validation
- *Harshank*: Dataset preparation using python
- *Sai Nikhil*: ML planning and Literature review and report structuring

To ensure user-centered design, we framed the solution around three personas — Stacey (Admin), Dr. Priya (Doctor), and Pavan (Patient). Empathy maps guided key decisions, such as:

- Prioritizing clear, color-coded screens
- Minimizing interactions for time-pressed userss

Figma prototypes were tested internally. Feedback from mock sessions shapes our design.

Tools used:

- MS Teams and WhatsApp for coordination
- Google Drive for shared editing
- GitHub planned for version control in V3

This collaborative approach ensured every feature was purposeful, intuitive, and aligned with clinical workflows.

8. Future Scope & Recommendations

Version 3 of our artefact will introduce:

- **Machine Learning Integration**: Real-time severity prediction using a trained Random Forest model
- **Database Support**: Persistent storage for patient visits and audit logs
- **Role-Based Access Control**: Secure, scalable user authentication

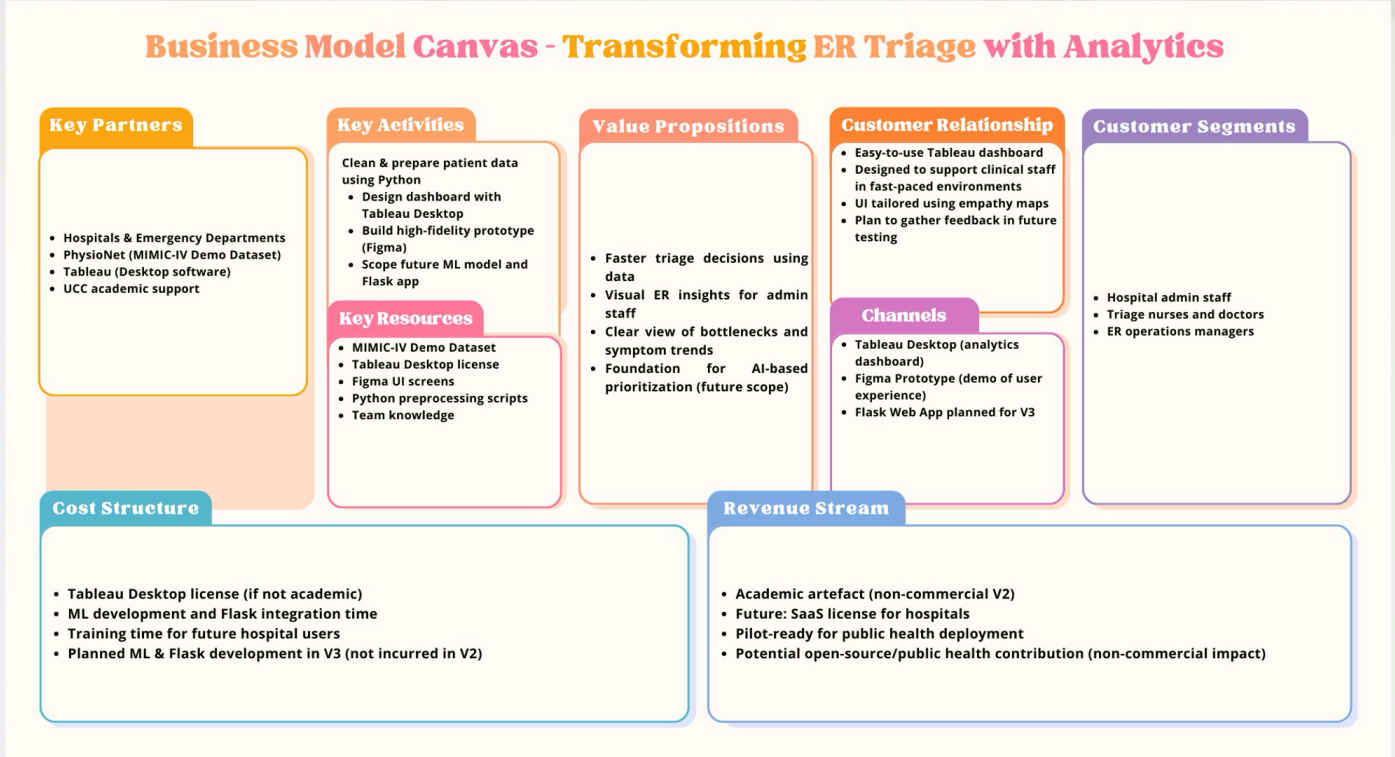
As shown in **Figure 16**, our long-term goal is to evolve the system into a **low-cost SaaS solution** for small and rural hospitals — offering core triage features at a fraction of the cost of commercial tools like Salesforce Health Cloud.

By improving triage accuracy and reducing delays, the artefact contributes meaningfully to:

- **SDG 3**: Ensuring timely care
- **SDG 9**: Promoting healthcare innovation

It holds strong potential for real-world impact in underserved settings.

Figure 16: Business Model Canvas - Transforming ER Triage with Analytics



9. References

Al-Hraishawi, H. and Thabtah, F., 2020. A mobile-based COVID-19 risk triage: Design and implementation. *Health and Technology*, 10(6), pp.1429–1437.

Nagle, T. and Sammon, D., 2017. Data Value Map. Cork University Business School, University College Cork. [online] Available at: http://datavaluemap.com [Accessed 27 May 2025].

Chen, L., Wang, L., Yu, S., Xu, L. and Wu, H., 2019. Data-driven triage prediction for emergency department using ensemble learning. *BMC Medical Informatics and Decision Making*, 19(1), p.51.

Goldberger, A.L., Amaral, L.A.N., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.K. and Stanley, H.E., 2000. PhysioBank, PhysioToolkit, and PhysioNet. *Circulation*, 101(23), pp.e215–e220.

MIT Laboratory for Computational Physiology, 2022. *MIMIC-IV Emergency Department Module*. [online] PhysioNet. Available at: https://physionet.org/content/mimic-iv-ed/ [Accessed 27 May 2025].

Nagendran, M., Chen, Y., Lovejoy, C.A., Gordon, A.C., Komorowski, M., Harvey, H., Topol, E.J. and Ioannidis, J.P.A., 2020. Artificial intelligence versus clinicians: systematic review of design, reporting standards, and claims of deep learning studies. *BMJ*, 368, p.m689.

Pandas Development Team, 2023. *pandas: powerful Python data analysis toolkit*. [online] Available at: https://pandas.pydata.org/ [Accessed 27 May 2025].

Salesforce, 2024. *Salesforce Health Cloud Overview*. [online] Available at: https://www.salesforce.com/products/health-cloud/overview/ [Accessed 27 May 2025].

Scikit-learn Developers, 2024. *scikit-learn: Machine Learning in Python*. [online] Available at: https://scikit-learn.org/ [Accessed 27 May 2025].

Tan, J., 2019. *Healthcare Information Systems: A Practical Approach for Health Care Management*. 3rd ed. San Francisco: Jossey-Bass.

World Health Organization (WHO), 2022. *Sustainable Development Goals (SDG 3 and SDG 9)*. [online] Available at: https://sdgs.un.org/ [Accessed 27 May 2025].

Zhang, Z., 2016. Introduction to machine learning: k-nearest neighbors. *Annals of Translational Medicine*, 4(11), p.218.