

PRODUCT REQUIREMENT DOCUMENT

BERTH PLANNING AND ALLOCATION OPTIMISATION

Date	Author	Version	Affected Module
29/01/26	Niraj Kumar	1.0	Berthing

Team Member(s)

Name	Role
Dipesh Pansuriya	Team Lead – Developer
Ankur Rai	Developer
Sayandeep Haldar	Developer
Niraj Kumar	Product Manager

Reviewer(s)

Name	Role	Sign Off Date
Shruti Pandit	Mentor – AI/ML	

PROBLEM DEFINITION

Ports handle multiple vessel calls each day, with every vessel having unique characteristics such as estimated arrival time, cargo volume, service priority, and berth requirements. Current berth planning practices are often **manual or rule-based**, making them highly sensitive to delays caused by weather conditions, tidal variations, or operational constraints.

Even minor deviations in vessel arrival times can result in:

- Berth congestion
- Increased vessel waiting and idling time
- Sub-optimal utilization of berth infrastructure
- Reduced port throughput and operational inefficiencies

There is a need for an **intelligent, data-driven berth planning system** that can accurately predict vessel arrival times and dynamically optimize berth allocation to ensure efficient port operations.

PROBLEM STATEMENT

How can Artificial Intelligence be used to **predict vessel arrival times** and **optimise berth allocation** in order to:

- Minimise vessel waiting time at anchorage
- Improve vessel turnaround time (TAT)
- Increase overall port throughput and berth utilisation?

OBJECTIVE OF SMARTBERTH AI

SmartBerth AI is an AI-powered berth planning and allocation system that enables Terminal Operators to:

- Proactively plan berth schedules based on predicted ETAs
- Dynamically re-optimise berth allocation in response to delays and disruptions
- Improve operational efficiency and decision-making through data-driven recommendations

SmartBerth AI – AI Use Cases

1. Vessel Arrival & Readiness Intelligence

1.1 Predictive ETA Calculation

AI predicts actual vessel arrival time instead of relying on declared ETA.

Data Points

- Historical arrival/departure data
- AIS movement patterns
- Weather & tidal data
- Port congestion patterns

1.2 Arrival Deviation Detection

AI continuously compares:

Planned ETA vs Predicted ETA vs Actual movement

Use Case

- Early detection of delays (6–24 hrs before arrival)
- Automatic alert to planners

1.3 Vessel Readiness Prediction

AI predicts whether a vessel is *berth ready*.

Factors

- Pilot/Tug availability
- Tidal window
- Previous port departure delays
- Regulatory clearance readiness (high-level)

2. Berth Allocation, Dynamic Re-Planning

2.1 Constraint-Based Berth Allocation

AI matches vessels to berths using operational constraints:

Constraints

- Vessel LOA / beam / draft
- Cargo type compatibility
- Berth equipment availability
- Tidal restrictions

2.2 Real-Time Re-Optimisation

When delays occur, AI:

- Re-calculates berth schedules
- Minimizes cascading impact on other vessels

2.3 Conflict Detection & Resolution

AI detects:

- Berth overlaps
- Resource clashes
- Tidal window conflicts

Then auto-suggests resolutions, not just alerts.

3. What-If Simulation & Decision Support

3.1 Delay Impact

Planner asks:

"What happens if Vessel AAA arrives 8 hours late?"

AI simulates:

- Impact on waiting times
- Berth utilisation
- Downstream vessel delays

3.2 Capacity Management

AI simulates:

- Surge in arrivals
- Berth closure (maintenance)
- Weather disruption scenarios

USERS & PERSONAS

Primary User(s)

Terminal Operator(s)

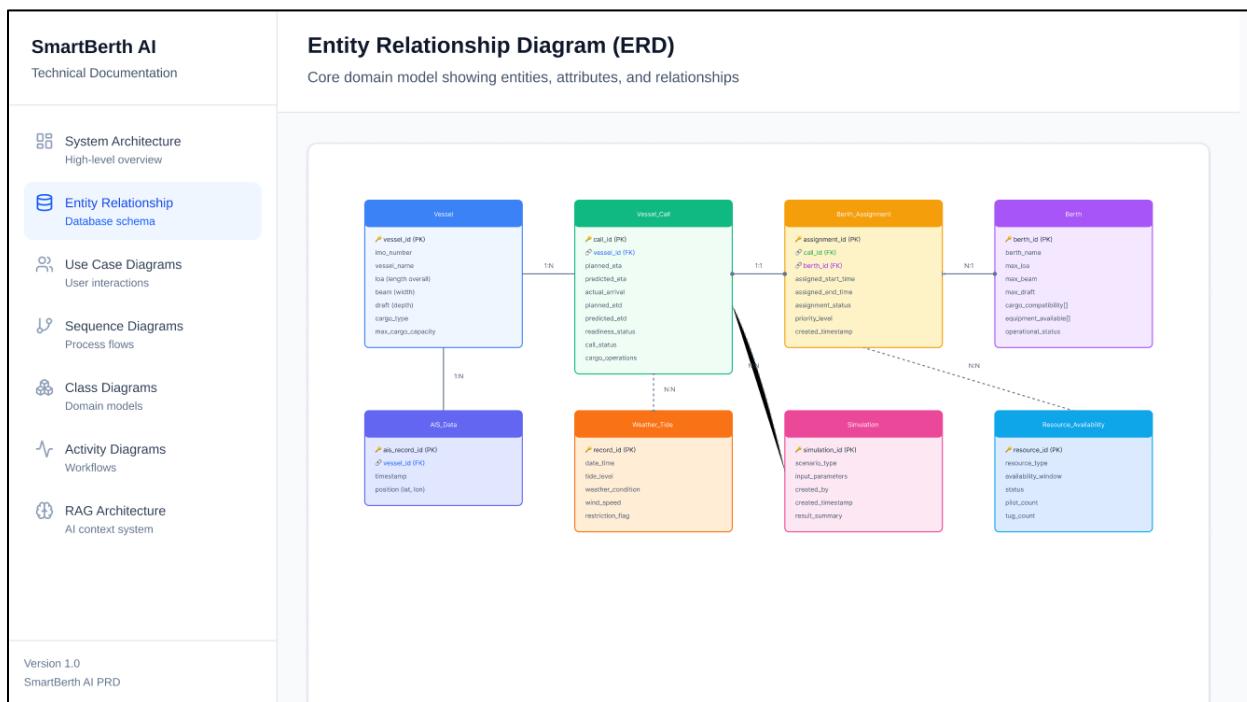
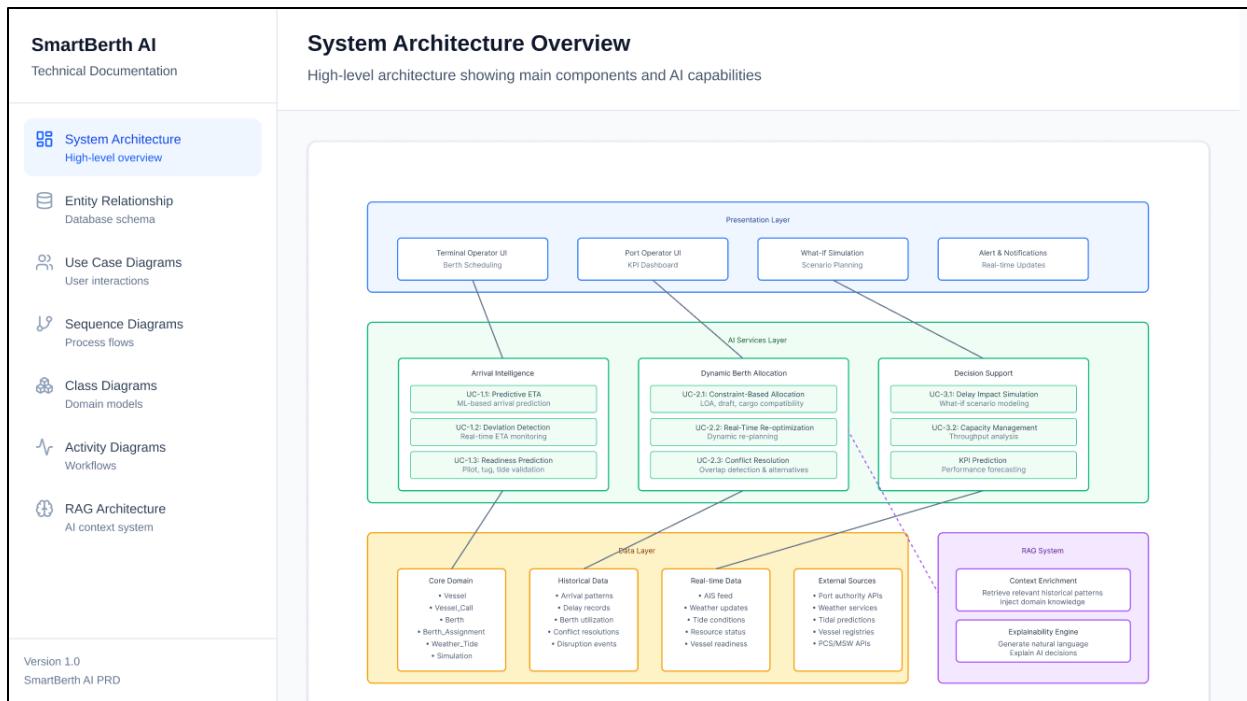
- Responsible for berth scheduling and vessel sequencing
- Needs advance visibility and quick re-planning during disruptions

Secondary User(s)

- Port Operator(s) – monitors throughput & KPIs

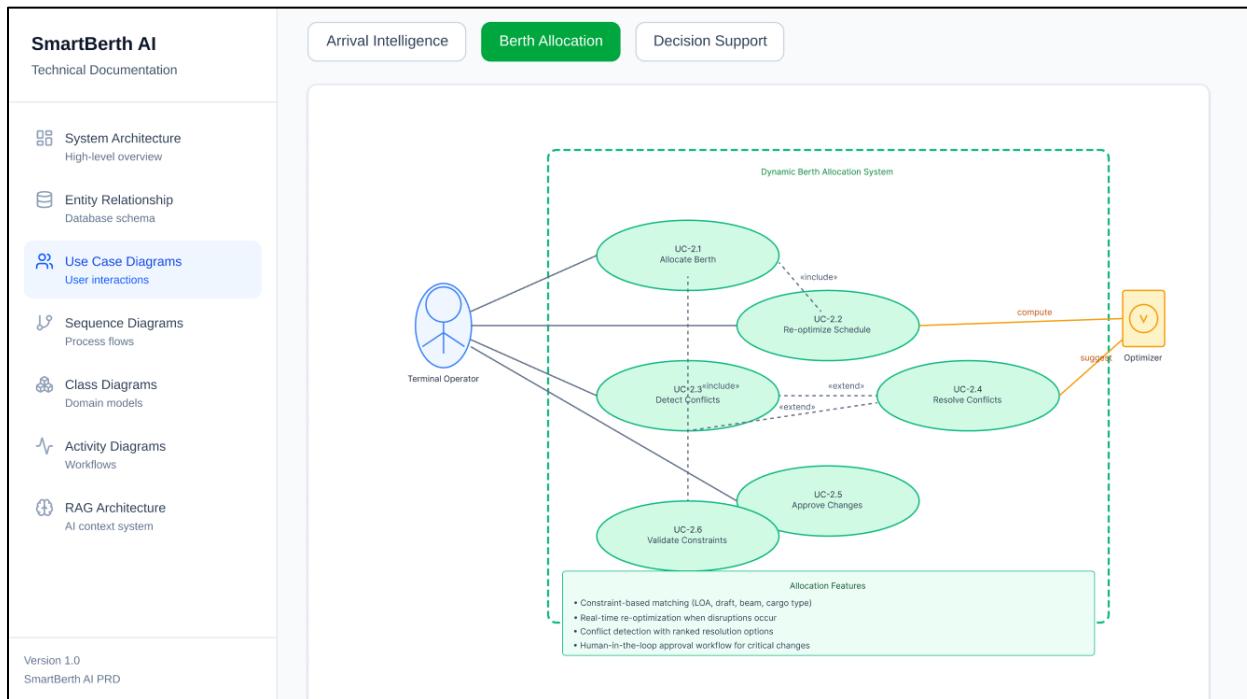
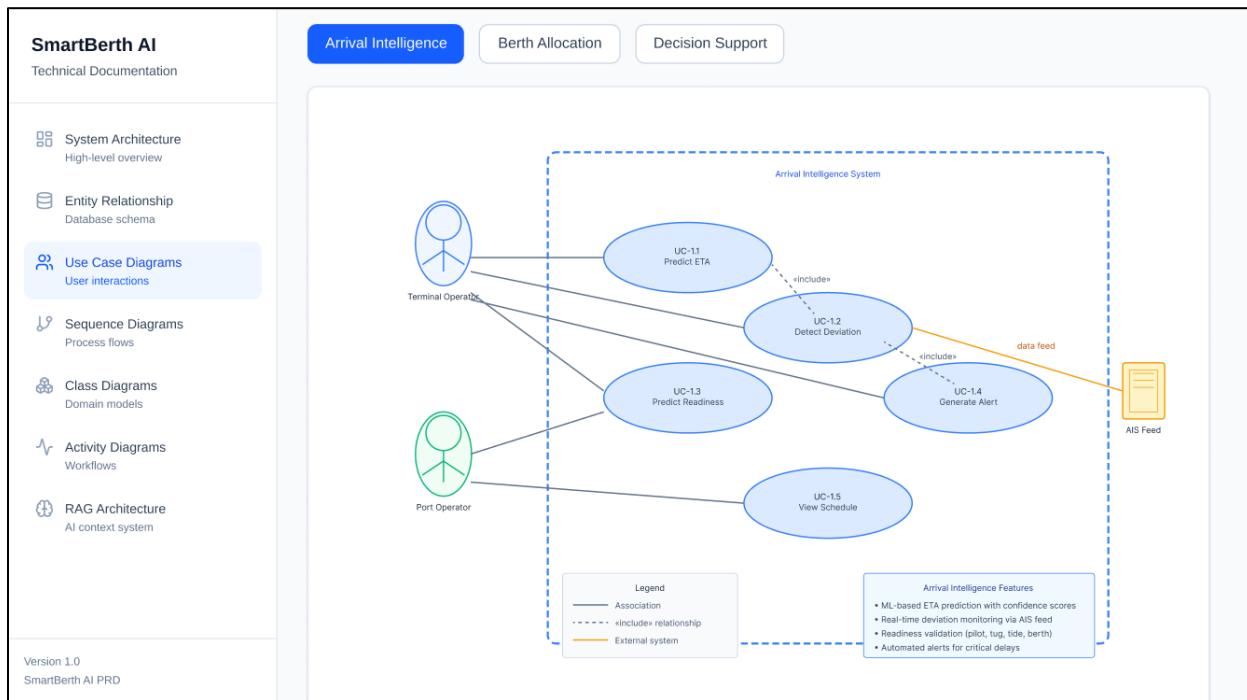
Berth Planning and Allocation Optimisation

SmartBerth AI – UML, ERD, RAG Mapping Diagrams

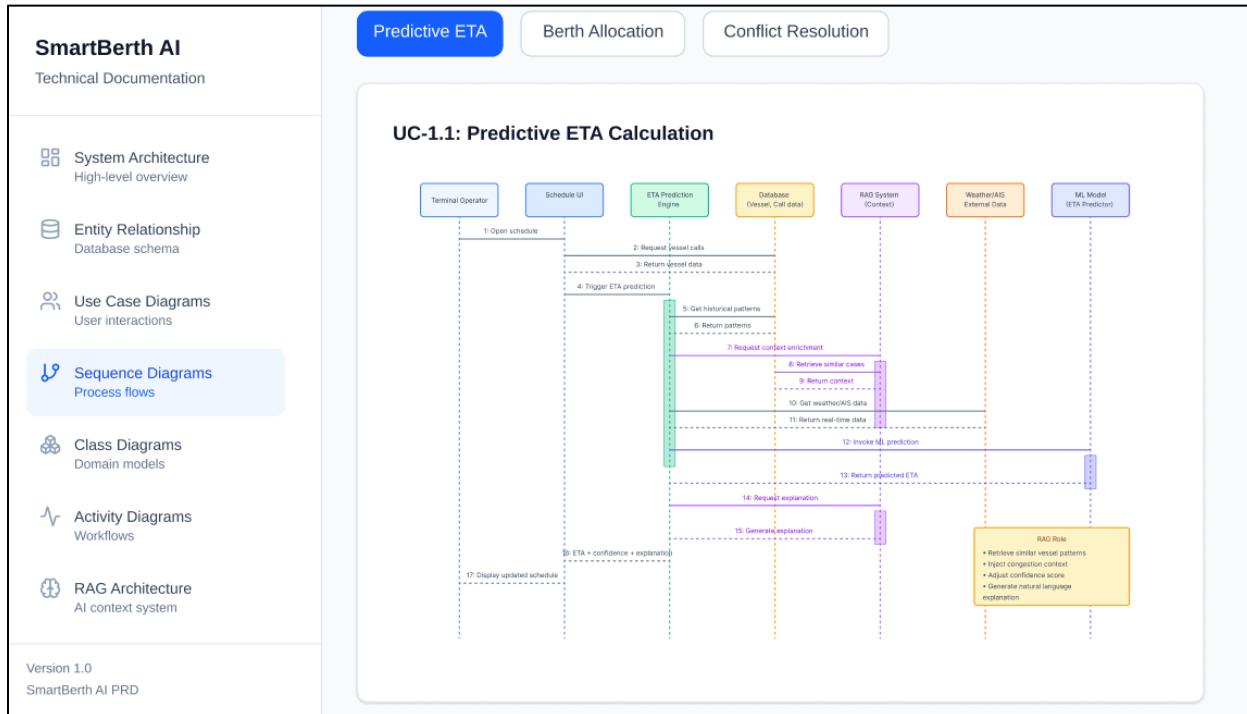
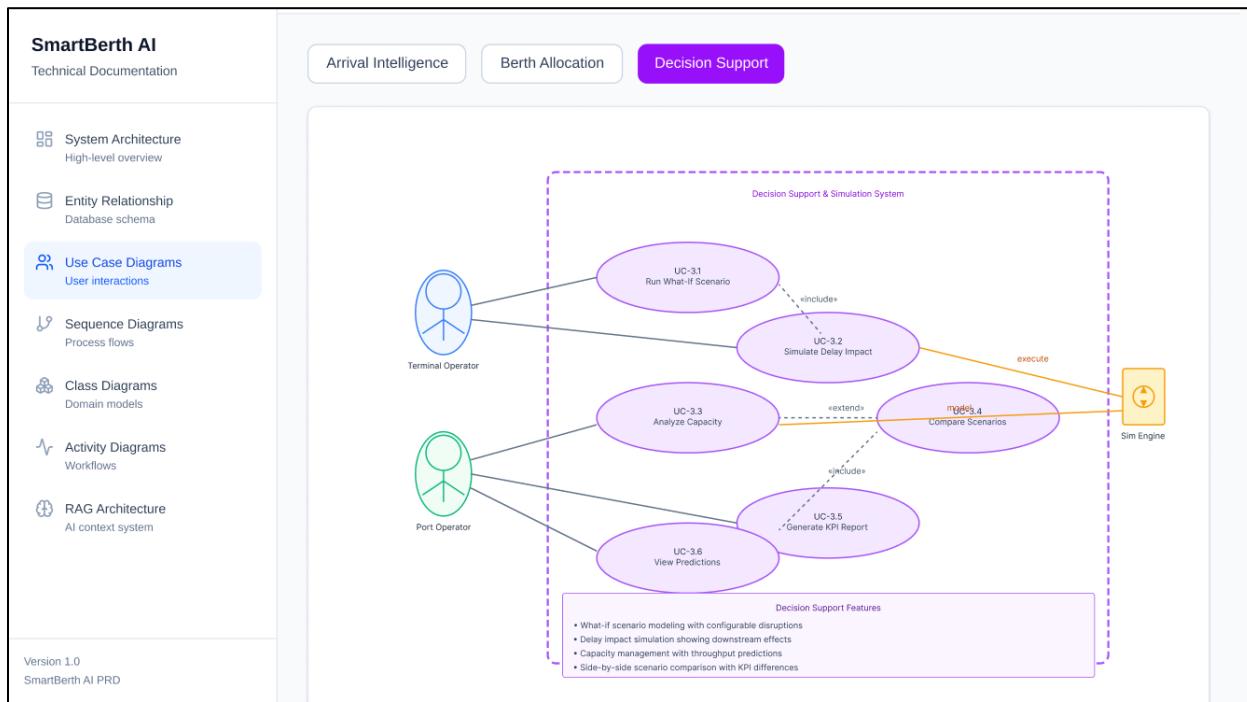


KaiTETHON - India's premier AI-powered logistics and cargo hackathon

Berth Planning and Allocation Optimisation



Berth Planning and Allocation Optimisation



Berth Planning and Allocation Optimisation

SmartBerth AI
Predictive ETA
Berth Allocation
Conflict Resolution

Technical Documentation

- System Architecture High-level overview
- Entity Relationship Database schema
- User Use Case Diagrams User interactions
- Sequence Diagrams Process flows
- Class Diagrams Domain models
- Activity Diagrams Workflows
- RAG Architecture AI context system

Version 1.0
SmartBerth AI PRD

UC-2.1: Constraint-Based Berth Allocation

```

sequenceDiagram
    participant VesselManager
    participant AllocationService
    participant ReadinessAssessment
    participant ConstraintEngine
    participant Database
    participant RAGSystem
    participant OptimizationEngine

    VesselManager->>AllocationService: 1. Request allocation
    AllocationService->>ReadinessAssessment: 2. Verify readiness
    ReadinessAssessment->>VesselManager: 3. Confirm ready
    AllocationService->>ConstraintEngine: 4. Get available berths
    ConstraintEngine->>Database: 5. Return berth list
    Database->>AllocationService: 6. Get constraint rules
    AllocationService->>OptimizationEngine: 7. Rank berths
    OptimizationEngine->>AllocationService: 8. Evaluate constraints
    AllocationService->>ConstraintEngine: 9. Compatible berths
    ConstraintEngine->>AllocationService: 10. Rank berths (optimized)
    AllocationService->>RAGSystem: 11. Best berth selected
    RAGSystem->>AllocationService: 12. Create assignment
    AllocationService->>OptimizationEngine: 13. Assignment created
    OptimizationEngine->>VesselManager: 14. Berth allocated
  
```

RAG Previews

- Berth capacity definitions
- Cargo compatibility rules
- Equipment requirements
- Explanation for selection

SmartBerth AI
Predictive ETA
Berth Allocation
Conflict Resolution

Technical Documentation

- System Architecture High-level overview
- Entity Relationship Database schema
- User Use Case Diagrams User interactions
- Sequence Diagrams Process flows
- Class Diagrams Domain models
- Activity Diagrams Workflows
- RAG Architecture AI context system

Version 1.0
SmartBerth AI PRD

UC-2.3 & UC-2.4: Conflict Detection & Resolution

```

sequenceDiagram
    participant SystemTimer
    participant ConflictDetector
    participant Database
    participant ResolutionEngine
    participant RAGSystem
    participant OptimizationEngine
    participant TerminalOperator

    SystemTimer->>ConflictDetector: 1. Periodic check
    ConflictDetector->>Database: 2. Get assignments
    Database->>ConflictDetector: 3. Assignment list
    ConflictDetector->>ResolutionEngine: 4. Trigger resolution
    note over ResolutionEngine: ▲ Conflict Found
    ResolutionEngine->>Database: 5. Get past resolutions
    ResolutionEngine->>RAGSystem: 6. Historical patterns
    RAGSystem->>ResolutionEngine: 7. Generate alternatives
    ResolutionEngine->>OptimizationEngine: 8. Rank options
    OptimizationEngine->>ResolutionEngine: 9. Request expansion
    ResolutionEngine->>TerminalOperator: 10. Explanation text
    ResolutionEngine->>Database: 11. Present options
    Database->>ResolutionEngine: 12. Select resolution
    ResolutionEngine->>OptimizationEngine: 13. Update assignments
    OptimizationEngine->>TerminalOperator: 14. Conflict resolved
  
```

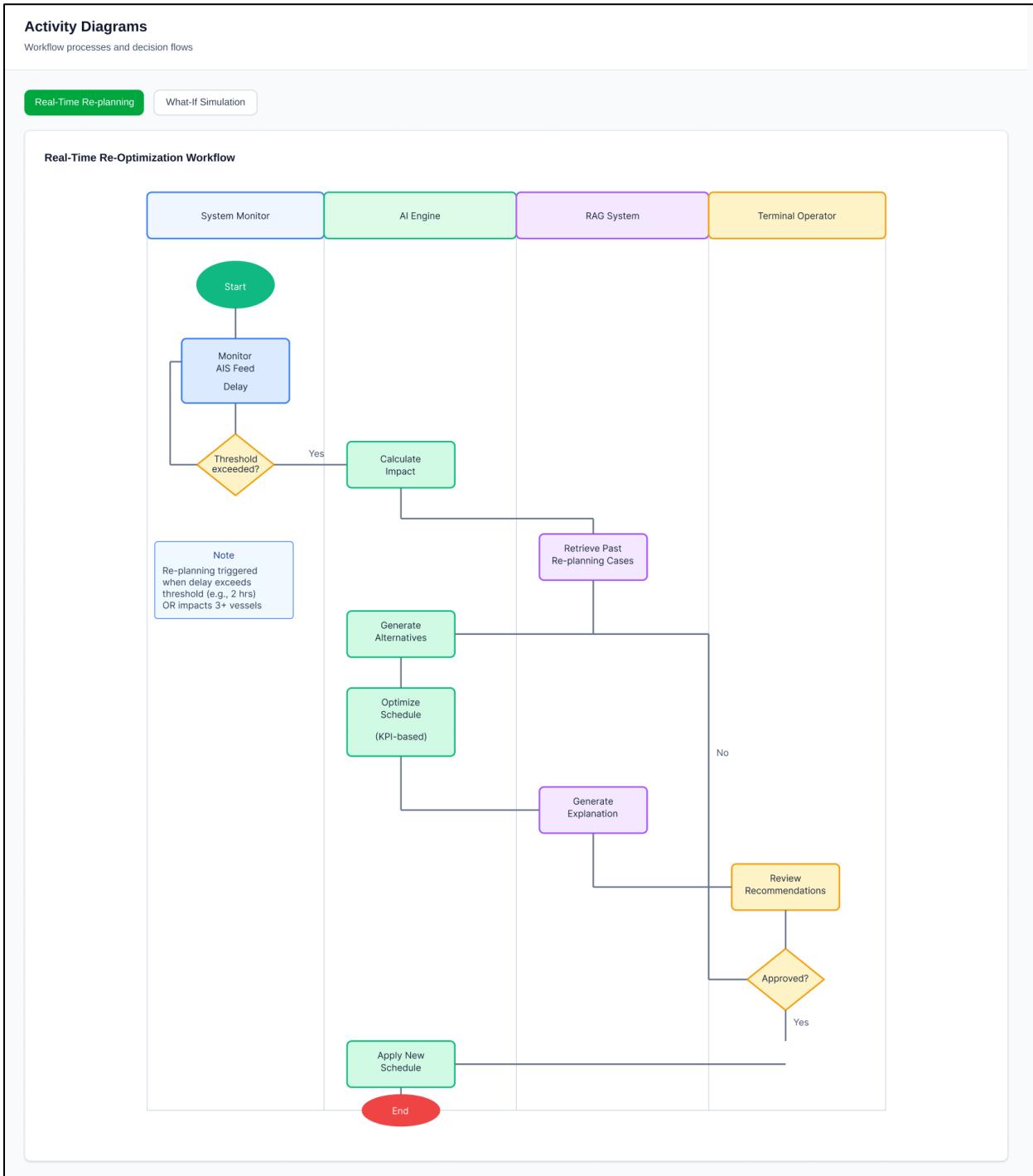
Conflict Types

- Berth overlap
- Time window conflict
- Resource unavailability
- Title constraint
- Equipment isolation

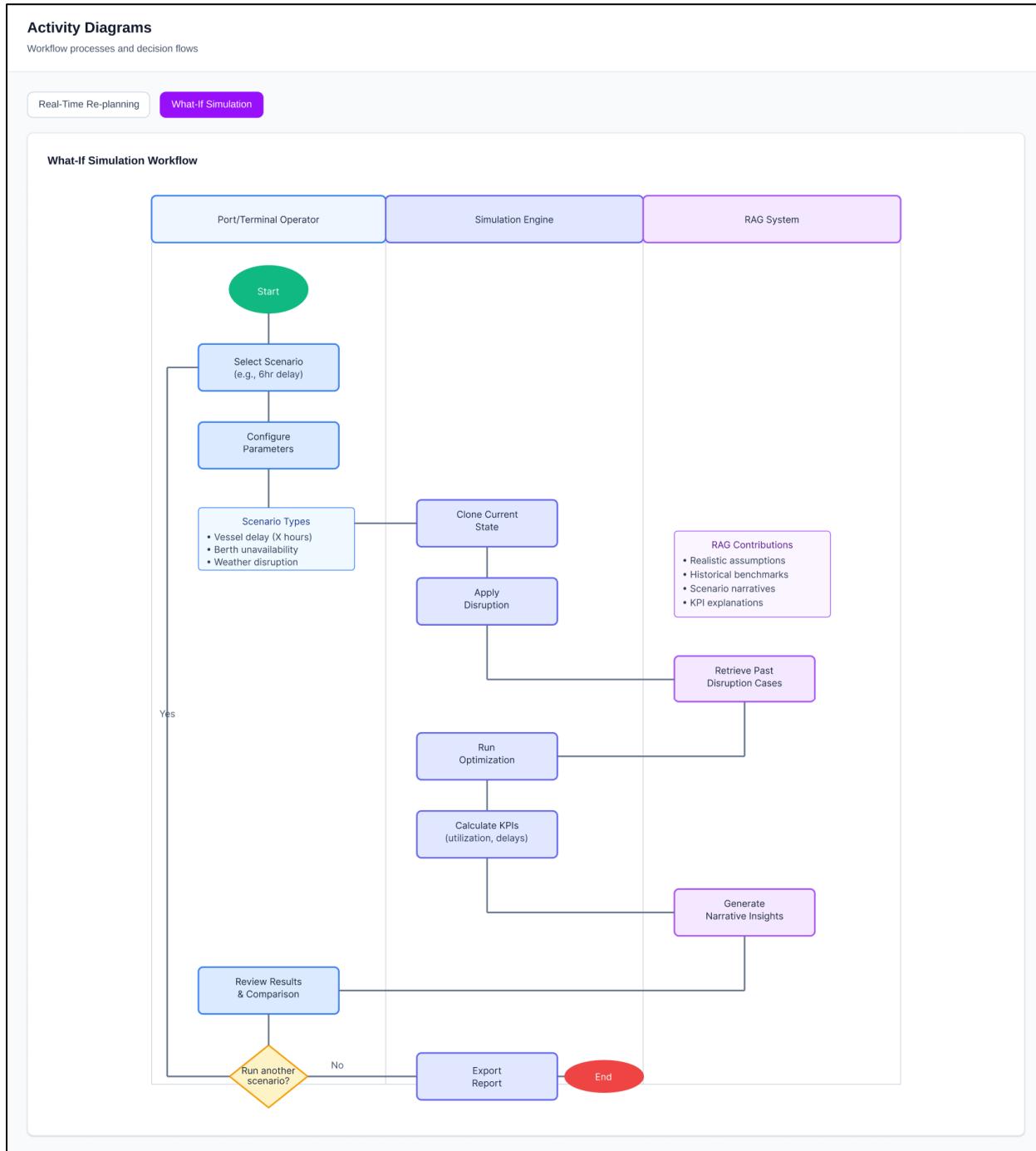
Resolution Options

- Shift vessel A to berth B
- Delay vessel C by 2 hrs
- Re-sequence vessels

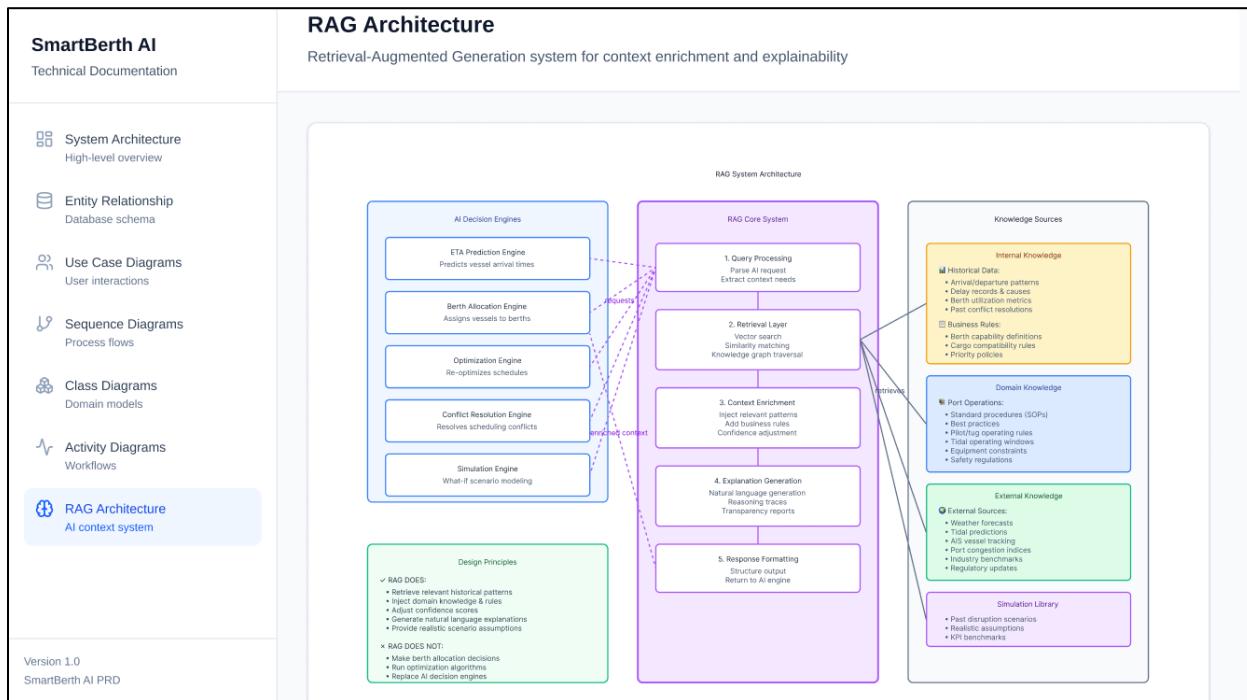
Berth Planning and Allocation Optimisation



Berth Planning and Allocation Optimisation



Berth Planning and Allocation Optimisation



FUNCTIONAL REQUIREMENTS – WIP