Requirements Definition

Production Optimiser

Politecnico di Milano, Information Engineering School, Italy
University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia
Distributed Software Development

21st of November 2024

Table of contents

1. Project Background

1.1 Project Purpose

The primary purpose of this project is to develop a sophisticated web-based platform that enables users to interact with and execute production optimization models. In an era where data-driven decision making is crucial for production efficiency, this platform aims to provide an intuitive interface for running complex optimization models while maintaining security and scalability.

The Production Optimiser will enable users to:

- Upload production data through Excel files
- Execute various optimization models
- Visualize results through PLT graphs
- Track historical optimizations
- Compare different optimization results
- Access results through a standalone API

1.2 Project Goals

1.2.1 [G1] Create a Secure and Scalable Platform

- Implement robust authentication and authorization
- Ensure data privacy and security
- Support concurrent model executions
- Enable horizontal scaling of services

1.2.2 [G2] Enable Efficient Model Management

- Support multiple service tools (e.g. Optimisation model)
- Manage linking of new tools and modifications of existing ones
- Enable model-specific access control
- Support standalone API access

1.2.3 [G3] Optimize Data Handling

- Support file uploads as tool input (e.g. Excel file)
- Simple representation of the tool output (e.g. graphs)
- Store and manage historical data
- Enable result comparisons

1.2.4 [G4] Implement Role-based Access

- Support admin and customer roles
- Enable granular permission management
- Allow model-specific access control
- Support invitation-based registration

1.2.5 [G5] Provide Comprehensive Analytics

- Track model usage and performance
- Monitor user activity
- View usage reports

1.2.6 [G6] Ensure System Reliability

- Implement error handling and recovery
- Provide system health monitoring
- Allow ease of modification

1.3 Product Scope

1.3.1 Frontend Scope

Core Features:

- Authentication and user management interface
- Model selection and configuration
- File upload management
- Results visualization
- Historical data access
- Analytics dashboard

Interface Elements:

- Login and registration forms
- Model selection dropdown
- File upload component
- Model response view
- Results download interface
- Admin dashboard

User experience:

- Responsive design for all screen sizes
- Intuitive navigation
- Clear error messages
- Loading indicators
- File Upload Progress tracking

1.3.2 Backend Scope

Api Endpoints:

- User authentication and management
- Model execution orchestration
- File upload handling
- Results management
- Analytics data collection

Data Management:

- User data storage
- Model configurations
- Input file storage
- Results storage
- Performance metrics

Security Features:

- JWT authentication
- Role-based access control
- Sensitive data encryption
- Secure file handling

1.3.3 Service Tool Scope

Core Functionality:

- Executing existing service functions
- Input file validation and cleaning (if necessary)
- Result optimization and cleaning (if necessary)

Integration Features:

- REST API endpoints
- Error handling
- Latency impact of up to 10% of the existing service execution time

Performance Monitoring:

- Execution time tracking
- Error and success rate tracking

2. Terminology and Definitions

2.1 Terminology

Abbr.	Full Name	Description
ANON	Anonymous User	Unauthenticated user with access only to login/registration request
REG	Regular User	Authenticated user with model access rights
ADMIN	Administrator	User with full system administrative privileges
PLT	Plot	Output graph from service tools
TOOL	Service Tool	Group of functions that form an algorithm that is being executed on command
FE	Frontend	Web-based user interface
BE	Backend	Server-side application logic
API	Application Programming Interface	Interface for programmatic access to models
UI/UX	User Interface/Experience	Application interface and interaction design

2.2 Actors and Stakeholders

2.2.1 Primary Actors

1. Anonymous User
    - Can request access to the platform
    - Can view basic platform information
2. Regular User
    - Can access assigned optimization models
    - Can upload input files
    - Can view and download results
    - Can track optimization history
3. Administrator
    - Can manage user access
    - Can configure models
    - Can view system analytics
    - Has all regular user capabilities

2.2.2 Supporting Actors

1. Service Tool
    - Executes different algorithms
    - Transfers service tool outputs to the API
    - Provides performance metrics
2. Authentication Service
    - Manages user authentication
    - Handles session management
    - Controls access permissions

2.3 System Boundaries and Constraints

1. Technical Boundaries
    - Web-based platform only
    - Excel file input only
    - REST API communication
2. Functional Boundaries
    - Versioning or service tools is not supported
    - Maximum file size restrictions
    - Processing time mostly depend on service tool
3. Security Constraints
    - HTTPS only
    - JWT authentication
    - Role-Based access
    - Sensitive data encryption
4. Performance Constraints
    - Response time will be up to 10% on top of the existing service execution time
    - Service tool performance mostly depend on internal algorithm and execution container resources

3. Requirements Analysis

3.1 Functional Requirements

3.1.1 User Authentication and Management

[FR1] User Authentication

- Support email + password login
- Implement session management
- Handle account deletion

[FR2] Account management

- Invitation-based registration
- Enable credential change
- Account deletion

[FR3] Admin controls

- User invitation
- Service Tool management
- User activity monitoring
- Model activity monitoring

[FR4] Session Handling

- Automatic session timeout
- Concurrent session management
- Remember me functionality
- Secure logout

3.1.2 Service Tool Management

[FR5] Service Tool Support

- Support for multiple tools
- Tool parameters configuration
- Tool input/output configuration
- Input validation

[FR6] Admin tool controls

- Service tools linking
- Access control
- Modifying existing models

[FR7] Service Tool Access

- Role-based access
- User-based access

[FR8] API integration

- RESTful endpoints
- Authentication tokens
- Error handling

3.1.3 File Operations

[FR9] File Upload

- Excel file support
- Upload progress tracking

[FR10] Results Download

- Multiple format support
- Whole model response download
- Single file download
- Error handling

[FR11] Service Tool output transfer

- Save and return output file (e.g. Graph)
- Download options

[FR12] Historical Data

- Input and output storage
- Comparing history results

3.1.4 Analytics and Monitoring

[FR13] Usage statistics

- Service tool execution counts
- Login tracking
- Success/failure rates
- Response times

[FR14] Health metrics

- System health monitoring
- DB health monitoring

3.2 Non-Functional Requirements

3.2.1 Security Requirements

[NFR1] Authentication Security

- Password encryption
- Token management
- Session security
- Access control

[NFR2] Data Protection

- Secure transmission
- Access logging
- Privacy compliance

[NFR3] Communication Security

- HTTPS for all calls
- API security
- Input validation
- Output sanitization

[NFR4] Security Monitoring

- Audit logging
- Vulnerability scanning
- Security updates

3.2.2 Performance Requirements

[NFR5] Concurrency

- Multiple user support
- Parallel model execution

[NFR6] Response Time

- API response SLA
- Page load times
- Model execution time

[NFR7] File Handling

- Efficient storage
- Quick retrieval
- Compression

[NFR8] Scalability

- Docker deployment ready
- Horizontal scaling

3.3 Use Cases

3.3.1 [UC1] User Registration

Actor	Anonymous User
Entry Condition	User needs platform access
Event Flow	1. User visit registration request page 2. Fills email and company details 3. Submits access request 4. Admin receives notification 5. Admin reviews request 6. System sends invitation if approved
Exit Condition	User receives platform invite
Exception	• Invalid email format • Duplicate email request
Special Requirements	• Request must include business justification • Admin response within 48 hours

3.3.2 [UC2] Service Tool Execution

Actor	Regular User
Entry Condition	User is authenticated and has tool access
Event Flow	1. User selects service tool 2. Uploads input file 3. System validates file format 4. System sends file to the tool 5. Service tool processes data 6. Service tool generates result 7. System returns generated result 8. Results available for download
Exit Condition	Service Tool results available
Exception	• Invalid file format • Tool execution error • Max File size exceeded
Special Requirements	• Excel file size limit: 50MB

3.3.3 [UC3] Result Management

Actor	Regular User
Entry Condition	User has completed service calls
Event Flow	1. User accesses service call history 2. Selects specific service call 3. Views results 4. Downloads results or compares with other runs
Exit Condition	User obtains required results
Exception	• Results not found • Download error
Special Requirements	

3.3.4 [UC4] User Management

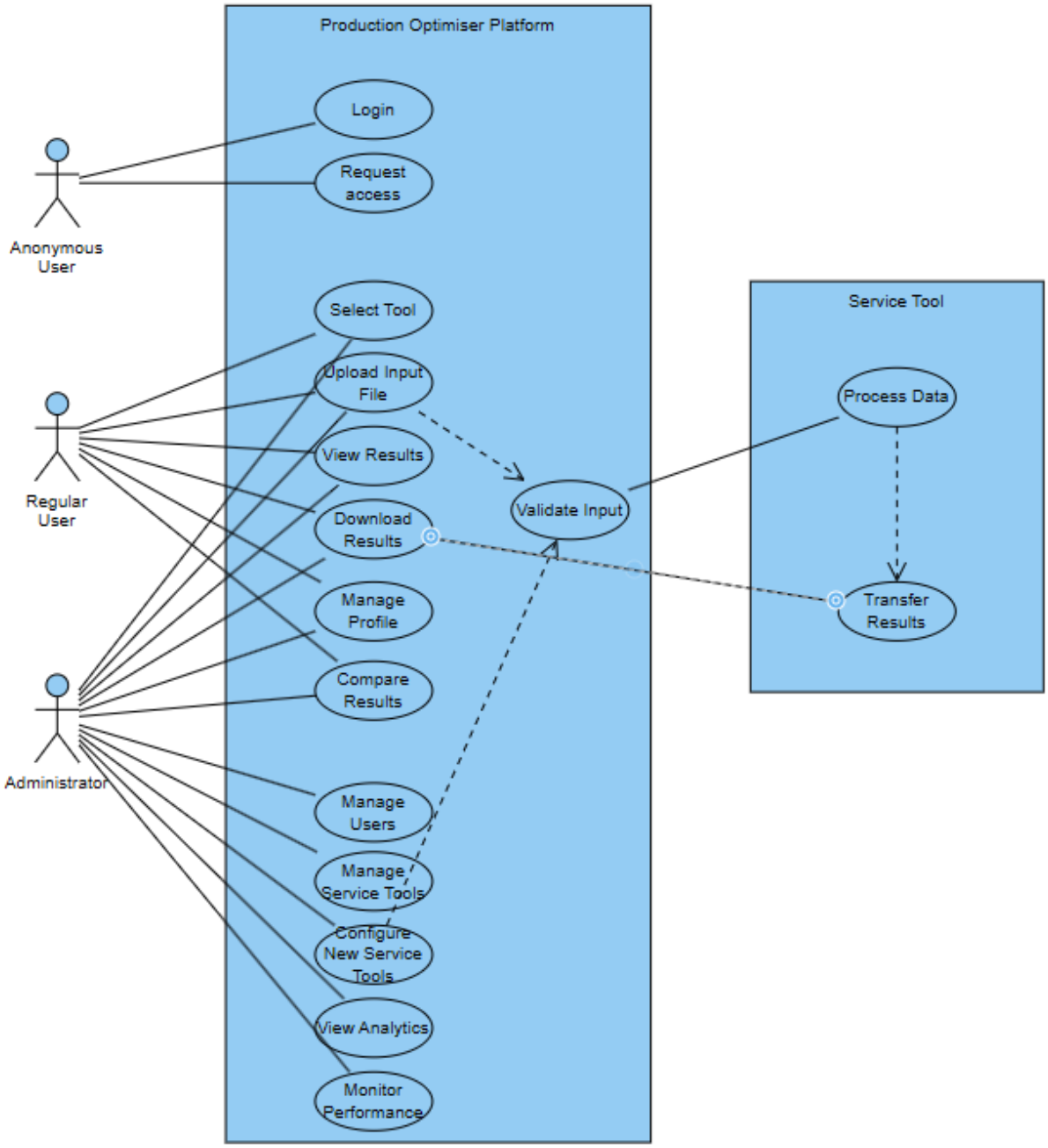
Actor	Administrator
Entry Condition	Admin is authenticated
Event Flow	1. Admin accesses user management 2. Reviews access requests 3. Manages user permissions 4. Assigns models to users 5. Reviews user activity
Exit Condition	User access configured
Exception	• Invalid permissions
Special Requirements	

3.3.5 [UC6] Model Configuration

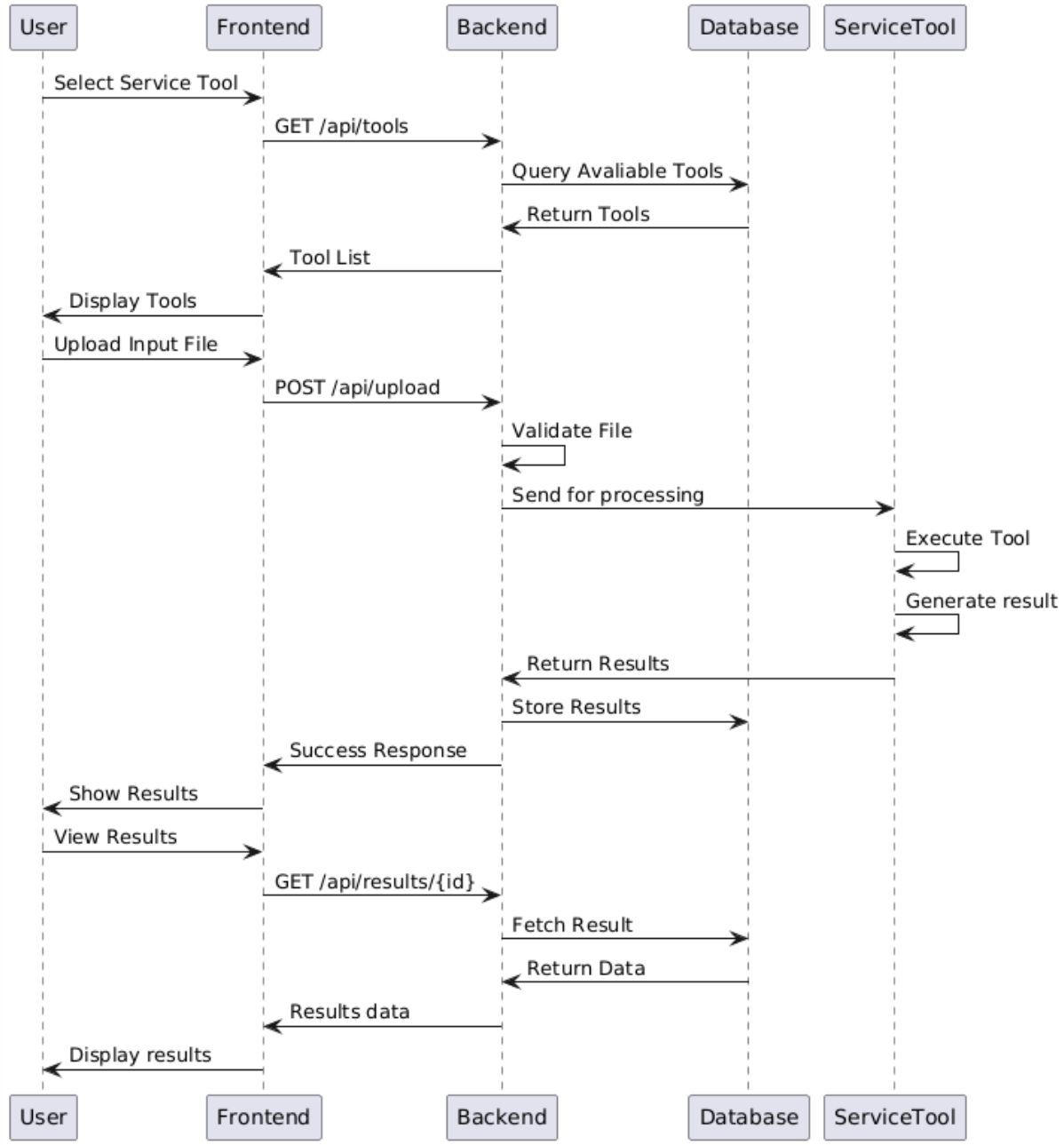
Actor	Administrator
Entry Condition	Admin is authenticated
Event Flow	7. Admin accesses service tool management 8. Configures tool parameters 9. Sets access permissions 10. Tests tool execution 11. Publishes changes
Exit Condition	Service tool ready for use
Exception	• Configuration error • Test execution failure
Special Requirements	Service tool healthcheck test must pass

3.5 Diagrams

Actor-Relationship Use Case Diagram



Model Execution – Sequence Diagram



3.4 User Stories

3.4.1 [US1] User Registration

As an	Anonymous User
I want to	request access to the platform
So that	I can use service tools for my data
Acceptance Criteria	• Can submit registration request • Receive confirmation email • Get notified of request status
Priority	High
Notes	

3.4.2 [US2] Model Access

As an	Regular User
I want to	run service tools on my data
So that	I can benefit from platforms tools on my data
Acceptance Criteria	• Can select available models • Upload input files • Receive tool results • Download results graphs
Priority	High
Notes	• Ensure input validation

3.4.3 [US3] Result Analysis

As an	Regular User
I want to	view and compare tool results
So that	I can make informed decisions
Acceptance Criteria	• Access historical results • Compare multiple runs • Download comparison reports • View result graphs
Priority	Medium
Notes	• Include export functionality

3.4.4 [US4] User Management

As an	Administrator
I want to	manage user access and permissions
So that	I can control platform usage
Acceptance Criteria	• Review access requests • Set user permissions • Monitor user activity • Manage model access
Priority	High
Notes	• All admins have the same permissions

3.4.5 [US5] Model Management

As an	Administrator
I want to	configure and monitor tools
So that	I can ensure optimal platform performance
Acceptance Criteria	• Configure tool parameters • Monitor execution metrics • Update tool endpoints • Set access controls
Priority	High
Notes	

4. System Architecture

The system follows a component-based architecture. This architecture ensures clear separation of concerns and enables scalability of individual components.

4.1 Frontend

Technologies: React, Vite, Tailwind CSS, Axios

Components:

- Authentication module
- Model interface
- File management
- Results viewer
- Admin dashboard
- Analytics display

State Management:

- User session
- Model state
- File uploads
- UI preferences

4.2 Backend

Technologies: Spring Boot, Spring Security, JWT, PostgreSQL, Test Containers

Components:

- User service
- Service Tool Service
- File service
- Analytics service

Data Layer:

- Database management
- File storage
- Session storage

4.3 Service Tool Backend

Service Tool backend is separate container running customers algorithms wrapped inside RESTful API to send input files and retrieve its data results.

Technologies: Python, FastAPI

Components:

- Model executor
- Input processor
- Output processor