

Zabala Gailetak - App Unification Plan

Executive Summary

This document outlines the plan to unify the Android app and web portal by:

1. Adding missing features from Android to web
2. Aligning the Android app visual design with the web design system
3. Ensuring feature parity across both platforms

Status: Planning Phase

Target Completion: Q2 2026

Priority: High

1. Feature Gap Analysis

1.1 Android Features Missing from Web

A. Payslips Module (HIGH PRIORITY)

Current State:

- X Web: Not implemented (TODO placeholders in routes)
- ✓ Android: Fully functional with modern UI

Required Implementation:

- [] Database schema for payslips table
- [] Backend API endpoints:
 - GET /api/payroll - List employee payslips
 - GET /api/payroll/{id} - Get specific payslip details
 - GET /api/payroll/{id}/download - Download PDF
- [] Web UI Pages:
 - Payslips list view with cards
 - Detailed payslip view showing breakdown
 - PDF generation/download functionality
- [] Features:

- Month/year filtering
- Salary breakdown (gross, net, deductions, IRPF, social security)
- Bonus display
- Historical data access

Android UI Elements to Port:

- Gradient summary cards
 - Clean breakdown layout
 - Month/year navigation
 - Download icons and actions
-

B. Documents Module (HIGH PRIORITY)

Current State:

- Web: Not implemented (TODO placeholders)
- Android: Full document management with categorization

Required Implementation:

- [] Database schema for documents table (already exists in models)
- [] Backend API endpoints:
 - GET /api/documents - List documents
 - GET /api/documents/{id} - Get document details
 - POST /api/documents/upload - Upload new document
 - GET /api/documents/{id}/download - Download document
 - DELETE /api/documents/{id} - Delete document (admin only)
- [] Web UI Pages:
 - Documents list view with tabs (My Documents / Public Documents)
 - Document upload form
 - Document viewer/preview
- [] Document Categories:
 - CONTRACT (Kontratuak)
 - PAYSLIP (Nominak)
 - CERTIFICATE (Ziurtagiriak)
 - POLICY (Politikak)
 - OTHER (Beste batzuk)

- [] Features:
 - File type icons and colors
 - Category filtering
 - Search functionality
 - Bulk download option

Android UI Elements to Port:

- Tabbed interface
 - Category badges with colors
 - Icon-based file type display
 - Empty state designs
-

C. Enhanced Dashboard (MEDIUM PRIORITY)

Current State:

-  Web: Basic dashboard with stats
-  Android: Rich dashboard with multiple sections

Required Enhancements:

- [] Quick Action Cards Section:
 - Gradient card designs
 - Icon-based navigation
 - Direct links to common actions
- [] Statistics Overview:
 - Vacation days remaining
 - Hours worked today
 - Pending requests count
 - Total documents count
- [] Recent Activity Feed:
 - Timeline of recent actions
 - Status indicators (approved, pending, rejected)
 - Timestamps
- [] Personalization:
 - Time-based greetings (Basque/Spanish)
 - User-specific data display

- Notification badges

Android UI Elements to Port:

- Gradient quick action cards
 - Stat cards with icons
 - Activity timeline component
 - Notification badges
-

D. Profile/Settings (MEDIUM PRIORITY)

Current State:

- ! Web: Basic profile edit
- ✓ Android: Comprehensive profile management

Required Enhancements:

- [] Profile viewing/editing
 - [] Avatar/photo upload
 - [] Password change functionality
 - [] MFA settings toggle
 - [] Language preference
 - [] Notification preferences
 - [] Logout functionality
-

1.2 Web Features Not in Android

- ✗ Employee List Management (Admin only - not needed in mobile app)
 - ✗ HR Approval Workflows (Admin only - mobile has read-only view)
-

2. Design System Alignment

2.1 Color Palette Unification

Current Web Colors (from industrial-v2.php):

```
--primary: #1D4ED8 (Deep Blue)
--accent: #0EA5E9 (Light Blue)
--success: #059669 (Green)
--warning: #D97706 (Amber)
--danger: #DC2626 (Red)
```

Current Android Colors (from Color.kt):

```
PrimaryBlue = #2C3E95 (Royal Blue)
SecondaryTeal = #06B6D4 (Teal/Cyan)
AccentOrange = #FF6B35 (Orange)
AccentPurple = #9333EA (Purple)
SuccessGreen = #10B981 (Green)
```

⚠ CONFLICT: Different primary blues

Decision: Use Web Colors as Standard (already in production)

Action Items:

- [] Update Android Color.kt to match web palette:

```
PrimaryBlue = Color(0xFF1D4ED8) // Match web --primary
AccentBlue = Color(0xFF0EA5E9) // Match web --accent
SuccessGreen = Color(0xFF059669) // Match web --success
WarningAmber = Color(0xFFD97706) // Match web --warning
ErrorRed = Color(0xFFDC2626) // Match web --danger
```

- [] Keep gradient colors as enhancement (not in web yet)
 - [] Update all Android UI components to use new colors
 - [] Test color contrast ratios (WCAG AA compliance)
-

2.2 Typography Alignment

Web (Industrial v2):

- Font: System fonts (-apple-system, Segoe UI, Roboto)
- Sizes: xs(0.75rem) to 5xl(3rem)

Android (Material 3):

- Font: Default Material Typography
- Scale: bodySmall to displayLarge

Action Items:

- [] Ensure Android uses system font (already does via Material 3)
 - [] Map text sizes consistently between platforms
 - [] Use same font weights (400, 600, 700, 800)
-

2.3 Component Style Alignment

Buttons

Web: Rounded (–radius-md: 10px), solid colors **Android:** Rounded (RoundedCornerShape(16.dp)), gradient options

Action:

- [] Update Android buttons to match web radius (10dp)
- [] Remove gradients from buttons (keep for cards)
- [] Use solid colors matching web palette

Cards

Web: Subtle shadows (–shadow-md), light borders **Android:** Elevation-based shadows, gradient options

Action:

- [] Reduce Android card elevation to match web
- [] Keep gradient cards for dashboard quick actions only
- [] Use consistent border radius (16dp / 16px)

Navigation

Web: Top navbar with horizontal links **Android:** Bottom navigation bar with icons

Action:

- [] Keep platform-specific navigation (expected pattern)
- [] Ensure same color scheme applies
- [] Use same icon set (Material Icons)

2.4 Iconography

Both platforms: Use Material Icons

Action:

- [] Audit all icons used in both platforms
- [] Ensure consistent icon usage (e.g., both use BeachAccess for vacations)
- [] Update icon colors to match new palette

3. Implementation Roadmap

Phase 1: Backend Foundation (Week 1-2)

Priority: HIGH

Effort: 40 hours

- [] Design payslips database table
- [] Design documents table (already in models, verify)
- [] Create migration scripts
- [] Implement PayrollController with CRUD operations
- [] Implement DocumentController with upload/download
- [] Create API endpoints for both modules
- [] Write unit tests for controllers
- [] Add audit logging for sensitive operations

Deliverables:

- Functional API endpoints for payslips
 - Functional API endpoints for documents
 - Database schema updated
 - API documentation updated
-

Phase 2: Web Frontend - Payslips (Week 3-4)

Priority: HIGH

Effort: 35 hours

- [] Create payslips list view (PHP template)
- [] Create payslip detail view
- [] Implement PDF generation using TCPDF or similar
- [] Add filtering (month, year)
- [] Style using existing industrial CSS
- [] Add download functionality
- [] Implement salary breakdown display
- [] Add navigation link in main menu
- [] Test responsive design

Deliverables:

- Functional payslips module in web
- PDF download capability
- Responsive UI matching design system

Phase 3: Web Frontend - Documents (Week 5-6)

Priority: HIGH

Effort: 35 hours

- [] Create documents list view with tabs
- [] Implement file upload form with validation
- [] Create document categorization UI
- [] Add download functionality
- [] Implement category badges with icons
- [] Add search/filter functionality
- [] Create empty state designs
- [] Add navigation link in main menu
- [] Implement file type detection and icons
- [] Test file upload security

Deliverables:

- Functional documents module in web
 - File upload/download working
 - Category-based organization
 - Responsive UI
-

Phase 4: Web Frontend - Enhanced Dashboard (Week 7)

Priority: MEDIUM

Effort: 20 hours

- [] Add quick action cards section
- [] Implement statistics overview cards
- [] Create recent activity feed
- [] Add time-based greeting
- [] Style with gradient cards (match Android)
- [] Add notification badge functionality
- [] Implement AJAX for real-time updates
- [] Test performance with live data

Deliverables:

- Enhanced dashboard matching Android design

- Quick actions working
 - Real-time statistics
-

Phase 5: Android Style Updates (Week 8)

Priority: MEDIUM

Effort: 25 hours

- [] Update Color.kt with web palette
- [] Update all screens to use new colors
- [] Adjust button styles (remove gradients)
- [] Update card elevations
- [] Test all screens with new design
- [] Update theme configurations
- [] Rebuild and test app thoroughly
- [] Update screenshots and marketing materials

Deliverables:

- Android app visually aligned with web
 - Consistent color palette across platforms
 - No functional regressions
-

Phase 6: Testing & Documentation (Week 9-10)

Priority: HIGH

Effort: 30 hours

- [] Cross-platform feature testing
- [] Security audit (file uploads, downloads)
- [] Performance testing (PDF generation)
- [] Accessibility testing (WCAG AA)
- [] Update API documentation
- [] Update user guides
- [] Create training materials
- [] Conduct user acceptance testing
- [] Fix identified bugs

Deliverables:

- Test reports

- Updated documentation
 - UAT sign-off
 - Production-ready code
-

4. Technical Specifications

4.1 Database Schema

Payslips Table

```
CREATE TABLE payslips (
    id INT PRIMARY KEY AUTO_INCREMENT,
    employee_id INT NOT NULL,
    month TINYINT NOT NULL, -- 1-12
    year SMALLINT NOT NULL,
    gross_salary DECIMAL(10,2) NOT NULL,
    net_salary DECIMAL(10,2) NOT NULL,
    bonuses DECIMAL(10,2) DEFAULT 0.00,
    overtime DECIMAL(10,2) DEFAULT 0.00,
    social_security DECIMAL(10,2) NOT NULL,
    irpf DECIMAL(10,2) NOT NULL,
    other_deductions DECIMAL(10,2) DEFAULT 0.00,
    pdf_path VARCHAR(255),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    FOREIGN KEY (employee_id) REFERENCES users(id),
    UNIQUE KEY unique_payslip (employee_id, month, year),
    INDEX idx_employee (employee_id),
    INDEX idx_period (year, month)
);
```

Documents Table (verify existing schema)

```
CREATE TABLE documents (
    id INT PRIMARY KEY AUTO_INCREMENT,
    employee_id INT NULL, -- NULL for public documents
    title VARCHAR(255) NOT NULL,
    description TEXT,
    category ENUM('CONTRACT', 'PAYSLIP', 'CERTIFICATE', 'POLICY', 'OTHER') NOT NULL,
    file_path VARCHAR(255) NOT NULL,
    file_name VARCHAR(255) NOT NULL,
    file_type VARCHAR(10) NOT NULL,
    file_size INT NOT NULL,
    is_public BOOLEAN DEFAULT FALSE,
    uploaded_by INT NOT NULL,
    uploaded_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (employee_id) REFERENCES users(id),
```

```
FOREIGN KEY (uploaded_by) REFERENCES users(id),  
INDEX idx_employee (employee_id),  
INDEX idx_category (category),  
INDEX idx_public (is_public)  
);
```

4.2 API Endpoints

Payslips API

GET	/api/payroll	- List employee's payslips
GET	/api/payroll/{id}	- Get specific payslip details
GET	/api/payroll/{id}/download	- Download payslip PDF
POST	/api/payroll	- Create new payslip (ADMIN only)
PUT	/api/payroll/{id}	- Update payslip (ADMIN only)
DELETE	/api/payroll/{id}	- Delete payslip (ADMIN only)

Documents API

GET	/api/documents	- List documents (personal + public)
GET	/api/documents/{id}	- Get document details
GET	/api/documents/{id}/download	- Download document file
POST	/api/documents/upload	- Upload new document
DELETE	/api/documents/{id}	- Delete document (owner or ADMIN)
GET	/api/documents/categories	- Get document categories

4.3 Security Considerations

File Upload Security

- [] Validate file types (whitelist: PDF, DOC, DOCX, JPG, PNG)
- [] Limit file size (max 10MB)
- [] Sanitize file names (remove special characters)
- [] Store files outside web root
- [] Generate unique file names (UUID)
- [] Scan files for malware (ClamAV integration)
- [] Verify MIME types (don't trust extension)

Access Control

- [] Users can only view their own payslips
- [] Users can only view their own documents + public documents

- [] ADMIN/RRHH_MGR can view all payslips
- [] ADMIN can upload/delete documents
- [] Audit log all download actions

PDF Generation

- [] Use secure PDF library (TCPDF)
 - [] Prevent XSS in PDF content
 - [] Add watermark with generation timestamp
 - [] Include digital signature (optional)
-

5. Quality Assurance

5.1 Testing Checklist

Unit Tests

- [] PayrollController methods
- [] DocumentController methods
- [] File upload validation
- [] PDF generation
- [] Access control logic

Integration Tests

- [] API endpoint responses
- [] File upload flow
- [] PDF download flow
- [] Authentication middleware
- [] Authorization checks

E2E Tests

- [] User views payslips
- [] User downloads payslip PDF
- [] User uploads document
- [] User downloads document
- [] Admin creates payslip
- [] Access denied scenarios

Security Tests

- [] File upload vulnerabilities
- [] Path traversal attacks
- [] Unauthorized access attempts
- [] XSS in document names
- [] SQL injection in filters

Performance Tests

- [] PDF generation time (< 3 seconds)
 - [] File upload time (< 10 seconds for 10MB)
 - [] List queries (< 500ms)
 - [] Concurrent downloads (100 users)
-

5.2 Acceptance Criteria

Payslips Module

- Employees can view list of their payslips
- Employees can filter by month/year
- Employees can view detailed breakdown
- Employees can download PDF
- PDF contains all required information
- Admin can create/edit payslips
- Responsive design (mobile, tablet, desktop)

Documents Module

- Employees can view personal documents
- All users can view public documents
- Documents categorized correctly
- Employees can download documents
- Admin can upload documents
- File upload validates correctly
- No security vulnerabilities

Design Alignment

- Android app uses web color palette

- Consistent typography
 - Consistent component styling
 - Consistent iconography
 - Both platforms feel cohesive
-

6. Risk Management

6.1 Identified Risks

Risk	Probability	Impact	Mitigation
PDF generation performance issues	Medium	High	Implement caching, async generation
File upload security vulnerabilities	High	Critical	Comprehensive security validation
Color change breaks Android UI	Low	Medium	Thorough testing, rollback plan
Database schema changes break existing code	Medium	High	Comprehensive migration testing
User confusion with new features	Medium	Medium	User training, in-app guides
Timeline overrun	Medium	Medium	Prioritize features, agile approach

6.2 Contingency Plans

If PDF generation is slow:

- Implement asynchronous generation with email notification
- Cache generated PDFs
- Consider external service (WeasyPrint, wkhtmltopdf)

If file upload has security issues:

- Temporarily disable upload for non-admins
- Implement stricter validation
- Add additional malware scanning

If Android color changes cause issues:

- Maintain color variants in theme

- Use semantic color naming
 - Implement feature flags for gradual rollout
-

7. Success Metrics

7.1 Technical Metrics

- [] 100% API endpoint coverage for new features
- [] 90%+ unit test coverage for new code
- [] < 3 second PDF generation time
- [] < 2 second page load times
- [] Zero critical security vulnerabilities
- [] WCAG AA accessibility compliance

7.2 User Experience Metrics

- [] 95%+ user satisfaction rating
- [] < 5% support tickets related to new features
- [] 80%+ feature adoption within first month
- [] Mobile app rating maintains 4.5+ stars
- [] Zero data breach incidents

7.3 Business Metrics

- [] Reduce HR administrative time by 30%
 - [] 100% digital payslip distribution
 - [] Reduce paper document storage costs
 - [] Improve compliance audit scores
 - [] Increase employee self-service usage
-

8. Post-Launch Activities

8.1 Monitoring

- [] Set up monitoring for new API endpoints
- [] Monitor file storage usage
- [] Track PDF generation performance
- [] Monitor download bandwidth

- [] Track feature usage analytics

8.2 Maintenance

- [] Monthly security audits
- [] Quarterly performance reviews
- [] Regular backup verification
- [] File storage cleanup (old documents)
- [] PDF cache maintenance

8.3 Future Enhancements

- [] Bulk payslip upload for HR
 - [] Electronic signature for documents
 - [] Document versioning
 - [] Document expiry notifications
 - [] Advanced search with filters
 - [] Document preview in browser
 - [] Mobile document scanning
-

9. Team & Resources

9.1 Required Skills

- PHP backend development
- PostgreSQL database design
- PDF generation (TCPDF)
- File upload handling
- Kotlin/Jetpack Compose (Android)
- Material Design 3
- Security best practices
- Testing (Unit, Integration, E2E)

9.2 Estimated Effort

- **Total:** 185 hours (~23 days)
- **Backend:** 40 hours
- **Web Frontend:** 90 hours
- **Android Updates:** 25 hours

- **Testing & QA:** 30 hours

9.3 Dependencies

- PHP 8.4+
 - PostgreSQL 16
 - TCPDF library
 - File storage space (estimated 100GB)
 - Testing environments
 - CI/CD pipeline updates
-

10. Conclusion

This unification plan will bring feature parity between the Android app and web portal while ensuring a consistent visual design across both platforms. The phased approach allows for iterative development with regular testing and validation.

Next Steps:

1. Review and approve this plan
2. Allocate resources and timeline
3. Begin Phase 1: Backend Foundation
4. Set up project tracking (Jira/GitHub Projects)
5. Schedule weekly progress reviews

Sign-off Required:

- [] Technical Lead
 - [] Product Owner
 - [] Security Team
 - [] Stakeholders
-

Document Version: 1.0

Last Updated: 2026-02-06

Author: AI Development Assistant

Status: Awaiting Approval