Enterprise Application Project (EAP) - Product Vision

Version: 1.0
Date: 2026-01-06
Owner: Product Owner (trainee role)
Approved by: Staff

Executive Summary

The Enterprise Application Project (EAP) is a web-based request management system designed to streamline employee requests for resources, access, and services within an organization.

The system provides a structured workflow for submitting, reviewing, approving, and tracking requests while maintaining clear audit trails and automated notifications.

Target Users: Organizations needing to manage employee requests for hardware, software, access rights, and other corporate resources.

Vision Statement

> **"Enable employees to request resources efficiently while giving managers visibility and control over approvals, all within a transparent and auditable system."**

The Problem We're Solving

Current State (Without EAP)

Organizations typically handle employee requests through:

- **Email chains** - Difficult to track, easy to lose, no audit trail
- **Spreadsheets** - Manual updates, no automation, prone to errors
- **Ad-hoc processes** - Inconsistent, depends on who you know
- **Paper forms** - Slow, not scalable, hard to archive

Pain Points:

- 😞 Employees don't know status of their requests
- 😞 Managers lose track of pending approvals
- 😞 No visibility into request history
- 😞 Difficult to audit who approved what
- 😞 No standardized process across departments
- 😞 Time wasted on follow-ups and status checks

Future State (With EAP)

With the EAP system:

- ✅ **Self-service portal** for employees to submit requests
- ✅ **Automated routing** to correct approvers
- ✅ **Real-time status** visibility for all parties
- ✅ **Email notifications** at each workflow stage
- ✅ **Complete audit trail** of all actions
- ✅ **Standardized process** across organization
- ✅ **Dashboard** for managers and admins

Core Workflow

The EAP system implements a three-step workflow:

```
┌─────────────────┐     ┌─────────────────┐     ┌─────────────────┐
│   STEP 1:       │  →  │   STEP 2:       │  →  │   STEP 3:       │
│   Submit        │     │   Approve       │     │   Fulfill       │
│                 │     │                 │     │                 │
│ Employee fills  │     │ Manager/Admin   │     │ Request is      │
│ request form    │     │ reviews and     │     │ fulfilled and   │
│                 │     │ approves/rejects│     │ marked complete │
└─────────────────┘     └─────────────────┘     └─────────────────┘
        ↓                       ↓                       ↓
    Submitted          Under Review / Approved      Completed
```

Workflow States

State	Description	Who Can Act
Draft	Request being created	Requester
Submitted	Awaiting review	System (auto-routes to approver)
Under Review	Being reviewed	Approver
Approved	Approved, awaiting fulfillment	Admin/Fulfiller
Rejected	Denied with reason	- (terminal state)
Completed	Fulfilled and closed	- (terminal state)
Cancelled	Requester cancelled	- (terminal state)

User Roles

1. Requester (Employee)

Who: Any employee in the organization

Capabilities:

- Submit new requests
- View their own request history
- Track status of pending requests
- Receive email notifications on status changes
- Cancel own pending requests (before approval)
- Add comments/additional information to requests

Use Case Examples:

- "I need a new laptop for development work"
- "I need access to production environment"
- "I need a software license for Adobe Creative Suite"

---

2. Approver (Manager/Team Lead)

Who: Managers, team leads, budget owners

Capabilities:

- View all requests requiring their approval
- Approve or reject requests
- Add approval comments (reason for rejection, conditions, etc.)
- See request details and requester information
- Receive email notifications for new requests
- View approval history (what they've approved/rejected)

Decision Criteria:

- Budget availability
- Business justification
- Policy compliance
- Resource availability

---

3. Admin (System Administrator)

Who: IT admins, HR admins, facility managers

Capabilities:

- View all requests in the system
- Manage user accounts and roles
- Configure request types and workflows
- Generate reports and analytics
- Mark requests as fulfilled/completed
- Override workflows (if needed for exceptions)
- Manage system settings

Responsibilities:

- System configuration
- User management
- Fulfillment coordination
- Audit and compliance

Request Types (Examples)

The system supports various types of requests. Examples include:

Hardware Requests

- **Laptop/Desktop** - New computer for employee
- **Monitor** - Additional display
- **Peripherals** - Keyboard, mouse, headset, webcam
- **Mobile device** - Company phone or tablet

Software & Access

- **Software license** - Adobe, Microsoft Office, development tools
- **System access** - Production environment, admin rights
- **Application access** - CRM, ERP, internal tools
- **VPN access** - Remote work connectivity

Services & Facilities

- **Parking spot** - Reserved parking
- **Office equipment** - Desk, chair, standing desk
- **Training** - Course enrollment, conference attendance
- **Travel approval** - Business trip authorization

Note: The Product Owner will define the initial set of request types during Sprint 1. The system should be designed to allow easy addition of new types.

Key Features (MVP)

For Requesters

1. Request Submission Form

- Select request type from dropdown
- Fill in required fields (description, justification, etc.)
- Attach supporting documents (optional)
- Submit for approval

2. Request Dashboard

- View all own requests (past and present)
- See current status at a glance
- Filter by status (Pending, Approved, Rejected, Completed)
- Sort by date, type, status

3. Status Tracking

- Real-time status updates
- Timeline view of request progression
- See who approved/rejected and when
- Read approval comments

4. Notifications

- Email notification when status changes
- Reminder if action needed
- Confirmation on submission

For Approvers

1. Approval Queue

- List of requests awaiting approval
- Priority/urgency indicators
- Request details preview
- Quick approve/reject actions

2. Request Details View

- Full request information
- Requester details and history

- Business justification
- Cost implications (if applicable)

3. Approval Actions

- Approve with comments
- Reject with reason (required)
- Request more information
- Reassign to another approver

4. Approval History

- Track what I've approved/rejected
- See patterns and trends
- Export for reporting

For Admins

1. Admin Dashboard

- System-wide statistics
- Pending requests count
- Approval bottlenecks
- Request volume trends

2. User Management

- Add/remove users
- Assign roles (Requester, Approver, Admin)
- Manage approver hierarchies
- Deactivate users

3. System Configuration

- Define request types
- Configure approval workflows
- Set notification templates
- Manage system settings

4. Reporting & Analytics

- Requests by type, status, time period
- Approval times (average, median)
- Bottleneck identification
- User activity reports
- Export to Excel/PDF

5. Audit Trail

- Complete history of all actions
- Who did what and when
- Search and filter capabilities
- Compliance reporting

Non-Functional Requirements

Performance

- Page load time < 2 seconds
- Form submission response < 1 second
- Support 100 concurrent users (initial target)
- Database query optimization for large datasets

Security

- Role-based access control (RBAC)
- Secure authentication (OAuth2/JWT)
- HTTPS for all communications
- Input validation and sanitization
- SQL injection prevention
- XSS protection

Usability

- Mobile-responsive design
- Intuitive navigation
- Consistent UI patterns
- Clear error messages
- Helpful tooltips and guidance
- Accessibility (WCAG 2.1 Level AA)

Reliability

- 99% uptime during business hours
- Automated backups (daily)
- Error logging and monitoring
- Graceful error handling
- Rollback capability for deployments

Maintainability

- Clean, documented code
- Automated tests (unit, integration, e2e)
- CI/CD pipeline
- Version control (Git)
- ADR documentation for decisions

Privacy & GDPR Compliance

Personal Data We Process:

- Employee names and email addresses
- Request details and justifications (may contain sensitive information)
- Approval decisions and comments
- Audit trail (who did what, when)
- User roles and permissions

GDPR Requirements:

Lawful Basis:

- Legitimate interest (internal business operations)
- Staff will consult with organization's Data Protection Officer (DPO) or legal team to confirm lawful basis

Core Principles:

- **Data minimization:** Only collect data necessary for request processing
- **Purpose limitation:** Data used only for request management, not for other purposes
- **Storage limitation:** Define and implement retention policies
- **Integrity and confidentiality:** Secure storage and access controls

User Rights:

- **Right to access:** Users can view their own request history
- **Right to erasure:** Anonymization approach for closed requests (retain audit trail, remove PII)
- **Right to data portability:** Export functionality for user's own data
- **Right to rectification:** Users can update their own information

Technical Implementation:

- Separate PII from audit trail (enable anonymization without losing audit capability)
- Configurable retention policies (e.g., anonymize after 2 years)
- User data export functionality (JSON/CSV format)
- Anonymization capability for closed requests (replace names with "User [ID]")
- Privacy-aware logging (no sensitive request details in application logs)
- Access control prevents unauthorized viewing of personal data

Out of Scope for MVP:

- Formal Data Protection Impact Assessment (DPIA) - Staff responsibility
- Privacy dashboard for end users
- Automated data deletion workflows
- Consent management system
- Third-party data processor agreements
- Cross-border data transfer mechanisms

Staff Responsibilities:

- DPO/legal consultation before Sprint 1 (confirm lawful basis, retention period)
- Provide GDPR requirements to Product Owner
- Review architecture for GDPR compliance
- Approve retention and anonymization policies

Team Responsibilities:

- Document GDPR approach in ADR (e.g., ADR-004: GDPR Compliance Approach)
- Implement privacy by design from Sprint 1
- Include privacy review in Definition of Done
- Design database schema with anonymization in mind

Note: Product Owner will receive GDPR requirements from staff/DPO before Sprint Planning. The team is responsible for technical implementation, staff is responsible for legal compliance.

User Stories (Examples)

Epic: Request Submission

US-001: Submit Hardware Request

```
As an employee,
I want to submit a request for a new laptop,
So that I can get the equipment I need for my work.

Acceptance Criteria:
- I can select "Laptop" from request type dropdown
- I can specify laptop specifications (processor, RAM, storage)
- I can provide business justification
- I receive confirmation email after submission
- Request appears in my dashboard with "Submitted" status
```

US-002: Track Request Status

```
As an employee,
I want to see the current status of my requests,
So that I know if my request is being processed.

Acceptance Criteria:
- I can view all my requests in a dashboard
- Each request shows current status (Submitted, Under Review, Approved, etc.)
- I can see who is currently reviewing my request
- I can see timestamp of last status change
```

Epic: Request Approval

US-010: Approve/Reject Request

```
As a manager,
I want to review and approve or reject requests from my team,
So that I can control resource allocation.

Acceptance Criteria:
- I see a list of requests pending my approval
- I can view full request details including justification
- I can approve with one click
- I can reject with mandatory reason
- Requester receives email notification of my decision
```

US-011: Approval Queue Management

```
As a manager,
I want to see all pending approvals in one place,
So that I can efficiently process multiple requests.

Acceptance Criteria:
- Dashboard shows count of pending approvals
- I can sort by date, priority, requester
- I can filter by request type
- I can bulk-approve similar requests (stretch goal)
```

Epic: System Administration

US-020: User Management

```
As an admin,
I want to add new users and assign roles,
So that the system reflects our organization structure.

Acceptance Criteria:
- I can create new user accounts
- I can assign roles (Requester, Approver, Admin)
- I can deactivate users who leave the organization
- I can update user information
```

US-021: Generate Reports

```
As an admin,
I want to generate reports on request activity,
So that I can analyze trends and identify bottlenecks.

Acceptance Criteria:
- I can generate report for date range
- Report shows requests by type, status, approver
- I can export report to Excel
- Report includes average approval time
```

Out of Scope (Not in MVP)

The following features are **NOT** included in the initial release:

❌ **Multi-level approvals** - Only single approver per request (not chain of approvals)
❌ **Budget tracking** - No integration with financial systems
❌ **Asset inventory** - No tracking of physical assets after fulfillment
❌ **SLA management** - No automatic escalation based on time
❌ **Advanced workflows** - No conditional routing based on request details
❌ **Mobile apps** - Web only (responsive design, not native apps)
❌ **Integrations** - No integration with external systems (Slack, JIRA, etc.)
❌ **Real-time chat** - No in-app messaging between users
❌ **Approval delegation** - Approvers cannot delegate to others
❌ **Recurring requests** - No templates for repeated requests

Note: These features may be considered for future releases based on user feedback and business priorities.

Success Metrics

How do we know the EAP is successful?

Usage Metrics

- **Adoption rate:** 80% of employees submit at least one request in first 3 months
- **Active users:** 50+ active users per week
- **Request volume:** 100+ requests processed per month

Performance Metrics

- **Average approval time:** < 48 hours from submission to approval
- **Time to fulfillment:** < 5 days from approval to completion
- **System uptime:** 99% during business hours

Quality Metrics

- **User satisfaction:** Average rating 4/5 or higher
- **Rejection rate:** < 10% of requests rejected
- **Bug reports:** < 5 critical bugs per sprint

Process Metrics

- **Email reduction:** 70% fewer emails about request status
- **Admin time saved:** 20% reduction in admin time spent on request tracking
- **Audit compliance:** 100% of requests have complete audit trail

---

Technology Constraints

Based on **eap-architecture ADR-002: Technology Stack** (to be documented), the system will use:

Backend:

- Python 3.11+
- FastAPI framework
- PostgreSQL database
- SQLAlchemy ORM
- Alembic for migrations

Frontend:

- React 18
- TypeScript
- Modern component library (to be decided by team)

Infrastructure:

- Docker containers
- GitHub Actions for CI/CD
- TransIP VPS for deployment

API:

- RESTful API design
- OpenAPI/Swagger specification
- JWT for authentication

Product Roadmap (Tentative)

Sprint 1-2: Foundation (Weeks 1-4)

- User authentication and authorization
- Basic request submission (single request type)
- Simple approval workflow
- Email notifications
- Basic dashboard

Sprint 3-4: Core Features (Weeks 5-8)

- Multiple request types
- Request history and tracking
- Admin user management
- Status updates and comments
- Enhanced dashboards

Sprint 5-6: Polish & Scale (Weeks 9-12)

- Reporting and analytics
- Performance optimization
- Enhanced UI/UX
- Complete audit trail
- Documentation and training materials

Note: This roadmap is flexible and will be refined by the Product Owner based on team velocity and stakeholder feedback.

Stakeholders

Primary Stakeholders

- **Product Owner (Trainee)** - Defines and prioritizes features
- **Development Team (Trainees)** - Builds and delivers the system
- **End Users** - Employees who will use the system (represented by Product Owner)

Secondary Stakeholders

- **Staff/Coaches** - Provide guidance and assess learning outcomes
- **Organization** - Benefits from streamlined request process
- **IT Department** - Potential future maintainers of the system

Product Owner Responsibilities

The Product Owner (trainee role) is responsible for:

✅ **Refining this vision** into detailed user stories
✅ **Prioritizing the Product Backlog** based on business value
✅ **Defining acceptance criteria** for each user story
✅ **Making scope decisions** when trade-offs are needed
✅ **Representing user needs** in all discussions
✅ **Accepting or rejecting** completed work in Sprint Reviews
✅ **Updating this document** as the vision evolves (with team input)

Staff role: Staff coaches and challenges the Product Owner but does not make product decisions.

Design Principles

The EAP system should embody these principles:

1. **User-Centric** - Design for end users, not admins
2. **Transparent** - Everyone can see status and history
3. **Efficient** - Minimize clicks and waiting time
4. **Reliable** - No lost requests, complete audit trail
5. **Simple** - Prefer simple solutions over complex features
6. **Secure** - Protect user data and prevent unauthorized access
7. **Maintainable** - Clean code, good documentation, automated tests

---

Questions & Clarifications

This section will be updated as questions arise during development:

Q: Can a requester edit a request after submission?
A: To be decided by Product Owner in Sprint 1. Initial stance: No (requester must cancel and resubmit).

Q: What happens if an approver doesn't respond?
A: Out of scope for MVP. No automatic escalation. Manual follow-up via email.

Q: Can an admin override an approval decision?
A: To be decided by Product Owner. Consider audit implications.

Q: How long is request history retained?
A: Staff will provide GDPR-compliant retention policy (e.g., 2 years, then anonymize). Product Owner will implement this requirement.

Q: How do we handle "right to be forgotten" requests?
A: Anonymization approach: Replace PII with "User [ID]" while retaining audit trail structure. Staff/DPO will provide specific requirements.

Q: What is the lawful basis for processing employee data?
A: Staff/DPO will confirm (likely: legitimate interest for internal operations). Product Owner will receive this information before Sprint 1.

References

- **Kickoff Presentation:** EAP_Kickoff_Presentation_v1.1.pptx (Slide 4: Product Vision)
- **Project Initiation Document:** EAP_Project_Initiation_Document_Product_Focused_v1.0.md
- **Architecture Repository:** eap-architecture (to be created in Sprint 1)
- **User Stories:** To be detailed in Jira during Sprint Planning

---

Document History

Version	Date	Changes	Author
1.0	2026-01-06	Initial product vision document	Staff (for Product Owner to own)
1.1	2026-01-06	Added GDPR/Privacy compliance section	Staff (for Product Owner to own)

Next Steps

For Product Owner:

1. Review this vision document with the team in Sprint 1
2. Refine and adjust based on team input
3. Break down into detailed user stories in Jira
4. Prioritize initial backlog for Sprint 1
5. Define acceptance criteria for each story
6. Update this document as vision evolves

For Development Team:

1. Read and understand the product vision
2. Ask clarifying questions to Product Owner
3. Provide technical input on feasibility
4. Help break down features into implementable stories
5. Challenge assumptions and suggest improvements

Document Control

- Created: 2026-01-06
- Owner: Product Owner (trainee role)
- Approved by: Staff

- Review Frequency: After each Sprint (update as needed)
- Status: Living document (will evolve throughout project)